Scientific Journal of Informatics
Vol. 6, No. 1, May 2019

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Compression and Decompression of Audio Files Using the Arithmetic Coding Method

Parasian D.P Silitonga[1], Irene Sri Morina[2]

[1]Computer Science Department, Faculty of Computer Scince, St. Thomas Catholic University
[2]Computer Science Department, Faculty of Computer Scince, St. Thomas Catholic University
E-mail: parasianirene@gmail.com, morina_ginting@yahoo.com

Abstract

Audio file size is relatively larger when compared to files with text format. Large files can cause various obstacles in the form of large space requirements for storage and a long enough time in the shipping process. File compression is one solution that can be done to overcome the problem of large file sizes. Arithmetic coding is one algorithm that can be used to compress audio files. The arithmetic coding algorithm encodes the audio file and changes one row of input symbols with a floating point number and obtains the output of the encoding in the form of a number of values greater than 0 and smaller than 1. The process of compression and decompression of audio files in this study is done against several wave files. Wave files are standard audio file formats developed by Microsoft and IBM that are stored using PCM (Pulse Code Modulation) coding. The wave file compression ratio obtained in this study was 16.12 percent with an average compression process time of 45.89 seconds, while the average decompression time was 0.32 seconds.

Keywords: Audio File, Wave File, Compression and Decompression, Arithmetic Coding.

1. INTRODUCTION

Audio (sound) is a physical phenomenon produced by the vibration of an object in the form of an analog signal with an amplitude that changes continuously with time called frequency [3]. Sounds in wave form are stored in digital audio data format in computer system files. There are several formats for storing audio files on computer systems including wave files.

Wave files are standard audio file formats developed by Microsoft and IBM [10]. Wave files are stored using PCM (Pulse Code Modulation) coding. The wave file is an audio file that is not compressed so that all audio samples are stored all in the storage media in digital form.

Audio files on computer systems tend to have a large size, according to the length of recording time. In the process of storing and sending data, large files have constraints that require large space to store and require a considerable amount of time on delivery

[9]. To overcome this, file compression can be done. Compression is the process of encoding information using fewer bits than the initial information [5]. There are two types of compression, namely Lossless Compression and Lossy Compression [8].
Data compression through the encoding process seeks to eliminate the repetition of data by changing it in such a way as to produce smaller data sizes [7]. The compression encoding process can be carried out for various types of data such as text, image, video, audio and others.

The arithmatic coding algorithm is a compression method that replaces one row of input symbols with a floating point number [2]. The basic idea of arithmatic coding is to create an opportunity line from 0 to 1 and give an interval for each character from the input text based on the chance of its appearance. The higher the chance a character has, the greater the interval that will be obtained [6]. After all the characters have an interval, coding is done to produce an output number. Based on this description, this study discusses the process of compression and decompression of audio files using the arithmatic coding method. The results obtained in this study are in the form of a percentage of average file compression performed on several sample wave files.

2. METHODS

2.1. Compression and Decompression

Compression or data compression is a method used to compress data so that it only requires a smaller storage room. The main purpose of the data compaction process is to increase efficiency in storage or shorten the data exchange time. Compression is the process of encoding information using fewer bits than the initial information [5]. The general principle in the compression process is to reduce data duplication so that the memory to represent becomes less when compared to the original digital data representation [8].

There are two types of compression, namely Lossless Compression and Lossy Compression [8]. In lossless compression, the data will initially be broken down into smaller sizes and eventually the data is reunited. Whereas, in lossy compression, there are bits of information that are eliminated after the compression process is done [5]. The decompression process is the process of returning a compressed file to the initial text. Decompression results depend on the nature of the compression used, namely Lossless Compression or Lossy Compression.

2.2. Arithmetic Coding

The arithmatic coding algorithm is a compression method that replaces one row of input symbols with a floating point number [2]. The basic idea of arithmatic coding is to create an opportunity line from 0 to 1 and give an interval for each character from the input text based on the chance of its appearance. The higher the chance a character has, the greater the interval that will be obtained [6].

The output of arithmatic coding is a number smaller than number 1 and greater than or equal to 0. This number can be uniquely decoded to produce a row of symbols used to produce that number. To produce the output number, each symbol that will be encoded is given a set of probability values. For example, note a sample audio data 00 3e 1f 00 9a 00 1f 9a 00 3e 00 1f 00 3e 1f which will be encoded, then the probability table generated is like Table 1.

Table 1. Audio Sample Probability Table

Character	Frequency	Probability
00	6	6/15=0,4
1f	4	4/15=0,3
9a	2	2/15=0,1
3e	3	3/15=0,2

After the probability of each character is known, each symbol / character will be given a certain range whose values range from 0 and 1 according to the probabilities that exist. In this case there is no stipulation sequence for segments, what is important is that both the encoder and decoder must do the same. The Probability Range table is generated as in Table 2.

Table 2. Table of Audio Sample Probability Range

Character	Frequency	Probability	Range
00	6	6/15=0,4	0 ≤ 00 <0,4
1f	4	4/15=0,3	0,4 ≤ 1f <0,7
9a	2	2/15=0,1	0,7 ≤ 9a <0,8
3e	3	3/15=0,2	0,8 ≤ 3e <1,0

The next step, the encoding process is carried out based on the following steps:

1) Set low = 0.0 (initial condition)
2) Set high = 1.0 (initial condition)
3) While (input symbol still exists) do
4) Take the input symbol.
5) CR = high - low.
6) High = low + CR * high_range (symbol)
7) Low = low + CR * low_range (symbol)
8) End while
9) Print low

Based on the steps in the encoding process, the results of sample data encoding are obtained as in Table 3.

Table 3. Audio Sample Encoding Results

No	Character	Low	High	CR
Initial		0,0	1,0	1,0
1	00	0,0	1.0	1,0
2	3e	0,0	0,4	0,4
3	1f	0,32	0,4	0,08
4	00	0,352	0,376	0,024
5	9a	0,352	0,3616	0,0096
6	00	0,35872	0,35968	0,00096
7	1f	0,35872	0,359104	0,000384
8	9a	0,3588736	0,3589888	0,0001152
9	00	0,35906944	0,35908096	0,00001152
10	3e	0,35906944	0,359074048	0,000004608

11	00	0,3590731264	0,359074048	0,0000009216
12	1f	0,3590731264	0,35907349504	0,000000036864
13	00	0,359073273856	0,359073384448	0,000000147456
14	3e	0,359073273856	0,3590733328384	0,0000000589824
15	1f	0,35907332104192	0,3590733328384	0,00000001179648

Based on the data in Table 3, the low value for the last data is 0.35907332104192. This value is used to replace audio data 00 3e 1f 00 9a 00 1f 9a 00 3e 00 1f 00 3e 1f. While the decoding process is carried out through the following stages:

1) Take an encoded-symbol (ES).
2) Do
3) Look for the range of symbols surrounding ES.
4) Print symbol
5) RC = high_range - low_range
6) ES = ES - low_range
7) ES = ES / CR
8) Until the symbol runs out

2.3. Audio File

Audio (sound) is a physical phenomenon produced by the vibration of an object in the form of an analog signal with an amplitude that changes continuously with time called frequency [3].

Analog sound waves cannot be represented directly on a computer so they must be converted to digital form. The computer measures the amplitude at a certain time unit to produce a number of numbers. Each unit of measurement is called a sample.

Analog To Digital Conversion (ADC) is the process of changing the amplitude of a sound wave to a certain interval (sampling), so as to produce a digital representation of sound. In a sampling technique it is known as the sampling rate, which is a number of waves taken in one second. For example, if the quality of an audio CD is said to have a frequency of 44100 Hz, then the number of samples is 44100 per second.

2.4. Wave File

Wave files are standard audio file formats developed by Microsoft and IBM [10]. Wave files are stored using PCM (Pulse Code Modulation) coding. The wave file is an audio file that is not compressed so that all audio samples are stored all in the storage media in digital form.

Wave allow various forms of audio to be recorded in various qualities, such as 8-bit or 16-bit samples with rates of 11025 Hz, 22050 Hz or 44100 Hz [M. Kaur and S. Kaur]. Digital audio data in wave files can have various qualities. The quality of the sound produced is determined by the bitrate, samplerate, and number of channels [1].

Bitrate is a bit size for each side, namely 8-bits, 16-bits, 24-bits or 32-bits [4]. In 8-bits WAV all the samples will only take 1 byte. Whereas 16-bits will take 2 bytes. The sampler states the number of samples played every second. Commonly used samplers are 8000 Hz, 1105 Hz, 22050 Hz, and 44100 Hz [4]. While the number of channels determines the sound produced is mono or stereo [4]. Mono has only 1 channel, while stereo 2 channel and takes up 2 times more space than mono.

3. RESULT AND DISCUSSION

Before the audio file is compressed, the audio file is read to get the data in the form of a header in byte size (8 bits) in the form of hexadecimal number pairs. File wave is an audio file that is not compressed, which consists of headers containing information about audio files. Header data obtained from audio format wave files as in Figure 1.

```
52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00 22 56 00 00
88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00 00 24 17 1e 3c 13 3c 14 16
f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 c0 1a 0d 00 7f 11 00 22 07 3d 3d 0 3d  0 2 9a 2a 4f
3e 3d 00 10 10 10 03 3d 10 10 2 f 10 10 12 00 12 10 25 23 13 00 04 11 23 00 00 12 11
32 2a 2a 9f 25  10 -10 02 12 10 53 13 00 a6 11 12 23 00  0f 10 10  10 12 10 5d 23 13 70
11 26 23 00 01 13 9a 2a 2f 10 10 10 12 10 53 90 56 11 23 00 01 13 0e 22 52 9a 2a 10 3d
10 02 2f 10 10 10 12 -10 11 33 13 00  07 3d 10 02 3f 10 10 12 12 53 13  30 30 37 07 13
3d 10 02 f 0 10 10 12 -10 53 14 13 00 07 3d 10 02 f 0 10 00 12 26 25.
```

Figure 1. Header Data Results of Reading Wave Files

The information obtained from the audio file data in Figure 1 is explained below:

1) The first four bytes always contain 52 48 46 46 (hexa) which if at the convention means R = 52, I = 49, F = 46, F = 46 is the same as RIFF.
2) The next four bytes containing 24 08 00 00 state the audio file size, which is 24 = 36, 08 = 8, 00 = 0, 00 = 0 which is equal to 36800, then the file size is 36800 kb - 1 kb = 36799 kb.
3) The next four bytes 57 41 56 45 state the file type: 57 = W, 41 = A, 56 = V, 45 = E
4) The next four bytes are 66 6d 74 20 declares ID "fmt", 66 = f, 6d = m, 74 = t and 20 = empty spaces.
5) The next four bytes are 10 00 00 00 which states the length of information, 10 = 16, 00 = 0, 00 = 0, 00 = 0 all of which are worth 16.
6) The next four bytes are 01 00 02 00 which is worth 1 and 2 channels (stereo).
7) The next four bytes are 22 56 00 00 which states the sample rate with the value 22 = ", 56 = V, 00 = 0, 00 = 0.
8) The next two bytes are BlockAlign which is worth 04 00 which states the size of the data for one full sample in bytes. One full sample is a sample that represents the value of the sample on all channels at a time.

9) The next two bytes are the value of bits per sample (BitsPerSample) that are worth 10 00 are 16 and 00 = 16 bits per sample for the right channel sample and for the left channel sample.
10) The next four bytes are 64 61 74 61 which states the ID with the value 64 = d, 61 = a, 74 = t, 61 = a with the meaning of "data" which states the sample data is digital audio.
11) Sixteen (16) next bytes are sample right channel audio samples 1 to sample 4 with values 00 00 00 00 24 17 1e f3 3c 13 3c 14 16 f9 18 f9.
12) The next sixteen (16) bytes are audio samples left channel sample 5 to sample 8 with values 34 e7 23 a6 3c f2 24 f2 11 c0 1a 0d 00 7f 11 00.
13) Continue until all audio sample data is obtained.

From the data from the audio file, data is obtained from the sample to 1 with the sign # 1 until the sample is 23 with the sign # 23 on the last block of audio sample data. For example, from the value of sample 1 audio right channel above that will be encoded are: **00 00 00 00 24 17 1e 3c 13 3c 14 16 f9 18 f9**. From the data to be encoded, a probability table can be created like Table 4.

Table 4. Probability of Wave File Data Tables

No	Value	Frequency	Probability
1	00	4	4/15=0,26
2	24	1	1/15=0,06
3	17	1	1/15=0,06
4	1e	1	1/15=0,06
5	3c	2	2/15=0,13
6	13	1	1/15=0,06
7	14	1	1/15=0,06
8	16	1	1/15=0,06
9	F9	2	2/15=0,13
10	18	1	1/15=0,06

Next will be obtained a probability range table such as Table 5.

Table 5. Table of Probability Data Wave Data Range

No	Value	Frequency	Probability	*Range*
1	00	4	4/15=0,26	0,0 ≤00< 0,26
2	24	1	1/15=0,06	0,26 ≤24< 0,32
3	17	1	1/15=0,06	0,32 ≤17< 0,38
4	1e	1	1/15=0,06	0,38 ≤1e< 0,44
5	3c	2	2/15=0,13	0,44 ≤3c< 0,63
6	13	1	1/15=0,06	0,63 ≤13< 0,69
7	14	1	1/15=0,06	0,69 ≤14< 0,75
8	16	1	1/15=0,06	0,75 ≤16< 0,81
9	F9	2	2/15=0,13	0,81 ≤f9< 0,94
10	18	1	1/15=0,06	0,94 ≤18< 1

For **data 00 00 00 00 24 17 1e 3c 13 3c 14 16 f9 18 f9** from the audio sample the arithmetic encoding process is carried out as follows:

1) **Calculation of Value 00 (0,0 ≤4< 0,26)**

Low = 0,0
High = 1,0
CR = High - Low
= 1,0 - 0,0
= 1
High_range (00) = 0,26
Low_range (00) = 0,0
Then, the following values are obtained:
High = Low + CR * High_Range (00)
= 0,0 + 1 * 0,26
= 0,26
Low = Low + CR * Low_Range (00)
= 0,0 + 1 * 0,0
= 0

2) **Calculation of Value 24 (0,26 ≤24< 0,32)**

Low(00) = 0
High(00) = 0,26
CR = High – Low
= 0,26 - 0 = 0,26
High_range (24) = 0,32
Low_range (24) = 0,26
Then, the following values are obtained:
High = low + CR* high_range (24)
= 0 + 0,26 * 0,32
= 0.0832
Low = low + CR * low_range (24)
= 0 + 0,26 * 0,26
= 0.0676

3) **Calculation of Value 17 (**0,32 ≤17< 0,38)

Low = 0,0676
High = 0,0832
CR = High - Low
= 0,0832– 0,0676
= 0.0156
High_range = 0,38
Low_range = 0,32
Then, the following values are obtained:
High = low + CR* high_range (17)
= 0,0676 + 0,0156 * 0,38
= 0.073528
Low = low + CR * low_range (17)
= 0,0676 + 0,0156 * 0,32
= 0.072592

.
.

.

10) **Calculation of Value 18 (0,94 ≤18< 1)**
Low = 0.074594625729676
High = 0.074596281582044
CR = High - Low
= 0.074596281582044 – 0.074594625729676
= 0.000001655852368
High_range = 1
Low_range = 0,94
Then, the following values are obtained:
High = low + CR* high_range
= 0.074594625729676 + 0.000001655852368 * 1
= 0.074596281582044
Low = low + CR * low_range
= 0.074594625729676 + (0.000001655852368 * 0,94)
= 0.07459618223090192

Table 6. Wave File Encoding Results Data

No	Nilai	Low	High	CR
Awal		0	1	1
1	00	0,0	0,26	1
2	24	0.0676	0.0832	0,26
3	17	0.072592	0.073528	0.0156
4	1e	0.0740968	0.0743344	0,00396
5	3c	0.074201344	0.074246488	0.0002376
6	13	0.07422978472	0.0745128376	0.000045144
7	14	0.0744250912072	0.0744250912072	0.00028305288
8	16	0.0745843084522	0.0745970458318	0.00021228966
9	F9	0.074594625729676	0.074596281582044	0.0000127373796
10	18	0.07459618223090192	0.074596281582044	0.000001655852368

From this process, the low value for the last data is the low value = 0.07459618223090192 which will be used to replace the encoding audio sample, namely the right channel sample audio value **00 00 00 00 24 17 1e 3c 13 3c 14 16 f9 18 f9**. The next sample 1 is changed to 0.074.

For the audio sample left channel calculate the low value as above. The table of probability values and audio sample symbols will be stored in text format as a data source in the decoding process for audio samples to be replaced. The results of testing compression and decompression carried out on several wave files are presented in Table 7 and Table 8.

Table 7. Testing of Wave File Compression

No	File Type	Initial File Size (Kb)	Final File Size (Kb)	Ratio (%)	Time (Detik)
1	chimes.wav	55,776	48,621	12,83	46,33
2	chord.wav	97,016	76,525	21,12	45,90
3	windows battery.wav	53,864	46,107	14,40	45,43

Table 8. Tests for Decompression of Wave Results

No	File Type	Initial File Size (Kb)	Final File Size (Kb)	Ratio (%)	Time (Detik)
1	chimes.wav	48,621	55,776	12,83	0,27
2	chord.wav	76,525	97,016	21,12	0,44
3	windows battery.wav	46,107	53,864	14,40	0,25

4. CONCLUSION

Based on the research results it can be concluded that the average wave file compression ratio is 16.12% and the average compression time is 45.89 seconds, while the average wave file decompression ratio is 16.12% and the average time is 0.32 seconds. The average decompression process time in this study is smaller when compared to the compression process because the probability value tables and audio sample symbols generated in the compression process are stored in text format as data sources in the decompression process for audio samples to be replaced. In this study testing was carried out on wave files with no limitations on file size. It is recommended that this study be developed with a variety of audio file formats such as mp3, mp4 and others.

5. REFERENCES

[1] Hazem, Kathem Qattous. (2017). Hiding Encrypted Data Into Audio File. *IJCSNS International Journal of Computer Science and Network Security, 17(6),* 162-170.

[2] Howard, Paul G & Jeffrey Scott Vitter. (1992). Analysis Of Arithmetic Coding For Data Compression. *Information Processing & Management, 28(6),*749-763.

[3] Iwan, Binanto. (2010). *Multimedia Basic Digital Theory & Development.* Yogyakarta: Andi Offset.

[4] Jawahir, Ahmad., & Haviluddin. (2015). An Audio Encryption Using Transposition *Method. International Journal of Advances in Intelligent Informatics, 1(2),* 94-106.

[5] K. Sayood. (1996). *Introduction To Data Compression*. Virginia Polytechnic University: Morgan Kaufmann Publishers Inc.

[6] Maan, Anmol Jyot. (2013). Analysis and Comparison of Algorithms For Lossless Data Compression. *International Journal of Information and Computation Technology*, *3(3),* 139-146.

[7] Said, Amir. (2004). *Comparative Analysis of Arithmetic Coding Computational Complexity*. Imaging Systems Laboratory. HP Laboratories Palo Alto.

[8] Salomon, D. A (2012). *Guide to Data Compression Methods*, Springer.

[9] Silitonga, Parasian D.P. et.al. (2018). Wave File Encryption using Huffman Compression and Serpent Algorithm.. *International Journal of Computer Trends and Technology (IJCTT), 64(1),* 2231-2803.

[10] Tamimi, A.A & A. M. Abdalla. (2104). An Audio Shuffle -Encryption Algorithm. *The World Congress on Engineering and Computer Science 2014 WCECS.* San Francisco.