Audio Compression Algorithm using Discrete Cosine Transform (DCT) and Lempel-Ziv-Welch (LZW) Encoding Method

Sankalp Shukla, Maniram Ahirwar, Ritu Gupta, Sarthak Jain, Dheeraj Singh Rajput
ECE Department
IGEC, Sagar
sankalp.shukla@mp.gov.in, maniram.ahirwar@mp.gov.in, ritu.gupta@mp.gov.in, jsarthak@outlook.com, dheeraj.rajput@mp.gov.in

***Abstract* – This paper proposes a new approach to Audio Compression that incorporates lossless text compression algorithm. The purpose of Audio Compression is to reduce the amount of data required to represent the digital audio by removing redundant data. In the present work Discrete Cosine Transform and lossless text compression method (Lempel-Ziv-Welch method) based Audio Compression algorithm has been proposed which also includes audio normalization, scalar quantization and encoding. The LZW dictionary obtained after compression is used for transmission and storage. The performance of the proposed algorithm is analyzed using various audio specimens of distinct size with distinct audio signal parameters. The compression performance is assessed using Peak Signal to Noise Ratio (PSNR) and Compression Ratio (CR). The audio specimen outcomes indicate the propitious performance of proposed algorithm. The compression ratio can be enhanced by changing various parameters of the system.**

Keywords: Audio Compression, Lempel-Ziv-Welch (LZW) Coding, Discrete Cosine Transform (DCT), Scalar Quantization, Audio Masking.

I. INTRODUCTION

Technological advancement in computers has inevitably resulted in demand for good quality audio data. The most popular audio coders use two techniques viz. sub-band coding and transform coding. Sub-band coding divides the signal into a number of sub-bands by making use of band-pass filter. Transform coding employs mathematical transformations such as FFT and DCT. The MP3 standard specifically exploits Modified Discrete Cosine Transform (MDCT) for compressing audio data. Scientists have examined characterization of Modified Discrete Cosine Transform (MDCT) in perceptual audio coding and error concealment with reference to frequency analysis [1,2].

DCTs are important in diverse applications in science and engineering from lossy audio and image compression to the numerical solution of partial differential equations through spectral methods. DCT is preferred because of its energy compaction property and application in perceptual audio coding [3].

The LZW is a widespread lossless data compression algorithm. Abraham Lempel, Jacob Ziv and Terry Welch created a universal data compression algorithm in 1984 as a modified version of the LZ78 algorithm published by Lempel and Ziv in 1978 [4].

In the proposed algorithm the audio samples are first normalized to [-1, 1] range and grouped into blocks of fixed length. This data is processed using Discrete Cosine Transform and then non-uniform scalar quantization is employed to quantize the transform coefficients. The quantized values are subsequently converted into ASCII characters which are compressed using the Lempel-Ziv-Welch coding method to reduce the size of the quantized data. This is done by transforming ASCII characters into dictionary indices which are encoded as bytes for transmission or storage.

II. AUDIO COMPRESSION SYSTEM

A salient feature of audio signals is the presence of unwanted data that links the subsequent specimens. The compression technique tends to eliminate this unwanted data which results in its de-correlation. Most of the audio compression systems contain three basic steps to attain audio compression. First step involves application of an appropriate transform. Second step reduces the bits required for encoding by quantizing the resulting transform coefficients; here the quantized data holds errors. Third step encodes the quantized values using an appropriate variable or fixed length coding scheme. The layout of the proposed system is shown in Fig. 1. Following sections describe modules of the system.

A. Pre-Processing

In pre-processing stage uncompressed audio file is obtained and the audio samples are read by the system. The system first analyzes the audio samples and normalizes the samples to the range [-1,1] by the equation

$$X_{new} = \frac{X_i}{\max(X)} \tag{1}$$

where X is the set of all the samples in the audio file, X_{new} denotes the new value of the current sample in the audio file, X_i is the current sample being operated. Then the audio samples are converted into blocks for the next stage, i.e. DCT.

B. Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is one of the most frequently used transformations for compression. It uses a combination of cosine functions with different frequencies to represent a finite sequence of data. A prominent characteristic of DCT, and in particular DCT-II, is strong "energy compaction" which makes it suitable for its use in lossy compression (signal & image processing). In audio signals, most of the information tends to be concentrated on a few low-frequency components. DCT is applied on audio sample blocks obtained after pre-processing where it makes use of aforementioned characteristic of audio signal. The general equation of a one-dimensional (N data points) DCT is defined by the equation given below [5]:

$$F(u) = \alpha(u) \sum_{i=0}^{N-1} \cos\left[\frac{\pi . u}{2N}\,(2i+1)\right] f(i) \qquad (2)$$

where $f(0,1,\dots N-1)$ denotes discrete data sequence of signal f, N denotes number of samples, $F(0,1,\dots,N-1)$ denotes cosine transform coefficients, and

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}\ , \text{for } u = 0 \\ \sqrt{\frac{2}{N}}, \text{otherwise} \end{cases} \qquad (3)$$

The inverse 1-D DCT equation is:

$$f(i) = \alpha(u) \sum_{u=0}^{N-1} \cos\left[\frac{\pi . u}{2N}\,(2i+1)\right] F(u) \qquad (4)$$

The first transform coefficient $F(0)$ is called the DC coefficient of signal which indicates average value of the discrete sequence while remaining coefficients are called AC coefficients [6].

C. Quantization

Quantization maps the input values from a continuous set to the output values in a smaller countable set. It is an integral part of all digital signal processing systems and an essential step for data compression. The transform coefficients obtained on application of DCT are approximately mapped to integer values of finite length with the help of quantization, thereby reducing the number of bits required to store an integer value [7].

In a transform-based coding, allocation of bits to each frequency component is done by the quantizer. Allocation of more number of bits to each sample increases storage space required but introduces less error in outcome. On the other hand, allocation of fewer bits to each sample tends to increase noise but reduces storage space required. Thus, the quantizer must be designed in such a way that it not only results in less noise but also reduces storage space required [8].

D. Entropy Coding

Entropy coding is a type of lossless coding which helps in minimizing the number of bits required for encoding purposes. It creates a unique prefix-free code and maps it to each unique symbol present in the quantized input sequence. Statistical based coding methods tend to remove data that occurs redundantly in the input sequence. In proposed system the Lempel-Ziv-Welch (LZW) encoding method removes long runs of quantized coefficients resulting in compression.

E. LZW Coding

LZW is a universally used lossless data compression algorithm which was introduced by Abraham Lempel, Jacob Ziv and Terry Welch. The quantized transform coefficients of variable length, which constitute the input source symbols, are mapped to the code words of fixed length. A salient feature of LZW coding is that apriori knowledge of the probabilistic occurrence of input symbols is not required for encoding purpose. The implementation of this algorithm is simple which has the

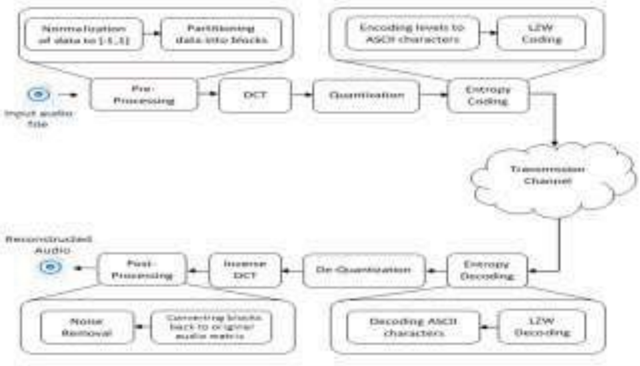

Fig. 1 Audio Compression System.

property of substantial compression throughputs in hardware implementations. This compression algorithm is widely used in the GIF image format and in the Unix file compression utility [9].

III. PROPOSED SYSTEM

The proposed system embodies two main sections - Encoding Section and Decoding Section.

A. Encoding Section

The encoding section incorporates the following steps:

1. The uncompressed audio file is loaded for processing. It involves extracting the basic file and signal specifications by reading the header data. This information contains number of samples, number of channels and sampling resolution. Subsequently these audio samples are stored as a group of unsigned bytes.

2. The stored audio samples are normalized to the range [-1,1] using the following equation:

$$W_n(i) = \frac{W(i)-128}{128} \qquad (5)$$

where, $W(i)$ is the value of i^{th} sample.

3. The group of data is divided into a number of non-overlapping equal sized units, excluding the last unit which may have a smaller size. Encoding of audio samples gives rise to less compaction (i.e. less compression is obtained) if DCT is applied without signal partitioning. This is due to the fact that the characteristics of audio signal change with time. The number of partitions is computed as follows:

$$N_{blocks} = \frac{WavSize}{S_{blocks}} \qquad (6)$$

$$\text{LastBlockSize} = \text{WavSize} \bmod B_{size} \qquad (7)$$

where,

N_{blocks} = number of units derived from the primary wave.

S_{block} = predefined size for each unit.

WavSize = number of samples of the stored audio data.

LastBlockSize = size of the last unit of the primary wave.

4. The transform coefficients are generated by applying DCT on each audio block.
5. For entropy encoding the transform coefficients are represented using minimum number of bits. This is accomplished by employing repetitive scalar quantization to the DCT coefficients of each unit. The encoded values are in the interval [0,255]. Hence, an 8-bit encoding is performed on the DCT coefficients of each block. These levels are then converted into ASCII characters using type conversion and the whole block of data is converted into a single string of ASCII characters.
6. In the proposed system the Lempel-Ziv-Welch (LZW) coding is used to eliminate the inherent redundancy in data sequences. LZW coding can effectively handle the text compression which is obtained after entropy encoding and ASCII character representation of the data. The LZW coding is conceptually very simple. It begins with the construction of dictionary incorporating the source symbols to be encoded and initializing the ASCII characters with the first 256 dictionary indices (i.e. 1,2,3…..,256). The compression performance is influenced by size of the dictionary. If it is too small or too large, it results in poor compression.

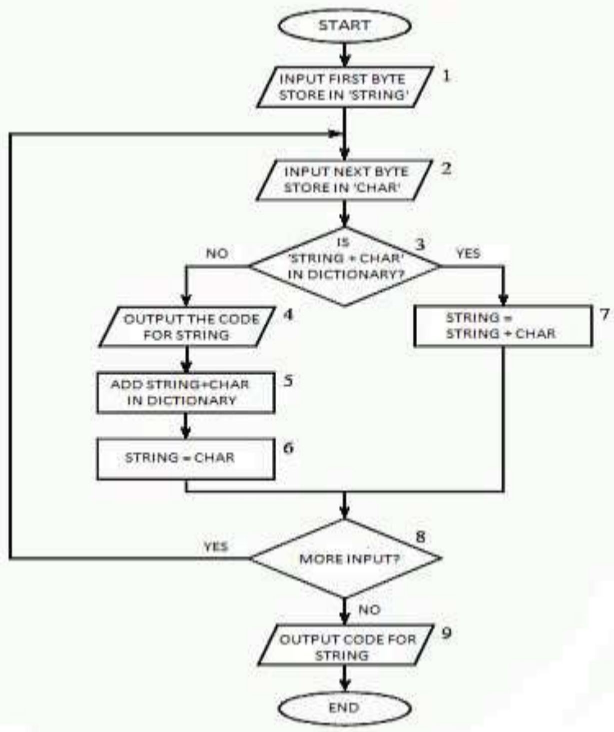

Fig. 2 LZW Encoding Process.

A salient feature of LZW coding is the creation of coding dictionary or code book during data encoding process. It is noteworthy that an identical decompression dictionary is constructed as encoded data stream is decoded by LZW decoder. In some applications, a dictionary handling mechanism needs to be incorporated whenever the size of dictionary exceeds a certain level. In such cases it is either reinitialized or flushed. Alternatively, the least used dictionary entries can be tracked and replaced when necessary. Fig. 2 shows the flow chart of the LZW encoding process [9].

7. The LZW dictionary indices obtained after encoding are encoded as byte array and transmitted or stored as the compressed audio file.

B. Decoding Section

The decoding section incorporates the following steps:

1. Loading the compressed file – This process involves reading the compressed file to extract encoded byte stream and convert it back to integers for decoding process.
2. LZW decoding – It converts the integer indices back to uncompressed data. The basic flowchart of the LZW decoder is shown in Fig. 3. The LZW decoder constructs an identical dictionary during decompression. It begins with the default dictionary

which contains 256 initialized ASCII characters. This dictionary is continuously modified with the extraction of incoming byte stream. Decoding is accomplished by translating the encoded byte stream through the dictionary being constructed.

3. After decoding process, the ASCII characters obtained as a result of decompression are converted back to integers by type conversion. Then these integers are converted to unsigned data using de-quantization process.
4. The data obtained as a result of decompression and de-quantization is a single array of elements. The group of data is divided into a number of non-overlapping equal sized units, excluding the last unit which may have a smaller size. This division is performed in a way similar to the encoding section.
5. The application of Inverse DCT on each block results in a set of inverse transform coefficients.

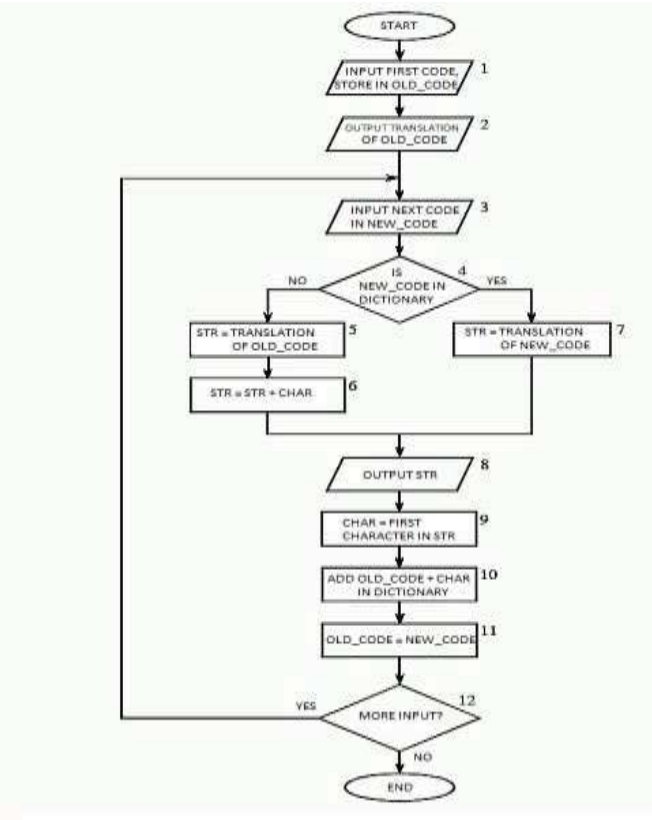

Fig. 3 LZW Decoding Process.

6. The inverse DCT coefficients are the actual audio samples but these are not in order. So, these elements are converted to an array which is similar to the original audio file and depends on the number of channels in it.
7. In the final step for post processing, the audio coefficients obtained are filtered and any quantization noise involved is reduced and hence the original audio is obtained.

IV. TEST RESULTS

Compression gain (Compression Ratio, CR) and Fidelity criteria (either root-mean-square error or PSNR) are the major factors which form the basis to evaluate the performance of audio compression algorithms [5]:

$$\mathrm{CR} = \frac{\text{Original audio file size}}{\text{Compressed audio file size}} \tag{8}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=0}^{n-1}(x(i) - y(i))^2 \tag{9}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{R^2}{\mathrm{MSE}}\right) \tag{10}$$

$$\mathrm{SNR} = 10\log_{10}\left(\frac{Ps}{Pn}\right) \tag{11}$$

where, $x(i)$ denotes the i^{th} element of primary audio data, $y(i)$ denotes the i^{th} element of recovered audio data, n denotes number of audio samples and R denotes dynamic range of audio samples under consideration. The significance of proposed algorithm lies in the fact that the size of audio file is reduced considerably while preserving an acceptable audio quality. The performance of proposed audio compression algorithm has been evaluated by considering numerous parameters. MATLAB has been used for implementation of proposed algorithm and for development of additional programs useful for the testing procedure.

The features of the original audio files used for the experimental procedure are shown in TABLE I. The results after the execution of the various steps involved in the compression system are shown in TABLE II. TABLE III shows the comparison of proposed algorithm with the most widely used MP3 algorithm. It is evident from this comparison that the proposed algorithm results in a better compression of one of the test signals while improved SNR for all the test signals.

Furthermore the SNR of MP3 algorithm ranges from 25-30 dB while the proposed algorithm yields a much higher SNR ranging from 40-60 dB for the test signals, thus, resulting in an enhanced audio signal quality. The inclusion of Modified Discrete Cosine Transform along with Audio Masking in the proposed algorithm will boost the audio signal quality even further.

Table I. Test Signal Attributes

Attribute	**Audio Sample**		
	Test1.wav	***Test2.wav***	***Test3.winav***
Sampling Rate (KHz)	22	22	22
Sampling Resolution (bits per sample)	8	8	8
Size (KB)	480	711	547

Table II. Compressed File Attributes

Attribute	Audio Sample		
	Test1.wav	***Test2.wav***	***Test3.wav***
Size (KB)	112	108	68
SNR (dB)	57.9575	46.3997	39.6903
PSNR (dB)	110.5491	104.4904	102.5107

Fig. 4(a), Fig. 4(b) and Fig. 4(c) represent the waveform patterns of test audio signal samples.

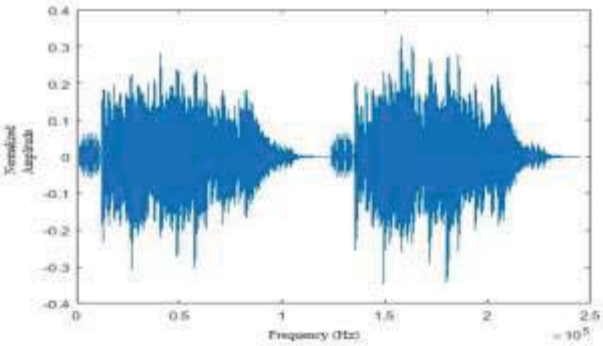

Fig. 4(a) Test1.wav

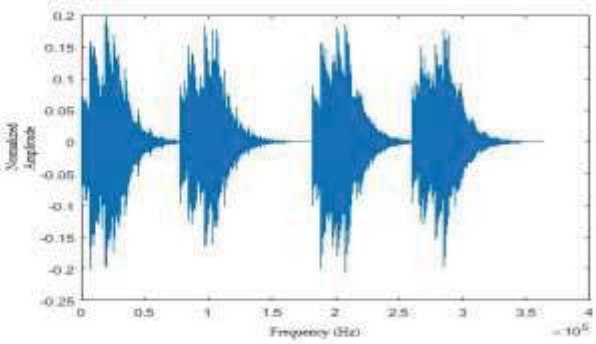

Fig. 4(b) Test2.wav

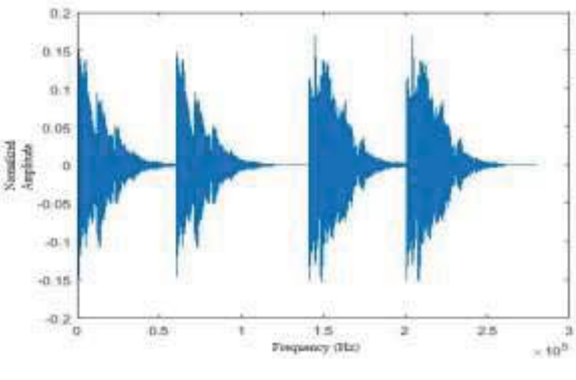

Fig. 4(c) Test3.wav

V. CONCLUSIONS

With the rapid growth of computer industry and internet, audio and other file compression techniques are required to reduce the transmission bandwidth and the storage space. This paper aims to propose an alternate technique for the compression of audio files, wherein the uncompressed audio file is compressed using the Discrete Cosine Transform and LZW coding method. This alternate solution, when tested on various uncompressed audio files with the required hardware and software, provided acceptable and promising test results when compared to the existing methods for the compression of audio files. Hence this approach can prove to be a potentially prudent solution for audio compression applications.

Table III. Comparison with MP3

File Size (KB)	Audio Sample		
	Test1.wav	***Test2.wav***	***Test3.wav***
Result	112	108	68
MP3	90	91	135

VI. FUTURE SCOPE

The existing MP3 compression uses Modified Discrete Cosine Transform and Audio Masking while the proposed algorithm uses DCT and LZW as major tools to reduce audio file size. The proposed algorithm can be further improved by incorporating the techniques used in MP3 compression algorithm. The use of Audio Masking will also result in an improved SNR.

REFERENCES

[1] J.S. Jacaba, “Audio Compression using Modified Discrete Cosine Transform the MP3 Coding Standard”, B.Sc. research paper, Philippines University, October 2001.

[2] Y.Wang and M. Vilermo, “The Modified Discrete Cosine Transform: Its Implications for Audio Coding and Error Concealment”, AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, Finland, 2002

[3] M. V. Patil, A. Gupta, A. Varma, S. Salil, "Audio and Speech Compression Using DCT and DWT Techniques", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, Issue 5, Pp. 1712-1719, May 2013.

[4] Ziv, Lempel, A. (1978). "Compression of individual sequences via variable-rate coding", IEEE Transactions on Information Theory. doi:10.1109/TIT.1978.1055934.

[5] D. Solomon “Data Compression: The Complete Reference” Springer; New York, 2004.

[6] R.G. Moreno A., M.M. Garcia, "DCT-Compressive Sampling of Frequency-Sparse Audio Signals", Proceedings of the World Congress on Engineering 2011, Vol. II WCE 2011, London, U.K, July 6 - 8, 2011.

[7] Allan V. Oppenheim, Ronald W. Schafer, Digital Signal Processing, Pearson Publication, 2015, ISBN 978-93-325-5033-9.

[8] B. Cavagnolo, J. Bier, "Introduction to Digital Audio Compression", Berkeley Design Technology, Berkeley, CA, June 2000

[9] Rafael C. Gonzales, Richard E. Woods, Digital Image Processing, Pearson Publication, 2016, ISBN 978-93-325-7032-0.