Application Of Coding Arithmetic Methods And Methods Lzw For Compression Files

Siddik Karo-Karo[1], Cosmas Samuel Daeli[2], Mamed Rofendi Manalu[3]
[1,2,3] Universitas Imleda Medan, Indonesia

Article Info

Article history:

Received, May 18, 2023
Revised, May 30, 2023
Accepted, June 11, 2023

Keywords:

file compression,
Arithmetic Coding,
LZW
storage efficiency,
compression ratio.

ABSTRACT

Efficient data storage and transmission have become of utmost importance in today's digital era. File compression processes are crucial to addressing this issue. The purpose of this study is to examine and compare the use of the Arithmetic Coding Method and the LZW Method in file compression. Both of these methods are known for their excellent compression capabilities, but there has been limited research comparing them in the context of real-world use. This research employed a range of files, including text, photos, and recordings, to test both approaches. Results indicate that the arithmetic coding method typically offers a slightly better compression ratio compared to the LZW method. However, LZW provides faster compression and decompression speeds, making it more suitable for real-time applications or where response time is critical. Nonetheless, the choice between these two methods largely depends on the specific needs of a given application. For purposes requiring the highest compression ratio, the arithmetic coding method might be more appropriate. however, the LZW method might be more suitable for purposes demanding quick compression and decompression. In summary, both the arithmetic coding and LZW methods have their own strengths and weaknesses. A thorough understanding of the features and requirements of the file to be compressed is vital for choosing the most fitting method. the LZW method might be more suitable for purposes demanding quick compression and decompression. In summary, both the arithmetic coding and LZW methods have their own strengths and weaknesses. A thorough understanding of the features and requirements of the file to be compressed is vital for choosing the most fitting method. the LZW method might be more suitable for purposes demanding quick compression and decompression. In summary, both the arithmetic coding and LZW methods have their own strengths and weaknesses. A thorough understanding of the features and requirements of the file to be compressed is vital for choosing the most fitting method.

Corresponding Author:

Siddik Karo-Karo,
Universitas Imelda Medan,
Jl. Bilal No. 52 Village of Pulo Brayan Darat I, East Medan District, Medan - North Sumatra.
E-mail:siddikzidane85@gmail.com

1. INTRODUCTION

Computer science examines data compression to reduce file size before storing or sending data. Compression and decompression are the two basic steps in data compression. After decompression, a compressed file must be readable. Lossless and lossy compression methods can compress data. Lossless compression preserves all data during decompression. Lossy compression compresses data with missing information, but visual perception can still tolerate it.

A file is specified by inserting one or more end-of-file characters after the last line. Files normally include the type Multipurpose Internet Mail Extension to indicate an encoding. Windows considers files with extensions like .txt,.doc,.xlsx, and other files.

Arithmetic Coding compresses better than Huffman Coding, however the vast amount of calculation procedures slows compression and decompression. Arithmetic Coding replaces symbols without a code. Arithmetic Coding fractions the entire symbol stream. Mathematical coding produces a fraction higher than zero and less than one, which can be decoded to reveal the original symbols.

The LZW compression method is derived from the LZ78. A dictionary of terms is continuously entered into LZW while coding. The dictionary is updated with new phrases until its space is full. The LZW dictionary contains sentences with one character, all ASCII characters before encoding. Character ASCII codes are used to code these phrases.

2. RESEARCH METHOD

This research methodology is used as a guide in conducting research so that the results achieved do not deviate from the objectives. The research methodology is as follows:

1. Library Studies (Literature Review)
   Literature study is a data collection technique by studying sources of reading material in the form of books related to the material discussed such as improving the quality of files, which are related to the problems encountered in order to obtain the necessary data and materials.
2. Analysis and design (Analysis and Design)
   At this stage it is used to process existing data and then analyze the results of the literature study obtained so that it becomes information.
3. Testing (examination)
   This test will test the system as a whole whether the application that has been made can run properly according to the goals to be achieved.
4. System implementation (Implementation system)

This implementation will be carried out using the algorithm of the arithmetic coding method and the LZW method.

3. RESULTS AND DISCUSSION

3.1. Ziv Welch Paste Compression Algorithm (LZW)

The Lempel Ziv Welch (LZW) compression algorithm is an algorithm for compressing files, here is the compression algorithm:

Inputs:

W → Size*files*original in the character stream
K → Size*files*every character

output:

R → Compression ratio

Process:

W → New string in character stream
W = W+K

count
Compression ratio = (original file size - compressed file size x 100)
Original file size
Finished

Arithmatic Coding Compression Algorithm

Compression Arithmatic Coding Algorithm is an algorithm for compressing files, here is the compression algorithm:

Input I :
low = the value that has been calculated from the first character that will be used in the next data but in the initial conditions given

value 0.0.
high = the value that has been calculated from the first character which will be used in the next data but in the initial conditions given
value 0.0.
Process I:
CR = high – low
Output I:
CR = subtraction of high and low values.
Input II :
low = the value that has been calculated from the first character which will be used in the next data but in the initial conditions it is given a value of 0.0.
CR = subtraction of high and low values.
high_range = the value of the upper limit taken from the range probability table
low_range = the value of the lower limit taken from the range probability table
A = data or character being counted
Process II:
High = low + CR*high_range(A)
Low = low + CR*low_range(A)
Output II:
High = the calculation result of the low, CR, and high_range values then the result becomes the high value in the next data.
Low = the result of the calculation of the low, CR, and low_range values then the result will be a low value in the next data.

Implementation

Implementation is a step used to operate the system being built. In this study explained how to run the system. The program processing system is a processing unit consisting of procedures and implementation of data. Computers as a means of processing programs must provide supporting facilities in processing.

Provide a statement that is expected, as stated in the "Introduction" chapter can ultimately result in the "Results and Discussion" chapter, so there is compatibility. Moreover, it can also add prospects for the development of research results and application prospects for further studies into the next (based on results and discussion).

1. Main Menu Form

On the main menu display there is a Compression button, a Decompression button, a Help button and an Exit button. Main Menu display can be seen in Figure 1.

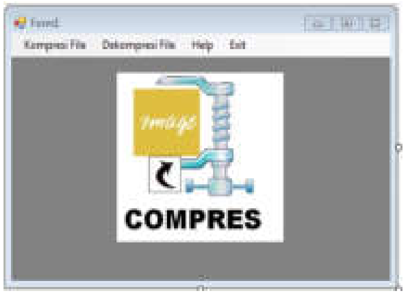


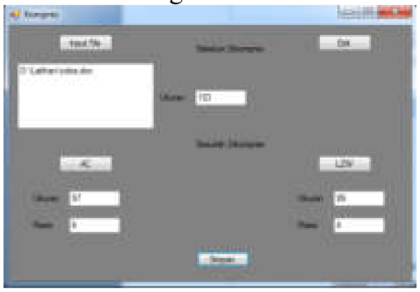
Figure 1 Display of the Main Menu Compression Form

The description of the main form interface is as follows:

a. The file compress menu functions to compress using the Arithmatic Coding and Lempel Ziv Welch (LZW) methods
b. The file decompression menu functions to decompress using the Arithmatic Coding and Lempel Ziv Welch (LZW) methods.
c. The Help menu functions to provide information on how to use the system. The exit menu functions to exit the program

2. Compression View

The Compression display is useful for reducing the bits in the sample file and saving the results. Compression display can be seen in Figure 2.



D:\Latihan\coba.doc

Ukuran 110

Ukuran 97

Rasio 6

Ukuran 25

Rasio 8

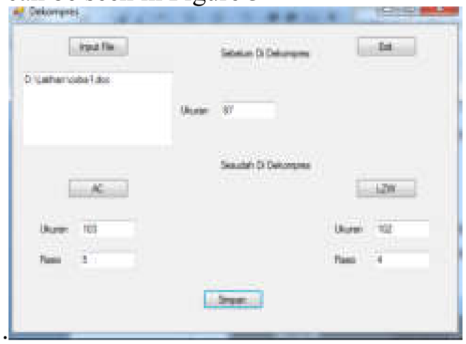
Figure 2 Display of RLE Compression

The description of the main form interface is as follows:

a. Input file functions to search for files contained in the directory to be compressed
b. The AC button functions for file compression from the Arithmatic Coding method
c. The LZW button works for file compression from the Lempel Ziv Welch (LZW) method
d. The Save button functions to save the decompressed file.
e. The exit button functions to exit the decompress program

3. Decompression View

The Compression display is useful for reducing the bits in the sample file and saving the results. Compression display can be seen in Figure 3



D:\Latihan\coba1.doc

Ukuran 97

Ukuran 103

Rasio 5

Ukuran 102

Rasio 4

Figure 3 Display of LZW Compression

The description of the main form interface is as follows:

a. Input file functions to find files in the directory to be decompressed
b. The AC button functions to decompress files from the Arithmatic Coding method
c. The LZW button functions to decompress files from the Lempel Ziv Welch (LZW) method
d. The Save button functions to save the decompressed file.
e. The exit button functions to exit the decompress program

4. CONCLUSION

In file compression calculations, the process begins by selecting the desired file, which is then transformed into a matrix used for compression computations. The implementation of the arithmetic coding and LZW methods for file compression starts with the creation of a probability table, followed by assigning a range to each data point and calculating using both the arithmetic coding and LZW methods to produce the final table. This compression design is integrated into Visual Studio 2008, creating a main form comprising actions for selecting the file to be compressed and actions to save it as an output file.

REFERENCES

[1]. Anton, 2005. Kompresi dan Teks. Fakultas Teknik Informatika. Univesitas Kristen Duta Wacan http://lecturer.ukdw.ac.id/anton/download/multimedia6.pdf Tanggal Akses: 12 Februari 2016

[2]. Mokhamad, Adi Pn Fazmah Arief Yulianto, Endro Ariyanto 2008 Analisis dan Implementassi Perbandinga Kinerja Algoritma Kompresi Huffman,LZW,dan DMC Pada Berbagai Tipe File Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom Tanggal Akses 5 Februari 2016

[3]. Taleumbanua, Pilipus 2011 Analisis Perbandingan Algoritma Kompresi LEMPEL ZIV WELCH, ARITHMETIC CODING, dan RUN-LENGTH ENCODING Pada File Teks Fakultas Matematika dan Ilmu Pengetahuan Alam, Ilmu Komputer, Universitas Sumatra Utara. Tanggal Akses : 12 Februari 2016

[4]. Subarka Aan, Fuad 2010 Rancangan Bangun Aplikasi Kompresi File Menggunakan Metode LZW Berbasis Java Fakultas Sains Teknologi, Teknik Informatika, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang http://lib.uinmalang.ac.id/files/thesis/fullchapter/05550055.pdf Tanggal Akses:01 Februari

[5]. Petrus Santoso, 2001 Studi Kompresi Data dengan Metode Arithmetic Coding JurusanTeknik Elektro, Fakultas Teknologi Industri – Universitas Kristen Petra http://puslit.petra.ac.id/journals/electrical/ Tanggal Akses : 01 Februari 2016

[6]. Suarga, 2006 Algoritma dan Pemograman. Andi Offset. Yogyakarta. ISBN : 978-979-292745-0.

[7]. Rajagukguk, D. M. (2014). Analisis Perbandingan Algoritma Huffman Dengan Algoritma (Lempel-ZIP-Welch) pada Kompresi Gambar Menggunakan Metode Exponensial. Pelita Informatika Budi Darma, 6(2), 21-25.