Compression of Audio Using Transform Coding

Razi J. Al-azawi[1] and Zainab T. Drweesh [2]
[1] Department of Laser and Optoelectronics, University of Technology, Baghdad, Iraq
[2] Department of Computer Science, College of Science University of Baghdad,Baghdad, Iraq
Email: 140009@uotechnology.edu.iq; myselfproudtobe@gmail.com

Abstract —In the databases application like storage and transmission, compression of audio is very necessary [1]. The problem of decrease the data's amount that needed to represent digital audio addressed by audio compression. It is applied for decreasing the redundancy by avoiding the unimportant duplicated data [2]. In this paper, we have included a review on mechanisms of audio compression using diverse transform coding techniques. This paper aims to concentrate on the advantage of transform coding comparing with today's techniques and to analyze the implementing of compression technique on audio using diverse transform coding. By using transform coding algorithm, many works have been doing to completely remove the noise from the audio or to reduce it, and have gained many good results from the work that has been researched [1].

Index Terms—Audio, compression, quantization, lossy and lossless compression, transforms coding, wavelet transform, DCT coding

I. Introduction

The sound can be represented by electrical form called Audio. Audio is sound within the range of human hearing. The human ear can recognize frequency ranges from 20 Hz to 20 kHz [3]. There are two reasons make audio compression is popular:

1. People's aversion of throw anything away and love them to accumulate data, not important how big a storage device one has, sooner or later it will be reached to overflow, data compression appear useful because it delays this fact.
2. People's hatred to wait a long time for transfer data. When we waiting for a file to download or for a web page to load, we feel that anything longer than a few seconds is long waiting time [4].

The major motivation behind development of speech/audio compression systems is to reduce the number of bits needed to represent an audio signal with the aim of minimizing memory storage costs and transmission bandwidth requirements. The basic way of audio compression is depend on removing signal redundancy while preserving the clearness of the signal [5].

The most popular audio coders are depending on using one of the two techniques (sub-band coding and transform coding). Transform coding uses a mathematical transformation like Discrete Cosine Transform (DCT) and fast Fourier transform (FFT), Sub-band coding divides signal into a number of sub-bands, using band-pass filter [2].

In this paper we focus on Run length Encoding (RLE), Transform Coding and Shift Coding, DCT and DWT, we made an attempt to discuss on these lossy and lossless algorithms [6].

II. Techniques for Audio Compression

The most popular attributes of audio signals is the existence of redundant (unnecessary) information place among the neighboring samples. Compression attempts to eliminate this redundancy and make the data de-correlated. To more particularly audio compression system consist of three essential modules. In the first module, a suitable transform coding is applied. Secondly, the produced transform coefficients are quantized to decrease the redundant information; the quantized data include errors but should be insignificant. at last, the quantized values are encoded using packed codes; this coding phase changes the format of quantized coefficients values using one of the fitting variable length coding technique [2].

Methods like DCT and DWT are used for natural data such images or audio signals. Reconstruction of the transformed signal by DCT can be done very efficiently; really this property of DCT is used for data compression. Likewise localization feature of wavelet along with time frequency resolution property makes Discrete Wavelet Transform (DWT) very suitable for speech/audio compression [7].

A. Discrete Cosine Transform

This transform had been invented by [Ahmed et al. 74]. Since that time it was studied commonly and extensively used in many applications. Currently, DCT is generally used transforms in video and image compression algorithms. Its popularity is due mostly to the fact that it performs a good data compaction; because it focus the information content in a relatively few transform coefficients [2]. DCT forming periodic, symmetric sequences from a finite length sequence in such a way that the original finite length sequence can be uniquely recovered. It consists basically of the real part of the DFT. This definition is realistic, since the Fourier series of a real and even function contains only the cosine terms. There are many ways to do this, so there are many definitions of the DCT. DCT-1 is used in signal compression applications in preference to the FFT

Manuscript received August 21, 2018; revised March 6, 2019.
doi:10.12720/jcm.14.4.301-306

because of a property energy compaction. The DCT-1 of a finite length sequence often has its coefficients more highly focus at low indices than the DFT does [8].

The DCT-1 is defined by the transform pair [3]:

$$c(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi(2x+1)}{2N}\right) \quad (1)$$

And for u = 0, 1, 2… N−1. Similarly, (2) represents the inverse transformation (IDCT).

$$f(x) = \sum_{x=0}^{N-1} \alpha(u) c(u) \cos\left(\frac{\pi(2x+1)}{2N}\right) \quad (2)$$

Likely for x= 0, 1, 2 …N− 1. In equations (1) and (2) α(u) is defined as:

$$\alpha(\mu) = \sqrt{1/N}\ for\ u = 0$$

and

$$\alpha(\mu) = \sqrt{2/N}\ for\ u \neq 0$$

B. Discrete Wavelet Transform (DWT)

A discrete wavelet transform can be define as a "small wave" that has its energy centered in time, and it supply a means for the analysis of transient, non-stationary or time varying phenomenon. It has oscillating wave like property. [7] The DWT is an execution of the wavelet transform by a discrete set of the wavelet scales and translations obeying a number of defined rules. In other words, the signal will be decomposed by this transform into mutually orthogonal set of wavelets, which is the major difference from the continuous wavelet transform, or its implementation for the discrete time series sometimes named Discrete-time continuous wavelet transform (DT-CWT). [6]

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \varphi\left(\frac{t-b}{a}\right) \quad (3)$$

where *"a"* is the scaling factor and *"b"* is the shifting factor. [7]

C. Quantization

Quantization is basically the process of decreasing the number of bits required to store coefficient values by decreasing its precision (e.g., rounding from float type to integer). The aim of quantization is to decrease most of the less significant high frequency coefficients to zero. [4].

D. Entropy Encoding

Entropy encoding is used to additional compresses the quantized values losslessly to provide enhanced overall compression. Diverse encoding methods can be used (e.g. Run Length, Huffman, Arithmetic, Shift Coding, and LZW). Statistical based encoding ways are used to eliminate data that are frequently occurring. Some encoding methods can, also, decrease the number of coefficients by eliminating the redundant data. In our proposed system the run length encoding way is applied initial to prune the long runs of quantized coefficients, then an enhanced progressive shift coding method is used to first of all to prune the existing second order statistical redundancy and, finally, to encode the created coefficients individually using variable encoding that depend on shift-key mechanism [2].

If the probability of occurrence of the element si is p(si), then it is most advantageous to represent this element - log2p(si) bits. If during coding it is possible to ensure that the length of all elements will be reduced to log2p(si) bits, then the length of the entire coding sequence will be minimal for all possible coding methods. Moreover, if the probability distribution of all elements F={p(si)} is constant, and the probabilities of the elements are mutually independent, then the average length of the codes can be calculated as

$$H = -\sum_i p(s_i) \cdot \log_2 p(s_i).$$

This value is called the entropy of the probability distribution F, or the entropy of the source at a given point in time.

However, usually the probability of the appearance of an element cannot be independent; on the contrary, it depends on some factors. In this case, for each newly encoded element si, the probability distribution F takes some value Fk, that is, for each element F = Fk and H = Hk.

In other words, we can say that the source is in the state k, which corresponds to a certain set of probabilities pk (si) for all elements si.

Therefore, given this amendment, we can express the average length of codes as

$$H = -\sum_k P_k \cdot H_k = -\sum_{k,i} P_k \cdot p_k(s_i) \log_2 p_k(s_i),$$

where Pk is the probability of finding the source in the state k.

So, at this stage, we know that compression is based on the replacement of frequently encountered elements with short codes, and vice versa, and also know how to determine the average length of codes. But what is code, coding, and how does it occur?

E. Huffman Algorithm

The Huffman algorithm uses the frequency of the appearance of identical bytes in the input data block, and assigns to frequently occurring blocks of a string of bits shorter and vice versa. This code is the minimum redundant code. Consider the case when, regardless of the input stream, the alphabet of the output stream consists of only 2 characters - zero and one.

First of all, when coding with a Huffman algorithm, we need to construct a scheme ∑. This is done as follows:

All letters of the input alphabet are ordered in decreasing order of probability. All words from the output stream alphabet (that is, what we will encode) are initially considered empty (recall that the output stream alphabet consists only of {0,1} characters).

The two characters aj-1 and aj of the input stream, which have the smallest probabilities of occurrence, are combined into one "pseudo-symbol" with probability p equal to the sum of the probabilities of the characters included in it. Then we append 0 to the beginning of the word Bj-1, and 1 to the beginning of the word Bj, which will subsequently be the codes of the characters aj-1 and aj, respectively.

We delete these characters from the alphabet of the original message, but we add a formed pseudo-character to this alphabet (naturally, it should be inserted into the alphabet at the right place, taking into account its probability).

Steps 2 and 3 are repeated until only 1 pseudo-character is left in the alphabet, containing all the original symbols of the alphabet. Moreover, since at each step and for each character, the corresponding word Bi is changed (by adding one or zero), after this procedure is completed, each initial symbol of the alphabet ai will correspond to a certain code Bi.

Suppose we have an alphabet consisting of only four characters - {a1, a2, a3, a4}. Suppose also that the probabilities of the appearance of these symbols are equal respectively to p1=0.5; p2=0.24; p3=0.15; p4=0.11 (the sum of all probabilities is obviously equal to one).

So, we will construct the scheme for the given alphabet.

We combine the two characters with the smallest probabilities (0.11 and 0.15) into the pseudo-character p '.

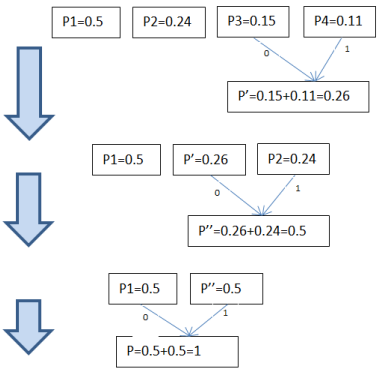
Remove the combined characters, and insert the resulting pseudo-character into the alphabet.

We combine the two characters with the lowest probability (0.24 and 0.26) into the pseudo-character p ".

Remove the combined characters, and insert the resulting pseudo-character into the alphabet.

Finally, combine the remaining two characters, and get the top of the tree.

If you make an illustration of this process, you get something like the following:



As you can see, with each union we assign codes 0 and 1 to the characters to be joined.

That way, when a tree is built, we can easily get the code for each character. In our case, the codes will look like this:

a1 = 0
a2 = 11
a3 = 100
a4 = 101

Since none of these codes is a prefix of any other (that is, we have received the notorious prefix set), we can uniquely identify each code in the output stream.

So, we have achieved that the most frequent symbol is encoded by the shortest code, and vice versa.

If we assume that initially one byte was used to store each character, then we can calculate how much we managed to reduce the data.

Suppose we had a line of 1000 characters at the entrance, in which the character a1 was encountered 500 times, a2 - 240, a3 - 150, and a4 - 110 times.

Initially, this line occupied 8000 bits. After coding, we get a string length of $\sum p_i l_i$ = 500 * 1 + 240 * 2 + 150 * 3 + 110 * 3 = 1760 bits. So, we managed to compress the data 4.54 times, spending an average of 1.76 bits on encoding each character of the stream.

Let me remind you that according to Shannon, the average length of the codes is. Substituting our probabilities into this equation, we obtain an average code length of 1.75496602732291, which is very, very close to the result we obtained.

However, it should be borne in mind that in addition to the data itself, we need to store the coding table, which will slightly increase the total size of the encoded data. It is obvious that in different cases different variations of the algorithm can be used - for example, sometimes it is more efficient to use a predetermined probability table, and sometimes it is necessary to compile it dynamically by traversing compressible data.

III. Related Work

Sumit Kumar Singh, et al., "**Discrete Wavelet Transform: A Technique for Speech Compression & Decompression**" authors used wavelet analysis to speech compression. A basis or mother wavelet is initially selected for the compression. The signal is then decomposed to a set of scaled and translated versions of the basis wavelet. The resulting wavelet coefficients that are unimportant or close to zero are truncated performing signal compression [9].

Rafeeq Mohammad and M. Vijaya Kumar, "**Audio Compression using Multiple Transformation Techniques**" They produce a comparative study of audio compression applying multiple transformation techniques. Audio compression with diverse transform techniques like Wavelet Transform, Discrete Cosine Transform, Wavelet Packet Transform (W.P.T) & Cosine Packet Transform is analyzed and compression ratio for each of

the transformation techniques is gained. Mean Compression ratio is computed for all of the techniques and compared. Performance measures like normalized root mean square error (NRMSE), signal to noise ratio (SNR), retained signal energy (RSE) are also computed and compared for each transform technique. Transform based compressed signals are encoded with encoding techniques like Mu-Law Encoding and Run-length Encoding (R.L.E) to decrease the redundancies [3].

Zainab T. Drweesh and Loay E.George, "**Audio Compression Based on Discrete Cosine Transform, Run Length and High Order Shift Encoding**" authors introduce an effective and low complexity coding scheme depend on discrete cosine transform (DCT). The proposed system composed of audio normalization, followed by DCT transform, scalar quantization, enhanced run length encoding and a new high order shift coding. To decrease the effect of quantization noise, which is notable at the low energetic audio segments, a post processing filtering stage is proposed as the last stage of decoding process [2].

Jithin James and Vinod J Thomas, "**A Comparative Study of Speech Compression using Different Transform Techniques**" This paper introduce a transform based methodology for compression of the speech signal. Where, diverse transforms such as DWT, DCT and FFT are exploited. A comparative study of performance of diverse transforms is made in terms of NRMSE, PSNR, SNR and compression factor (CF) [8].

Zainab T. Drweesh and Loay E.George, "**Audio Compression Using Biorthogonal Wavelet, Modified Run Length, High Shift Encoding**" The authors of research are design and implement a low complexity and efficient audio coding system depend on Biorthogonal tab 9/7 wavelet filter. The developed system composed of the audio normalization, followed by wavelet (Tap 9/7), progressive hierarchal quantization, modified run length encoding, and lastly high order shift coding to make the final bit stream. To decrease the effect of quantization noise, which is distinguished at the low energetic parts of the audio signal, a post processing filtering stage is inserted as final stage of the decoding processes [4].

Jithin James and Vinod J Thomas, "**Audio Compression Using DCT and DWT Techniques**" In this methodology, diverse transforms such as Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT) are exploited. A comparative study of performance of diverse transforms is made in terms of Peak signal-to-noise ratio (PSNR) and Signal-to noise ratio (SNR). The mean compression ratio is also computed for all the methods and compared [7].

IV. Methodology

At first the audio is in spatial domain which is hard for audio processing and compression, and need to be transformed into frequency domain in which a large amount of the audio information resides. For this reason, the Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) methods were used in the audio compression system [2]-[4], [7]-[9]. Transform techniques do not compress the signal, they provide information concerning the signal and using a variety of encoding schemes, compression of signal is done. Compression is performed by neglecting small magnitude coefficients as unimportant data and thus discarding them [8].

The system of audio compression consists of two units: first is the Encoding unit and second one is the Decoding unit. Each unit is carried out by using number of stages as in Fig. 1.

In first stage, the decomposition step, input speech/audio signal is decomposed into diverse resolution or frequency bands by using transform function (technique) like Discrete cosine transform, cosine packet transforms, discrete wavelet transform and wavelet packet transform [3].

After decomposition, compression involve quantization step which is used to reduce the information found in the transform coefficients in such a way that the process brings perceptually no error. There are two kinds of quantization are available: Non-Uniform and Uniform quantization [8]. In the papers above, uniform transform coding is used.

Encoding method is used to eliminate data that are repetitively happening. In encoding we can also decrease the number of coefficients by eliminating the redundant data. This helps in decreasing the bandwidth of the signal hence compression can be achieved [7].

The decoding unit consists of the opposite operations to those applied in the encoding process; also these operations are applied in reverse order [2].

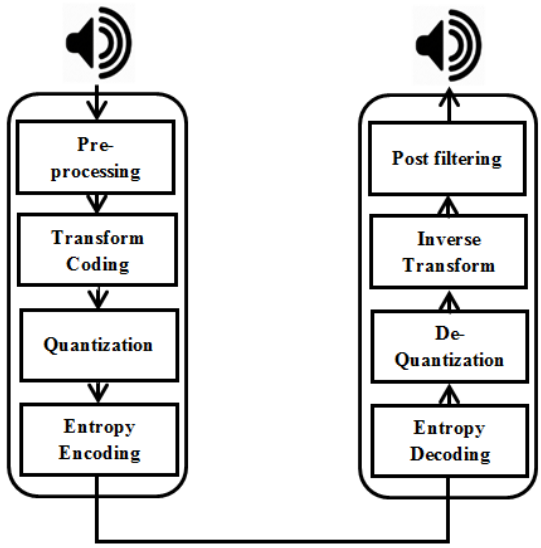


Fig. 1. Block diagram of speech/audio compression system

V. Performance Measures

For the audio compression method, depend on transform techniques, the performance are calculated in

terms of NMRSE, SNR, RSE, PSNR and Compression ratio (Factor) [3].

A. Compression Factor (CF)

Compression factor is also called as compression power used to quantify the reduction in data-representation size created by a data compression algorithm. It is the ratio of the original signal to the compressed signal [8].

$$CF = \frac{Original\ Signal\ (Audio) Length}{Compressed\ Audio\ Length} \tag{4}$$

B. Normalized Root Mean Square Error

$$NRMSE = \sqrt{\frac{\sum_n (x(n) - x'(n))^2}{\sum_n (x(n) - \mu(n))^2}} \tag{5}$$

$x(n)$ is the audio signal, $x'(n)$ is compressed audio signal or reconstructed. Generally RMSE represents the standard deviation of the differences between observed values and predicted values [3].

C. Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10 \log_{10} \frac{NX^2}{\|x - x'\|} \tag{6}$$

where N is the length of reconstructed signal, X is the maximum absolute square value of the speech signal x and $\|x\text{-}x'\|$ is the energy difference between the reconstructed and original signal [8].

VI. Theoretical Analysis

In [9] analysis of the compression process was performed by comparing the compressed-decompressed signal against the original. This was performed to compute the effect of the select of mother wavelet on the compression of speech. The outcomes however demonstrate that regardless of bases wavelet used the compression ratio is relatively close to one another.

In [3] the comparative study for audio compression using the D.W.T, D.C.T, W.P.T, C.P.T transform techniques have been carried out. And from the consequences Wavelet packet transform gives better compression ratio compared with the remaining transforms. Its gives enhanced compression ratio of about 27.8593 compared with the other three transforms. Mean SNR value is minimum for DCT 29.2830 and comparatively higher mean SNR value 43.4037 for CPT.

In [2] the performance of system is tested using diverse audio test samples; the test samples have diverse size and diverse in audio signal characteristics. The compression performance is evaluated using peak signal to noise (PSNR) ratio and compression ratio (CR). The test outcomes indicated that the compression performance of the system is hopeful. The compression ratio is greater than before with the increase of block size. Also the post processing stage enhanced the fidelity level of reconstructed audio signal.

In [8] the discrete wavelet transform executes very well in the processing and analysis of non-stationary speech signals. The main advantage of wavelet over other techniques is that the compression factor is not constant and it can be diverse while most other techniques have constant compression factors. Discrete wavelet transform safely improves the reconstruction of the compressed speech signal and also yields higher compression factor as compared to DCT and FFT. It is also observed that diverse wavelets have different effects on the speech signal and also global threshold yields best results than the level reliant threshold method.

Much work must be done to get better the wavelet compression. More particularly, the scheme could improve by (i) finding the more optimal mother wavelet and (ii) setting the truncation value which assure good compression factor and satisfactory signal quality. These schemes can play useful role in speech signal compression with reduced bitrates and excellent quality.

In [4] the performance effectiveness of the recommended audio encoding methods has been weighted using peak signal to noise (PSNR) ratio and compression ratio (CR). The attained consequences indicated that compression performance of the system is hopeful; it achieved enhanced results than the DCT based. The compression ratio is better with the increase of number of passes. Also the post processing stage enhanced the subjective quality of the reconstructed audio signal. Also, it improved the fidelity level of reconstructed audio signal when PSNR is fewer than 38 Db.

Finally, [7] experimental outcomes show that in common there is enhanced in compression factor and signal to noise ratio with DWT based technique. It is also observed that Specific wavelets have differed effects on the speech signal being represented.

VII. Conclusion

After doing audio compression by diverse transform coding techniques, we have found that the compression algorithm, such lossy and lossless and their coding techniques are best performing in their own fields. The idea of increasing storage capacity and reducing noise, bandwidth is achieved by these techniques. We conclude that the compression techniques like wavelet depend on quality of audio and computational complexity. In future, diverse transform coding techniques can be combined to improve the performance of the compression ratio and PSNR for the audio file.

References

[1] K. S. Solanki and N. Senani, “A survey on compression of an image using wavelet transform,” *International*

Journal of Computer Science and Information Technologies, vol. 6, no. 4, pp. 3859-3860, 2015.

[2] Z. T. Drweesh and L. E.George, "Audio compression based on discrete cosine transform, run length and high order shift encoding," *International Journal of Engineering and Innovative Technology*, vol. 4, no. 1, pp. 45-51, 2014.

[3] R. Mohammad and M. V. Kumar, "Audio compression using multiple transformation techniques," *International Journal of Computer Applications*, vol. 86, no. 13, pp. 9-14, 2014.

[4] Z. T. Drweesh and L. E. George, "Audio compression using biorthogonal wavelet, modified run length, high shift encoding," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 8, pp. 63-73, 2014.

[5] S. Bousselmi, N. Aloui, and A. Cherif, "DSP real-time implementation of an audio compression algorithm by using the fast hartley transform," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 472-477, 2017.

[6] K. A. Ramya and M. Pushpa, "A survey on lossless and lossy data compression methods," *International Journal of Computer Science and Engineering Communications*, vol. 4, no. 1, pp. 1277-1280, 2016.

[7] J. James and V. J. Thomas, "Audio compression using DCT and DWT techniques," *Journal of Information Engineering and Applications*, vol. 4, no. 4, pp. 119-124, 2014.

[8] J. James and V. J. Thomas, "A comparative study of speech compression using different transform techniques," *International Journal of Computer Applications*, vol. 97, no. 2, pp. 16-20, 2014.

[9] S. K. Singh, S. J. Khan, and M. K. Singh, "Discrete wavelet transform: A technique for speech compression & decompression," *International Journal of Innovations in Engineering and Technology*, Special Issue – ICAECE, pp. 99-105, 2013.

Dr. Razi J. Al-Azawi was born in Baghdad, Iraq, in 1971. Teaching at the University of Technology, Also work as Visitor Lecture at Informatics Institute for Postgraduate Studies, UITC, Baghdad, He got the scientific degree of assistant professor in 2009. He supervised numerous theses to students in undergraduate and postgraduate and He has a lot of papers at International Journals have impact factor He received the B.Sc. Degree in Laser and Optoelectronics Engineering from University of Technology, MSc degree in Modeling and Computer Simulation from University of Technology, in 1999 and the Ph.D. degree in Informatic's, in 2014. His research interests are Image processing, Mathematical Modeling, Optimization Theory, Information Theory, Modeling and Simulation, Web Security, Web design language, Finite Element, Artificial intelligence.

Zainab Talib Al-Ars was born in Baghdad, Iraq, in 1984. She received the B.S. degree from the University of Baghdad, College of Science, in 2006 and the M.Sc. degree from the same college in 2014, both in computer science. She is currently pursuing the Ph.D. degree with the Iraqi Commission for Computers and Informatics. Her research interests include Artificial Intelligence, Agent.