AUDIO COMPRESSION USING DCT

ECE4007 – Information Theory and Coding

Review-3

SLOT: **B1+TB1**

Team Members:

P.VAISHNAVI 17BEC0066
A.ABID ALI 17BEC0513

Submitted to:

Prof. K.S.Preetha

Title of the project:

AUDIO COMPRESSION USING DCT

Abstract:

As technology is growing, there is an increasing need for data storage and transmission. Compression makes the storage and transmission of data more effective. There are many forms of audio compression techniques that offer a range of complexity encoder and decoder, compressed audio quality, and varying amounts of data compression such as DCT, MPEG, DWT etc. This project aims to implement an audio signal compression algorithm using the Discrete Cosine Transform (DCT) technique. Implemented algorithm performance is assessed on the basis of compression ratio, Signal to Noise Ratio (SNR), and Root Mean Square Error (MSE). For further, efficient storage and transmission we also implemented Huffman Encoding.

Introduction:

DISCRETE COSINE TRANSFORM:

DCT is the real part of the Fourier transform. A discrete cosine transform (DCT) describes a finite sequence of data points in terms of the sum of oscillating cosine functions at different frequencies.

$$f(u) = \frac{\frac{\sqrt{2}}{n} C(u) \sum_{x=0}^{n-1} I(x) \cos(2x+1)\, \mu\pi}{2n}$$

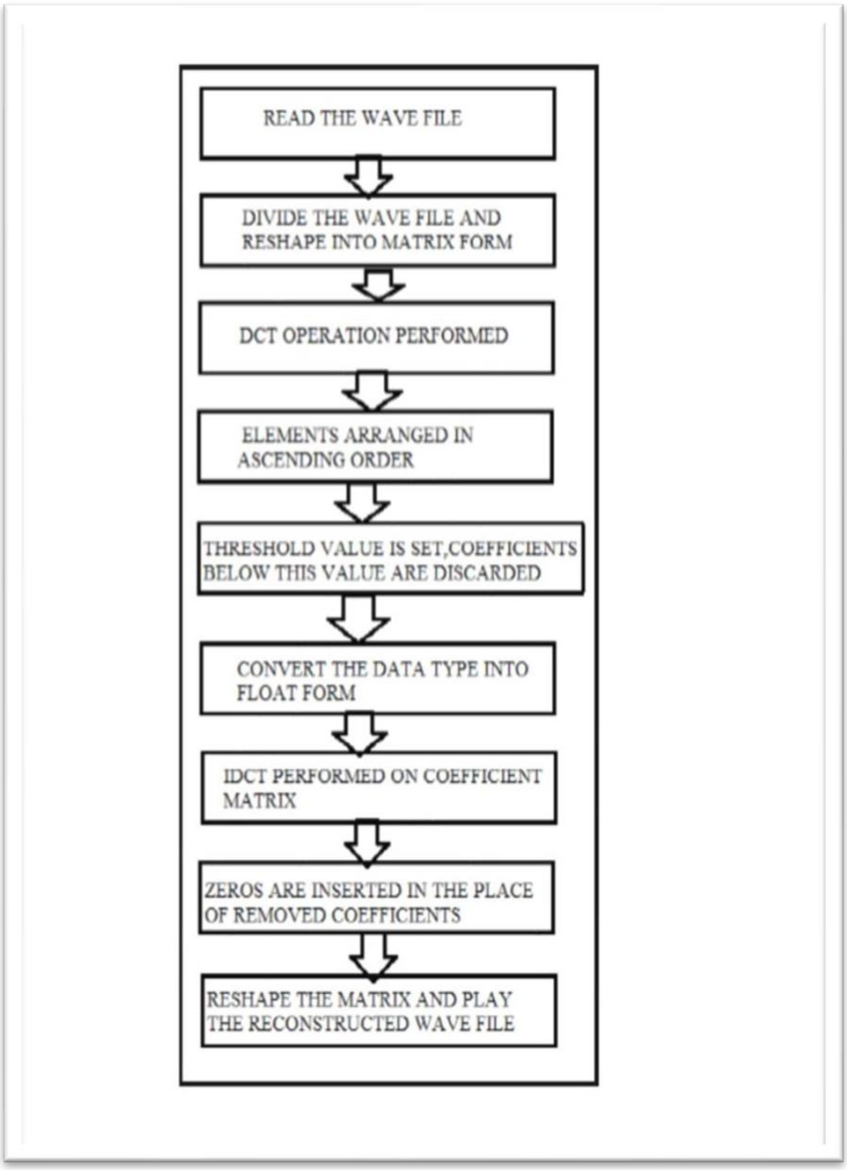
Thanks to the high correlation in neighboring coefficients, discrete Cosine Transform can be used for audio compression. We can reconstruct a series from very few DCT coefficients, quite accurately. This DCT property assists in successful data reduction. Its basic procedure is to take input audio data and convert it from one type of representation to another, while in our case the signal is an audio sample block. In this transformation, we transform set of spiral domain points to their equivalents in the frequency domain.

DCT retains those coefficients which are just enough to represent the signal at a particular desired quality. The DCT, as it has a strong energy compaction property in typical applications, appears to concentrate most of the signal information in a few low-frequency DCT components. It identifies pieces of information that can be effectively thrown away without seriously reducing the audio's quality.

Tools to be used:

MATLAB software

Block Diagram:



Methodology:

ALGORITHM:

DCT:

1) select the audio and find the actual signal size.
2) Find amplitude and frequency
3) Set a particular sampling frequency
4) In the procedure, we will decompose the signal into DCT basis vectors. There will be as many terms in the decomposition as there are samples in the signal. The expansion coefficients will be assembled in a vector X which will actually measure how much energy is stored in each of the components. Sort the coefficients from largest to smallest.
5) Then we will determine the number of DCT coefficients that comprises 99.9% of the energy in the signal.
6) we will set coefficients to zero that contain the remaining 0.1% of the energy.
7) 7.At last, using inverse discrete cosine transformation (IDCT) we can reconstruct the signal from the compressed representation of signal
8) 8.Finally we will, analyze the compressed signal through Compression factor, the quality of the signal by parameters such as Signal to Noise Ratio (PSNR), and Root Mean Square Error (MSE)

Huffman coding:

1) follow above same instruction up to 6^{th} step.
2) Then we quantize DCT coefficients with 256 levels between their Max and Min Value.
3) Then we will find the probability distribution of these quantized coefficients. 4.With the help of probability distribution we generate “dictionary” for Huffman coding.
4) Then we will perform Huffman encoding, which has much compressed for storage and transmission.
5) Then we will perform Huffman encoding. finally apply IDCT to get our file

EXPLANATION:

1) First, we will select the file to extract to our MATLAB simulator
2) Then we sample our audio file using read function at a specific frequency
3) After that we will take the DCT of the generated samples.
4) Then DCT coefficients are rearranged in descending order.
5) We then run a loop and iterate until the sorted vector's magnitude is equal to the original vector's magnitude of 99 percent. The vector magnitude gives us the energy that is in the signal.

6) Then we will apply Inverse cosine transform to reconstruct the signal.
7) Finally, by using the audio write function, these samples are again sampled back to the same sampling frequency thus creating an output compressed audio.
8) For Huffman coding, we need to quantize the coefficient, then we have found probability distribution, then dictionary, then encoding-decoding-IDCT to get compressed file.

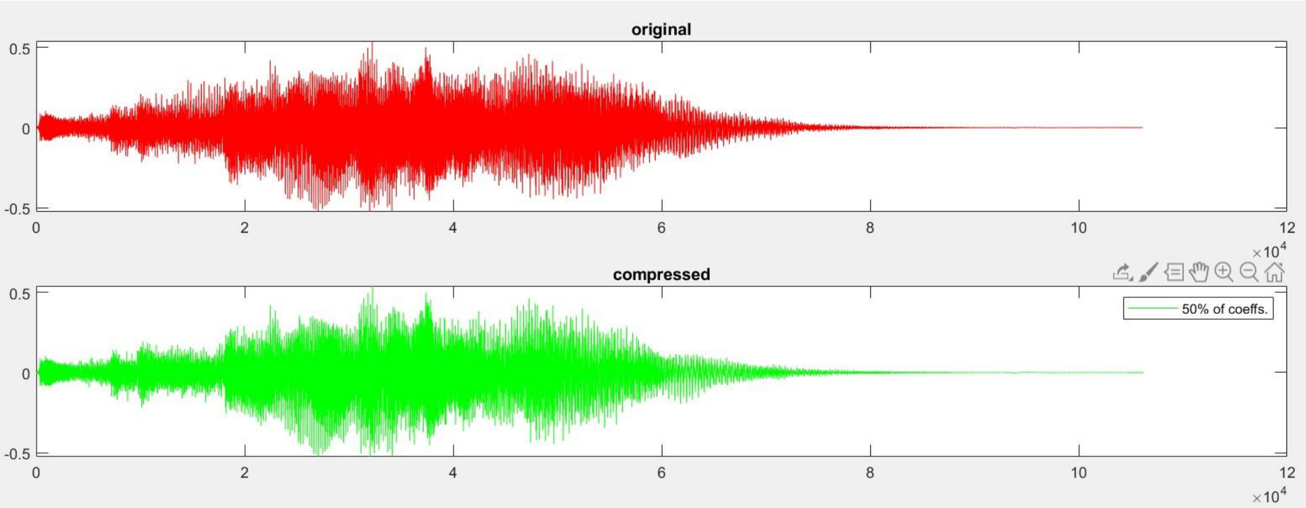
Work distribution among the batch members:

Member 1(17BEC066): Collection of various formats of Audio, performing DCT decomposing the signal into DCT basis vectors. finding the number of DCT coefficients that represent the signal are just enough to maintain quality of the audio. reconstruct the signal from the compressed representation using inverse discrete cosine transform (IDCT).

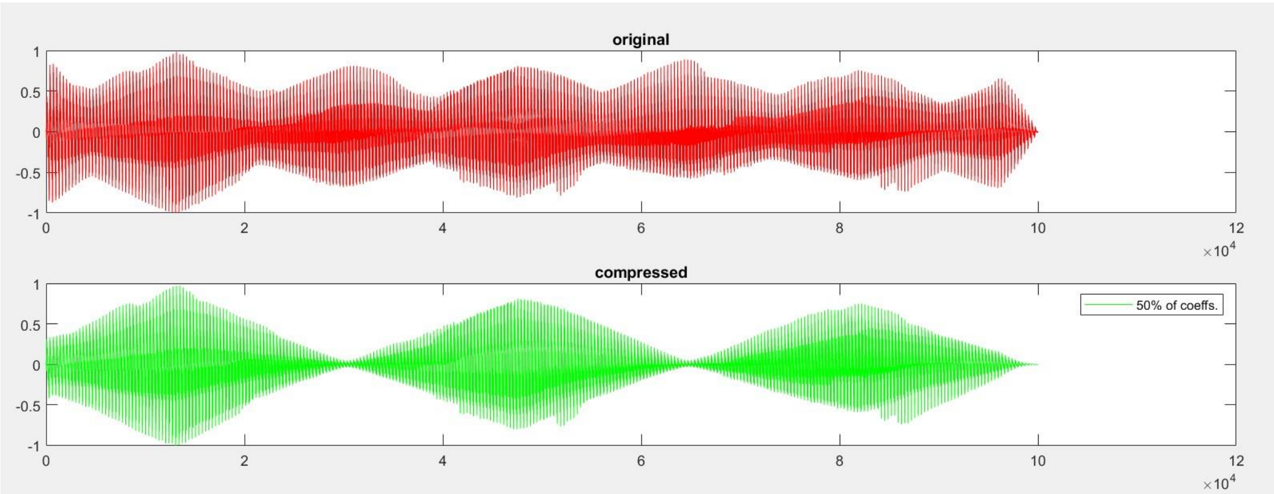
MEMBER 2(17BEC0513):
Finding the size of compressed signal and also analyze the quality of the signal by parameters such as Signal to Noise Ratio (SNR), and Root Mean Square Error (MSE). Performing Huffman encoding which involves quantization of coefficients, finding probability distribution, dictionary generation for Huffman coding, Encoding, Decoding, applying IDCT to get compressed file

Results and Discussions:

Audiosample_1:



Audiosample_2:



Sample	PSNR	MSE	Compression ratio
Audiofile_1	40.9280	2.292	1.9477
Audiofile_2	54.8459	0.46188	2.0011

Thus, we have successfully reconstructed with compression ratio 1.9477, 2.0011 for Audiofile_1 and Audiofile_2 respectively. We obtained PSN, MSE as 40.9280, 2.292 For Audiofile_1; 54.8459, 0.46188 for Audofile_2 thus providing us a compressed audio file of good quality without much deviation from the original audio signal.
We have also Implemented Huffman encoding even though there would be some quantization error, the quality of signal is still reasonable.

References:

1) Sankalp Shukla, Maniram Ahirwar, Ritu Gupta, Sarthak Jain, Dheeraj Singh Rajput. Audio Compression Algorithm using Discrete Cosine Transform (DCT) and Lempel-Ziv-Welch(LZW) Encoding Method, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th -16th Feb 2019

2) M. V. Patil, Apoorva Gupta, Ankita Varma, Shikhar Salil.Audio and Speech Compression Using DCTand DWT Techniques, International Journal of Innovative Research in Science, Engineering and Technology *Vol. 2, Issue 5, May 2013.*

3) Sudha Rawat, Ajeet Kumar Verma Survey paper on image compression techniques, ,2017

4) Mr. R. R. Karhe Ms. P. B. Shinde Ms. J. N. Fasale Audio Compressionusing DCT & CS, , 2015

5) 5.A.M.Raid, W.M.Khedr , M. A. El-dosuky and Wesam Ahmed .Jpeg Image Compression Using Discrete Cosin Transform, , 2014