Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique

Omar Adil Mahdi [1], Mazin Abed Mohammed [2] and Ahmed Jasim Mohamed [2]

[1] College of Education (Ibn - Haitham),
University of Baghdad,
Baghdad, Iraq

[2] Planning &Follow Up Department,
University of Anbar,
Ramadi, Anbar, Iraq

[2] Center of Computer and Information Technology,
, University of Anbar,
Ramadi, Anbar, Iraq

Abstract

Compression is the reduction in size of data in order to save space or transmission time. For data transmission, compression can be performed on just the data content or on the entire transmission unit (including header data) depending on a number of factors. In this study, we considered the application of an audio compression method by using text coding where audio compression represented via convert audio file to text file for reducing the time to data transfer by communication channel. Approach: we proposed two coding methods are applied to optimizing the solution by using CFG. Results: we test our application by using 4-bit coding algorithm the results of this method show not satisfy then we proposed a new approach to compress audio files, by converting an audio file in to a text file and then compressing the new text file by using the common compression techniques is 6-bit coding algorithm its used to convert a digitized Audio file into a text file it is show that a good compression ratio between 15-35 %.

Keywords: *Data Compression, Audio Compression, 4-bit coding algorithm****,*** *6-bit coding algorithm****.***

1. Introduction

Nowadays, in computer science and information theory, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying marginally important information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although its formal name is source coding (coding done at the source of the data, before it is stored or transmitted [1].

Compression is useful because it helps reduce resources usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to be used, this extra processing imposes computational or other costs through decompression, this situation is far from being a free lunch. Data compression is subject to a space-time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involve trade-offs among various factors, including the degree of compression, the amount of distortion introduced (e.g., when using lossy data compression), and the computational resources required to compress and uncompressing the data. Figure 1 show compression process.

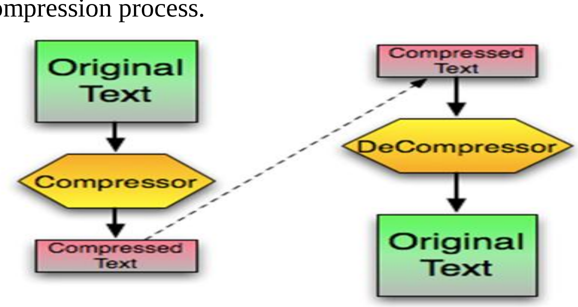

Fig. 1 show compression process

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

The basic idea behind any compression-for audio and image- is to find a way to represents data that take up less space. The most important advantage of data compression is reducing the time for data transfer via communication channels [2].Many algorithms used to improve compression method in the recent years, but two families of these algorithms exist in compression. When the information can be exactly recovered from the bits, the source coding or compression is called lossless; otherwise, it is called lossy. To achieve higher compression ratios, lossy algorithms remove information from that comes close to the original or that is not perceptible.

Lossless algorithms are typically used for text or executable codes, while lossy are used for images and audio where a little bit of loss in resolutions often undetectable, or at least acceptable. Lossy is used in an abstract sense; it does not mean random loss in samples, but instead means loss of a quantity such as a frequency component, or perhaps loss of noise [3].The storage requirements of sound are smaller than those of images or movies, but bigger than those of text, this is why audio compression has become important and has been subject of much research and experimentation throughout 1990s. The first well known method for effectively coding symbols is now known as Shannon-Fano coding .Claude Shannon at Bell Labs and R.M. Fano at MIT developed this method nearly simultaneously. It depends on simply knowing the probability of each symbol's appearance in a message.

Given the probabilities, a table of codes could be constructed that has several important properties:

- Different codes have different numbers of bits.
- Codes for symbols with low probabilities have more bits, and codes for symbols with high probabilities have fewer bits.
- Through the codes are of different bit lengths, they can be decoded.

Huffman coding shares character tics of Shannon-Fano coding. It creates variable-length codes that are an integral number of bits. Symbols with higher probabilities get shorter codes. Huffman codes have the unique prefix attribute, which means they can be correctly decoded despite being variable length .decoding a stream of Huffman codes is generally done by a binary decoder tree [4].

Several quantities are commonly used to express the performance of compression method. The compression ratio is defined as:

$$\text{Compression ratio} = \frac{\text{Size of the output stream}}{\text{Size of the input stream}}$$

The inverse of the compression ratio is called the compression factor

$$\text{Compression factor} = \frac{\text{Size of the input stream}}{\text{Size of the output stream}}$$

Audio data compression, as distinguished from dynamic range compression, has the potential to reduce the transmission bandwidth and storage requirements of audio data. Audio compression algorithms are implemented in software as audio codecs. Lossy audio compression algorithms provide higher compression at the cost of fidelity, are used in numerous audio applications. These algorithms almost all rely on psychoacoustics to eliminate less audible or meaningful sounds, thereby reducing the space required to store or transmit them. In both lossy and lossless compression, information redundancy is reduced, using methods such as coding, pattern recognition and linear prediction to reduce the amount of information used to represent the uncompressed data.

The acceptable trade-off between loss of audio quality and transmission or storage size depends upon the application. For example, one 640MB compact disc (CD) holds approximately one hour of uncompressed high fidelity music, less than 2 hours of music compressed losslessly, or 7 hours of music compressed in the MP3 format at a medium bit rate. A digital sound recorder can typically store around 200 hours of clearly intelligible speech in 640MB.[6] Lossless audio compression produces a representation of digital data that decompresses to an exact digital duplicate of the original audio stream, unlike playback from lossy compression techniques such as Vorbis and MP3. Compression ratios are around 50–60% of original size,[7] similar to those for generic lossless data compression. Lossy compression depends upon the quality required, but typically yields files of 5 to 20% of the size of the uncompressed original.[8] Lossless compression is unable to attain high compression ratios due to the complexity of wave forms and the rapid changes in sound forms. Codecs like FLAC, Shorten and TTA use

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

linear prediction to estimate the spectrum of the signal. Many of these algorithms used convolution with the filter [-1 1] to slightly whiten or flatten the spectrum, thereby allowing traditional lossless compression to work more efficiently. The process is reversed upon decompression when audio files are to be processed, either by further compression or for editing, it is desirable to work from an unchanged original (uncompressed or losslessly compressed). Processing of a lossiy compressed file for some purpose usually produces a final result inferior to creation of the same compressed file from an uncompressed original. In addition to sound editing or mixing, lossless audio compression is often used for archival storage, or as master copies.

2. Research Objectives

The main objective of the research is to reduction in size of data in order to save space or transmission time. The aim of the proposed techniques method is to provide quick and optimal solution to data compression methods. Additionally, for training purposes, it helps in reducing the knowledge gap between different individuals in data compression algorithms. The specific objectives of their search are as follows:

- To investigate the related works on data compression methods domains.
- To design appropriate representation architecture to the 4-bit coding and 6- bit coding proposed.
- To develop a CFG that supports the implementation of the proposed system's functionality.
- To test and validate the system's performance.

3. Research Theory

The theoretical background of compression is provided by information theory (which is closely related to algorithmic information theory) for lossless compression, and by rate distortion theory for lossy compression. The idea of data compression is deeply connected with statistical inference and to design appropriate representation architecture to the 4-bit coding and 6- bit coding proposed and to test and validate these algorithms performance.

4. Conventional Audio Compression Methods

Many techniques applied to optimizing data compression for reducing time for data transfer by communication channel such as RLE, statistical, and dictionary-based, can be used to losslessly compress sound files, but the results depend heavily on the specific sound. Some sounds may compress well under RLE but not under a statistical method. Other sounds may lend themselves to statistical compression but may expand when processed by a dictionary method. Here is how sounds respond to each of the three classes of compression methods. RLE may work well when the sound contains long runs of identical samples.

Statistical methods assign variable-size codes to the samples according to their frequency of occurrence. Dictionary-based methods expect to find the same phrases again in the date. This happens with text, where certain string may repeat often. Lossy sound compression methods discard data to which the human ear is not sensitive. This is similar to lossy image compression, where data to which the human eye is not sensitive is discarded. In both cases we use the fact that the original information (image or sound) is analog and has already lost some quality when digitized. Losing some more data, if done carefully, may not significantly affect the played-back sound, and may therefore be indistinguishable from the original [6].
Because there are so many of data people want to compress, there has been a lot of work over the years on lossless compression techniques. These are general-purpose algorithms that look for patterns in binary data. If they can locate patterns, they can compress the data by replacing blocks of data with codes indicating the pattern. The decompress or (which knows about the same types of patterns) can then undo this process. These algorithms are lossless because the decompressed data will be bit-for-bit identical to the original. This is an important feature for general-purpose algorithms that might be used for any type of data.

Many applications– including streaming Internet sound involve transferring sound data and playing it as it is received. Frequently, there are harsh restrictions on how quickly the data can be transferred [1].

5. Proposed Prototype Modeling

Our prototype modeling deferent other modeling applied to reducing time for data transfer by communication channel such as RLE, statistical and dictionary-based to solving these problems therefore we are used two technics are 4-bit coding and 6- bit coding Compression methods are applied on wave file format to produce a compressed file. The proposed schema starts with wave a file .This file is converted into a text file and then coding the new text file using proposed 6-bits characters table then compressed by using the common compression techniques via WinZip and

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

power archive one at a time. Figure 2 illustrates coding compression method**.**

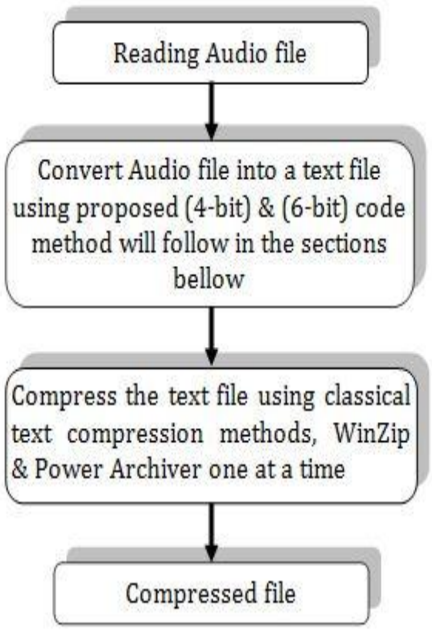

Fig. 2 illustrates the block diagram for the proposed coding compression system.

After finishing coding compression system convert compressed file to extract the text file at this time will be read it and convert the text file from the proposed code in to ASCII code all these levels are decoding compression system. Figure 3 illustrates decoding compression method**.**

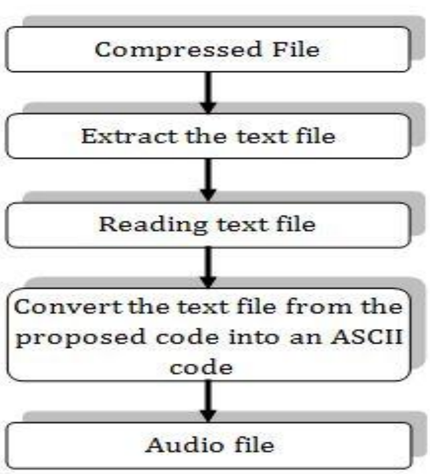

Fig. 3 illustrates the block diagram for the proposed decoding compression system.

5.1 Proposed 4-bit Code Method Algorithm

The proposed 4- bit code method presented as in following procedure:
Input: Audio file (wave or mp3 format)
Output: Compressed file
Begin
Open Audio file for Input
Open Text file for Output
While not eof (Audio) ***do***
Begin
Read one byte from audio file
Split this byte into two parts
Search in table(1) to find the corresponded character for each part
Save these two characters into the text file
End
Close Audio file
Close text file
Compress using classical text compression methods
END.

Table (1) Proposed 4 bit Coding Method

Character	Bits
A	0000
B	0001
C	0010
D	0011
E	0100
F	0101
G	0110
H	0111
I	1000
J	1001
K	1010
L	1011
M	1100
N	1101
O	1110
P	1111

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

5.2 Proposed 6-bit Code Method Algorithm

The proposed 6-bit code method presented as follow:

INPUT: Audio file (wave or mp3 format)
OUTPUT: compress file
BEGIN
OPEN Audio file for input
OPEN text file for output
While not eof (Audio) ***do***
Begin
Read three bytes from Audio file
Split these bytes into four parts each 6-bit
Search in table (3.2) to find the corresponded character for (decimal value for each part)
save these three characters in the text file
End
CLOSE Audio file
CLOSE text file
Compress the text file using classical text compression methods
END.

Table (2) Proposed 6 bit Coding Method

Character	Bits	Character	Bits	Character	Bits
A	000000	a	011010	0	110100
B	000001	b	011011	1	110101
C	000010	c	011100	2	110110
D	000011	d	011101	3	110111
E	000100	e	011110	4	111000
F	000101	f	011111	5	111001
G	000110	g	100000	6	111010
H	000111	h	100001	7	111011
I	001000	i	100010	8	111100
J	001001	j	100011	9	111101
K	001010	k	100100	,	111110
L	001011	l	100101	;	111111
M	001100	m	100110		
N	001101	n	100111		
O	001110	o	101000		
P	001111	p	101001		
Q	010000	q	101010		
R	010001	r	101011		
S	010010	s	101100		
T	010011	t	101101		
U	010100	u	101110		
V	010101	v	101111		
W	010110	w	110000		
X	010111	x	110001		
Y	011000	y	110010		

6. Implementation and Results

In our work, we try to deal with an optimal reduction and benefit from simple and clever ideas to solve the problem. If the user did not succeed in solving the problem, the application gives the user advice on what he/she should do. The implementation of using the (4 bit) coding method, it is shown that the results are not satisfactory because the duplication of files size. Therefore, our experiments relied on the second method which converts the ASCII (8 bits) to proposal code (6 bits).

The following figures of our experiments show the compression ratio. It shows a graphical representation of the proposed schema compared to the regular compression techniques. The following figure show the comparison results of WinZip on wave files with & without our proposal coding method see figure 4.

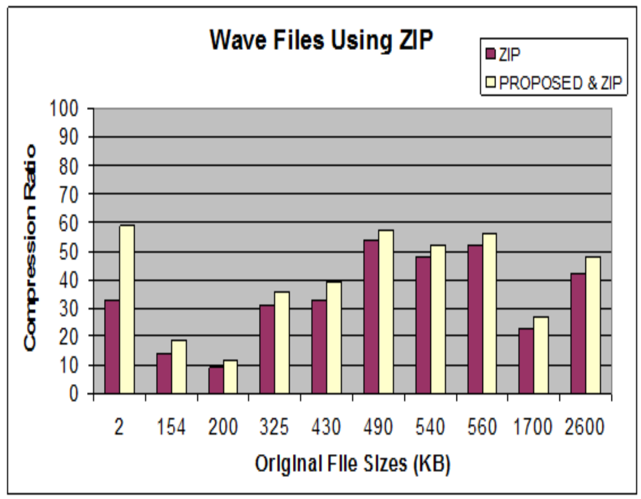

Fig.4 show wave files Using ZIP.

Figure 5 show the comparison results of Power Archiver .Rar on wave files with & without our proposed coding method see figure 5.

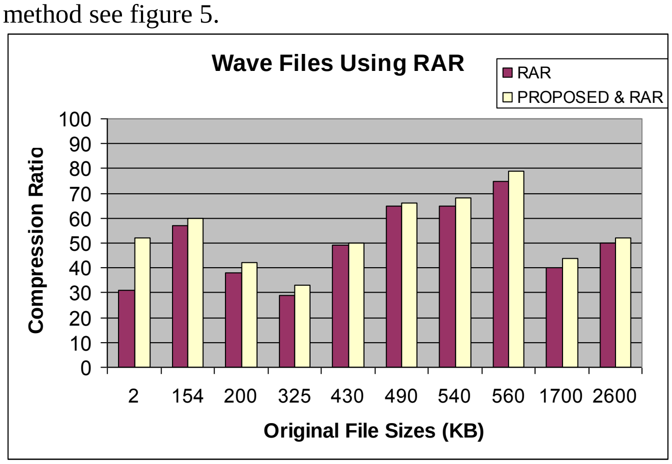

Fig. 5 show wave files Using RAR

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012
ISSN (Online): 1694-0814
www.IJCSI.org

Figure 6 illustrates the comparison results of both the compression rate using Power Archive .Rar on mp3 files and the compression rate using proposal with Power Archiver, Rar on the text file see figure 6.

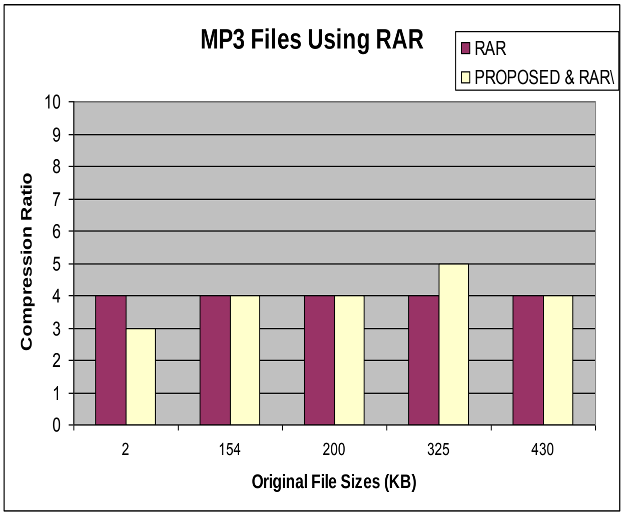

Fig. 6 show MP3 files Using RAR

Figure 7 illustrates the comparison results of both the compression rate Using Win Zip on mp3 files and the compression rate using proposal with Win Zip on the text file see figure 7.

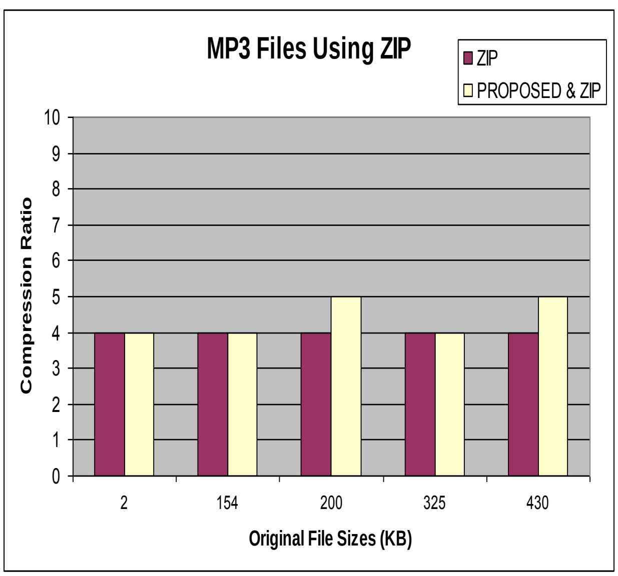

Fig.7 show MP3 files Using ZIP

Figure 8 illustrates the average of compression ratio on WAVE files with WinZip (.zip) & Power Archiver (.RAR) before and after using the proposed method.

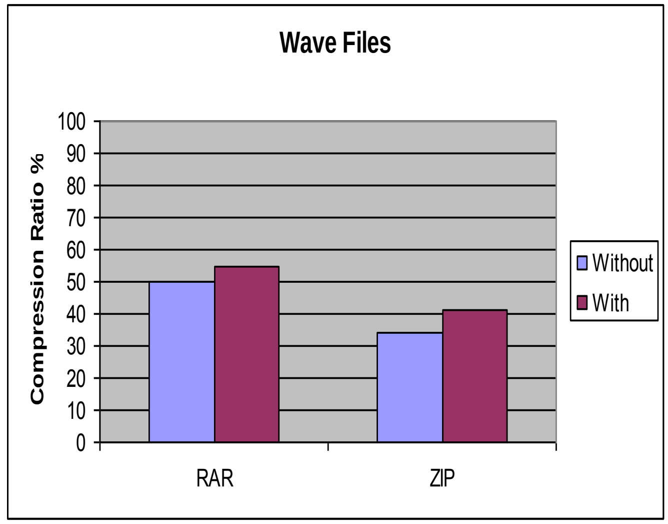

Fig. 8 show Wave Files Average

Figure 9 illustrates the average of compression ratio on MP3 files with WinZip (.zip) & Power Archiver (RAR) before and after using the proposed method.

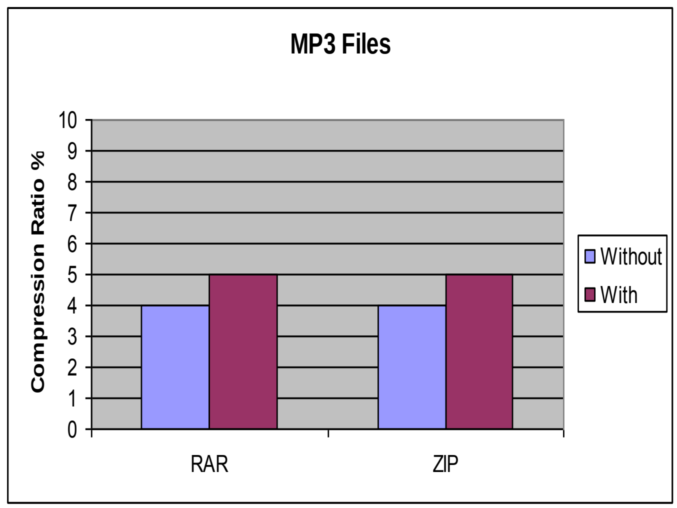

Fig. 9 show MP3 Files Average

7. Conclusion

In this paper, lossy compression methods are used on sound files (speech & audio) in order to obtain good compression ratio, it is obvious that the amount of information (audio) loss will have a minor effect on the outcome. In our proposed method we realized that there is some loss of information since the file we are using is a digital file obtained from an analog file. Further improvement to the system domain knowledge specifications is required to enhance domain knowledge representation. Furthermore, adopting another AI

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

technique to work in system rules revision to add more effectiveness to the diagnosis process will be considered. Our proposed method should not lose any information (audio) the following conclusion are drawn:

- The proposed coding methods (4-bit) and (6-bit) to convert a digitized audio file into a text file and according to our experiments the (6-bit) coding method was better than (4-bit) coding method.

- The Power Archiver is applied to a WAVE files with our proposed method good compression ratio 55% are obtained, while the average compression ratio of a direct compression of the wave files is 50%.

- The Win Zip is applied to a WAVE files with our proposed method good results 42%, are obtained while the average compression ratio of the WAVE files 34% without the proposed method.

- The Win Zip is applied on MP3 files with our proposed method we also get good results 5%, while the maximum compression ratio of the direct compression of the MP3 files is 4%.

It's clear that the results show that the Power Archive gets better results than Win Zip. At the same time both of them give better results than the direct compression of the audio files (WAVE & MP3). Generally, WAVE files compression gives better results than MP3 files.

8. References

[1] Tim Kientzle "A Programmer's Guide to Sound" An Imprint of Addison Wesley Longman Inc., 1998.
[2] J. A. Seidler, "Information System and Data Compression", University of Salzburg, Boston / Dordreocht / London 1997.
[3] D.L. Donoho, M.Vetterli, R.A. DeVore, and I. Daubechies, "Data Compression and Harmonic Analysis"gd,, July 1998.
[4] Nelson, Mark, and Jean-Loup Gailly "The Data Compression Book" 2nd Edition, New York, M&T Books, 1996.
[5] Storer, James A., (ed.), " Data Compression" Conference, Los Alamitos, CA, IEEE Computer Society Press.1997.
[6] David Salomon, "Data Compression the Complete Reference" 2nd Edition, Spring Verlag, New York, 2000.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.