Implementation of Audio Compression using Wavelet

Hauwa T. Abdulkarim
Department of Electrical/Electronic Technology, College of Education,
Minna, Nigeria
talatuabdulk@gmail.com

Tijjani S. Abdulrahman
Department of Electrical/Electronic Technology, College of Education,
Minna, Nigeria
teejays1569@gmail.com

Abubakar S. Mohammed
Dept. of Electrical/Electronic Engineering, Federal University of Technology, Minna, Nigeria
abussadiq@yahoo.com

ABSTRACT
The need to transmit audio signal has increased tremendously over the past decade. In view of this, audio compression is a sure technology of the multimedia age which facilitates ease of transmission. The change in the telecommunication infrastructure, in recent years, from circuit switched to packet switched systems has also reflected on the way that speech and audio signals are carried in present systems. In many applications, such as the design of multimedia workstations and high quality audio transmission and storage, the goal of audio compression is to encode audio data to take up less storage space and less bandwidth for ease of transmission. This paper presents the implementation of audio compression using wavelet. The implementation procedure, the Matlab code and the results obtained are duly presented and discussed. The final results indicate that a good reconstruction was performed and the performance of the wavelet was excellent with the performance variables all in the region well above 60%.

CCS Concepts
• Hardware → Communication hardware, interfaces and storage → Signal processing systems → Digital Signal Processing

Keywords
Audio; Wavelet; Compression; Transmission; MATLAB code; Performance

1. INTRODUCTION
Audio compression- a popular 21st century technique enables the substantial data rates associated with uncompressed digital audio signal to be efficiently stored and transmitted [Bowman et al., 1993]. In this modern day, sounds of telephone, television, radios etc. undergo some form of compression or the other to improve the quality of sound and ultimately reduce storage space and bandwidth.

The advancement in radio communication has geared up the development of wireless multimedia sensor networks (WMSNs) which can process multimedia data such as video and audio streams, still images collected from the application area[Ding and Marchionini 1997; Fröhlich and Plate 2000; Tavel 2007]. Energy is one of the scarcest resources [Sannella 1994; Forman 2003] in such networks and data compression is one of the implementing techniques to save energy in these networks [Tavel 2007].

CoRI'16, Sept 7–9, 2016, Ibadan, Nigeria.

The increase in data transfer has led to the need to develop appropriate signal processing techniques to handle audio and video compression [Brown et al. 2003]. Many types of digital data can be compressed in a way that reduces the size occupied on a computer memory or the bandwidth needed to stream it with no of the full information in the original signal. Audio compression can be achieved by either lossless compression (in which all the information from the original signal is recoverable) or by lossy data compression (in which the original signal is permanently changed by removing redundant information [Yu 2006]. Although, lossless compression would keep all the information of the original signal unaltered, it has the limitation of compression ratio of about 3:1 while with lossy compression algorithms, the compression ratio can be as high as 12:1 or higher [Spector 1989].

Audio compression is very much employed in this computer age where information can be sent over the internet and other ways[Zhao and Shen 2010]. Obviously the presence or absence of some details in a sound signal makes no difference to the user and removing the details during compression is of advantage to storage and bandwidth required and consequently maximizing the compression efficiency.

2. METHODOLOGY
2.1 Implementation
The implementation of the audio compression experiment was done using Matlab. An audio file 'short_beethoven.wav' and 'plot_time_scale.m' were both downloaded into the Matlab directory. The audio signal was loaded using a Matlab command 'wavread'. This original signal was plotted in order to be able to differentiate it with the compressed signal. Figure 1 shows the original signal.

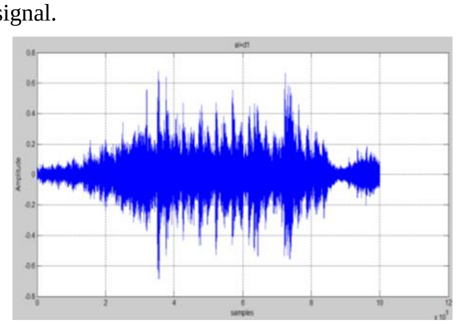

Figure 1: Plot of Original Signal

Discrete Wave Transform (DWT) analysis was then performed using the command *[ca1,cd1]=dwt(s,'db3')* which gives a one-level step decomposition sequentially. The three level decomposition for both the approximate and detail coefficient obtained are presented in The Matlab command 'soundsc' was used to listen to the decomposed signal and the effect of decomposition was observed.

After decomposition was complete the next was reconstruction of all the details and approximations values from their coefficients and levels of decomposition were done and the signal was checked for errors to be sure a perfect reconstruction was done before compression. Invert directly decomposition of the original signal was then done and this was followed by reconstruction of the original signal. The signal was compressed after inverse discrete wave transform (IDWT).. Error (k) was determined between the compressed and the original signal. The error in this case was a value of

$$error, k = (s - sd)$$
$$= 7.666 \times 10^{-6}$$

The error, k is a value which defines the deviation of the denoised signal from the original. This value is small enough to assume the deviation is negligible and this therefore implies that a near perfect reconstruction was made.

3. RESULTS AND DISCUSSION

Figure 2 shows the plot of the original signal and the approximation coefficient for three decomposition levels. 'db3' was used for the 3-level decomposition, this is shown in Figure 3.

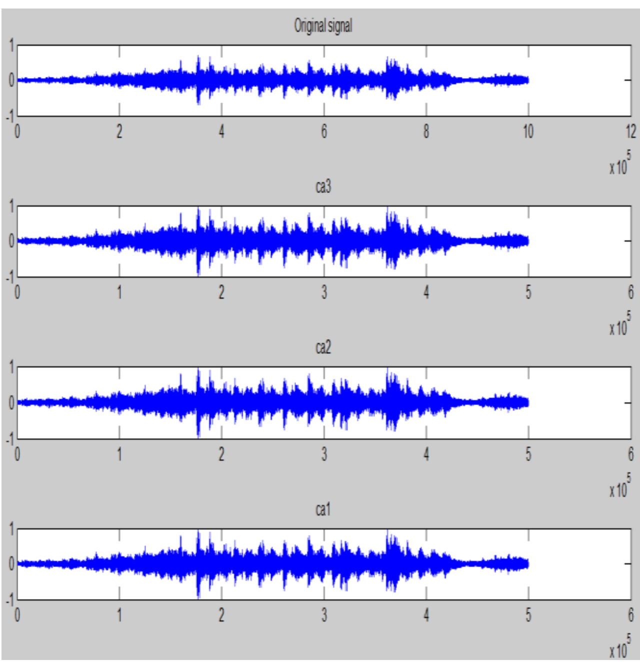

Figure 2: Approximation Coefficient

Figures 4 and 5 show the histogram of the original signal, Approximation and Detail values for varying levels of decomposition. Histogram is a very handy tool to present results of experiments, in this case it presents at a glance the detail and approximation values at different levels. This represents the energy and frequencies stored for decomposition.

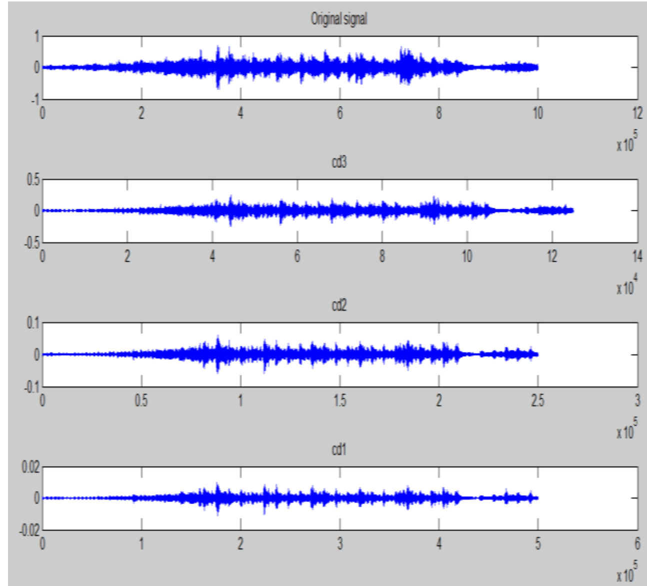

Figure 3: Detail Coefficient for 3 Levels

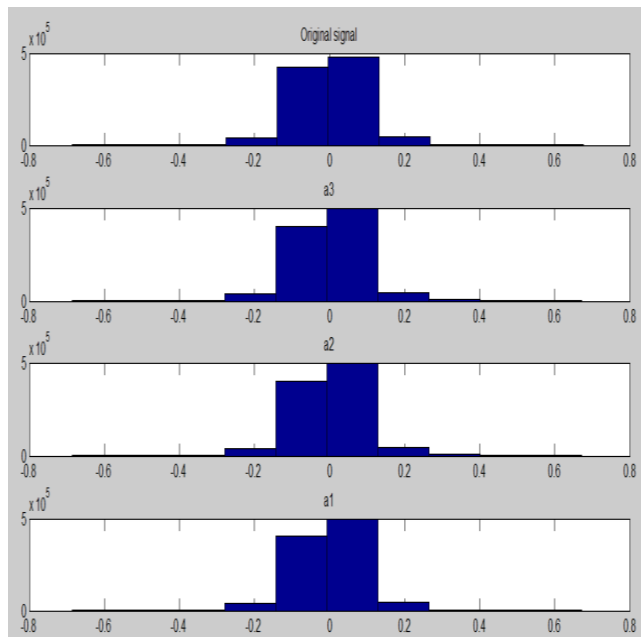

Figure 4: Plot of Histogram of original signal and Approximation values

3.1 Compression and De-noising

'ddencmp' Matlab command was used to automatically generate the thresholding needed. The function also denoises and compresses. Figure 6 presents the denoised and the original signals for assessment and comparison. To the eye, the two signals seem very identical although there are differences that may not be detected with human eye. The signal was denoised using global thresholding option applying the Matlab command 'wdencmp'.

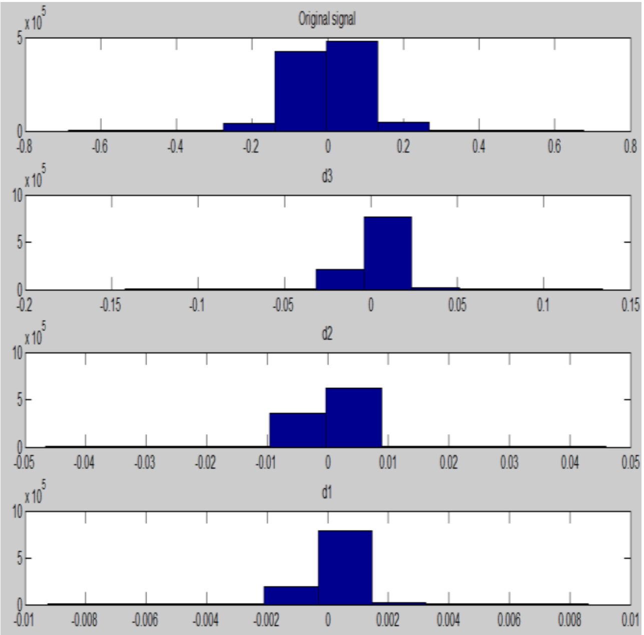

Figure 5:Plot of Histogram of original signal and Detail values

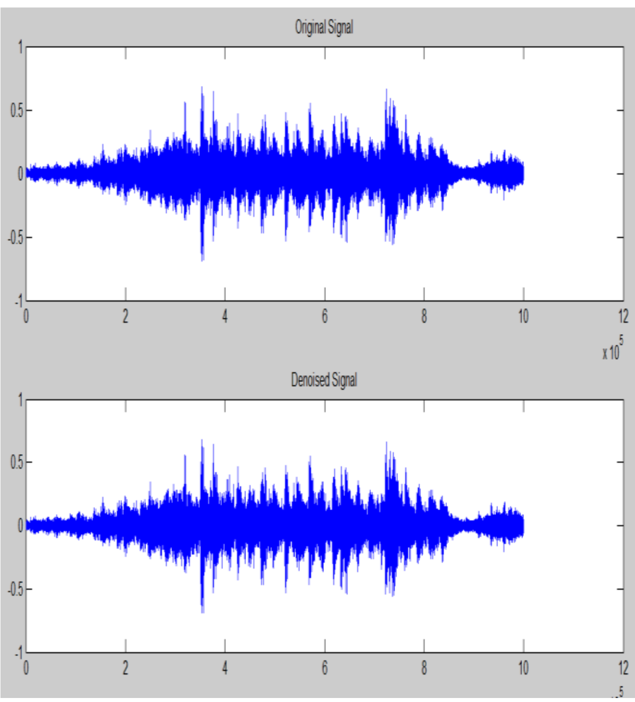

Figure 6: Plot of original and denoised signals

The Matlab code used for compressing the signal is
[thr,sorh,keepapp]=ddencmp('cmp','wv',s);
[sd,csd,lsd,perfo,perfl]=wdencmp('gbl',s,'db3',3,thr,sorh,keepapp);

'Perfo' and 'perfl' are the variables which defines the performance of the wavelet used for compression.'perfo' indicates the number of zeroed coefficients. For the present experiment a 68.0609% was obtained. This indicates that a good compression can be achieved at least beyond 60%. 99.9915% was obtained for 'perfl' which indicates almost equal energy in the compressed signal and the original signal. This implies that no data was loss as a result of the compression.

Plot_time_scale.m was used to plot the discrete transform in colour.

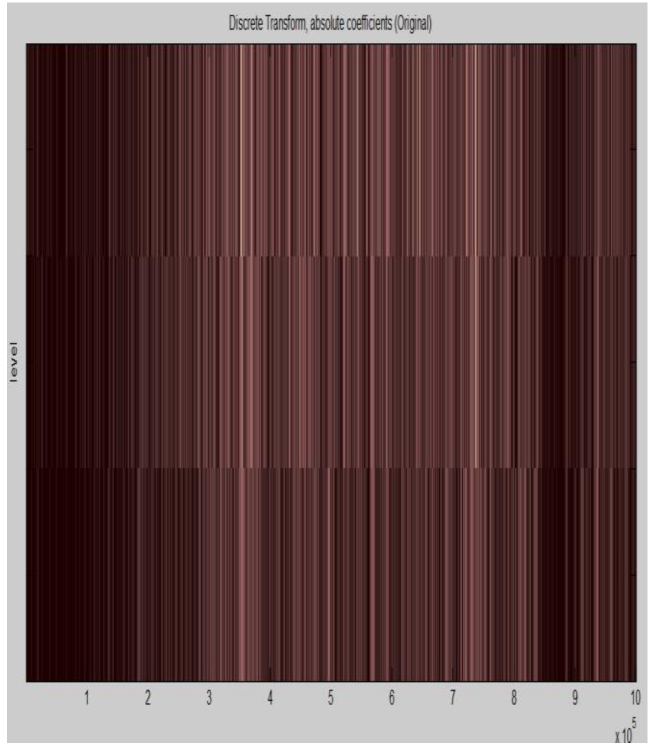

Figure 7: Image time-scale diagram representation of signal detail decomposition value levels

.

4. CONCLUSION

Audio compression was implemented using wavelet. The performance of the wavelet was excellent with 'perfo'=68.06%, 'perfl2'=99.9929%, and 'perfl'=99.9915%. This shows The reconstruction was good as well since the error is negligible. Audio compression is used for transmission and storage. The compression is achieved by representing each sample of digitized data by lesser number of bits and making it occupy lesser space and consequently easy to transmit or store.

5. ACKNOWLEDGMENTS

The authors wish to thank Tertiary Education Trust Fund (TETFund), Abuja, Nigeria and College of Education, Minna, Nigeria for the sponsorship.

6. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[2] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/964696.964697

[3] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[4] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305

[5] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= http://doi.acm.org/10.1145/332040.332491.

[6] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[7] Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= http://doi.acm.org/10.1145/90417.90738

[8] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.

[9] Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw*. 79, 5 (May. 2006), 577-590. DOI= http://dx.doi.org/10.1016/j.jss.2005.05.030.

[10] Zhao O. D. and Sheng-qian, M. A. 2010 “Speech Compression with Best Wavelet Packet Transform and SPIHT Algorithm” Second International Conference on Computer Modeling and Simulation in 2010