Discrete Wavelet Transform: A Technique for Speech Compression & Decompression

Sumit Kumar Singh
M.Tech Scholar, Deptt. of Computer Science & Engineering
Al-Falah School of Engineering & Technology, Faridabad, Haryana, India
er.sumit.software@gmail.com

Sher Jung Khan
Assistant Professor, Deptt. of Computer Science & Engineering
Al-Falah School of Engineering & Technology, Faridabad, Haryana, India
er.sumit.software@gmail.com

Mohit Kumar Singh
Deptt. of Electronics & Communication Engineering, R. D. Engineering College, Ghaziabad, UP, India
er.mksingh@gmail.com

Abstract: This paper applies wavelet analysis to speech compression. A mother or basis wavelet is first chosen for the compression. The signal is then decomposed to a set of scaled and translated versions of the mother wavelet. The resulting wavelet coefficients that are insignificant or close to zero are truncated achieving signal compression. Analysis of the compression process was performed by comparing the compressed-decompressed signal against the original. This was conducted to determine the effect of the choice of mother wavelet on the speech compression. The results however showed that regardless of bases wavelet used the compression ratio is relatively close to one another.

Keywords: Compression, Lossless & Lossy Compression, Speech Compression, Wavelet Transform, 1-D DWT.

I. INTRODUCTION

The primary purpose of early computers is computing. With the advancements in technology, computers were developed to provide a wide variety of applications. One of these is the representation and storage of non-numeric information into binary code. Such applications require longer data representations for quality. These data requires larger storage space and longer access time. Another set back would be in terms of data transmission. Transmission time would be longer since the data that would be sent is larger. With these disadvantages comes the need for the compression of data. Data compression refers to the reduction of the number of bits needed to represent the data used for storage and transmission. The ideal goal of data compression is to contain the original information in as few bits as possible. Data compression can be classified as either lossy or lossless.

II. LOSSLESS VS LOSSY COMPRESSION

Lossless compression refers to the technique that assures an exact representation of data after going through the compression-decompression process. This type of compression is well appropriate for applications where the loss of even minimal information has a tremendous negative effect. Such effects can be seen in storage of text files in databases where minimal loss would have tremendous consequences. On the other hand, Lossy compression represents the compression of data with a certain degree of loss in information content. This technique is more suited for applications where overall information content is not dramatically diminished with minimal loss in information. Examples of such applications are in image and speech signals.

III. ALGORITHM

Given a signal s of length N, the DWT consists of log2N stages at most. Starting from s, the first step produces two sets of coefficients: approximation coefficients cA_1, and detail coefficients cD_1. These vectors are obtained by convolving s with the low-pass filter Lo_D for approximation, and with the high-pass filter Hi_D for detail, followed by dyadic decimation. The first step is

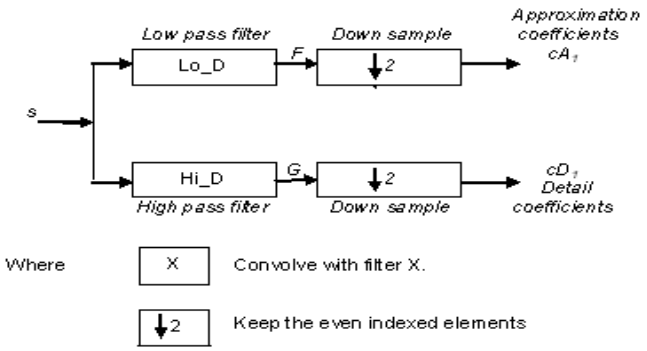

Fig. 1.

The length of each filter is equal to 2N. If n = length (s), the signals F and G are of length n + 2N - 1, and then the coefficients cA1 and cD1 are of length

$$floor\left(\frac{n-1}{2}\right)+N$$

The next step splits the approximation coefficients cA1 in two parts using the same scheme, replacing s by cA1 and producing cA2 and cD2, and so on.

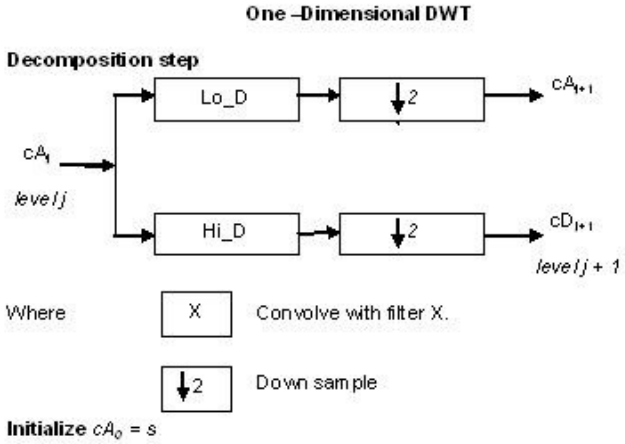

Fig. 2.

So the wavelet decomposition of the signal s analyzed at level j has the following structure: [cAj, cDj, ..., cD1].

Conversely, starting from cAj and cDj, the IDWT reconstructs cAj-1, inverting the decomposition step by inserting zeros and convolving the results with the reconstruction filters.

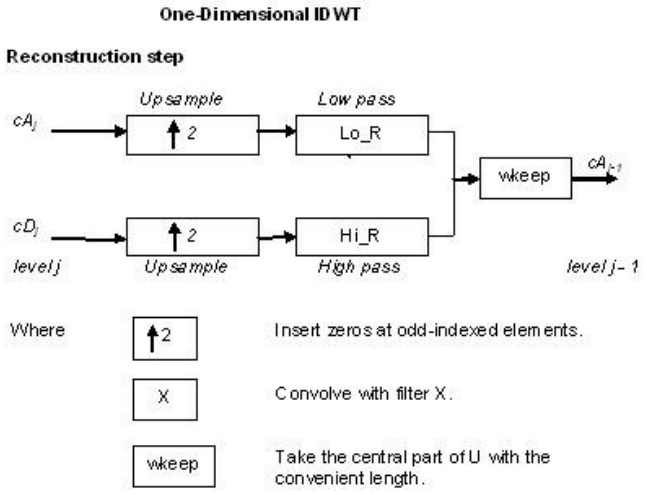

Fig. 3.

IV. FUNCTION USED

A. *dwt*

1) Syntax: [cA, cD] = dwt(X, 'wname')

2) Description: The dwt command performs a single level one-dimensional wavelet decomposition with respect to either a particular wavelet or particular wavelet decomposition filters (Lo_D and Hi_D) that you specify.

[cA, cD] = dwt(X, 'wname') computes the approximation coefficients vector cA and detail coefficients vector cD, obtained by a wavelet decomposition of the vector X. The string 'wname' contains the wavelet name.

B. *idwt*

1) Syntax: X = idwt(cA, cD, 'wname')

2) Description: The idwt command performs a single-level one-dimensional wavelet reconstruction with respect to either a particular wavelet or particular wavelet reconstruction filters (Lo_R and Hi_R) that you specify.

X = idwt(cA, cD, 'wname') returns the single-level reconstructed approximation coefficients vector X based on approximation and detail coefficients vectors cA and cD, and using the wavelet 'wname'.

C. *wavedec*

1) Syntax: [C, L] = wavedec(X, N, 'wname')

2) Description: wavedec performs a multilevel one-dimensional wavelet analysis using either a specific wavelet ('wname') or specific wavelet decomposition filters.

[C, L] = wavedec(X, N, 'wname') returns the wavelet decomposition of the signal X at level N, using 'wname'. N must be a strictly positive integer. The output decomposition structure contains the wavelet decomposition vector C and the bookkeeping vector L.

D. *waverec*

1) Syntax: X = waverec(C, L, 'wname')

2) Description: waverec performs a multilevel one-dimensional wavelet reconstruction using either a specific wavelet or specific reconstruction filters (Lo_R and Hi_R). waverec is the inverse function of wavedec. X = waverec(C, L, 'wname') reconstructs the signal X based on the multilevel wavelet decomposition structure [C,L] and wavelet 'wname'.

E. wdencmp

1) Syntax: [XC, CXC, LXC, PERF0, PERFL2] = wdencmp('gbl',X,'wname',N,THR,SORH,KEEPAPP)

2) Description: wdencmp is a one- or two-dimensional de-noising and compression-oriented function. Wdencmp performs a de-noising or compression process of a signal or an image, using wavelets.

[XC, CXC, LXC, PERF0, PERFL2]= wdencmp('gbl', X, 'wname', N, THR, SORH, KEEPAPP) returns a de-noised or compressed version XC of input signal X (one- or two-dimensional) obtained by wavelet coefficients thresholding using global positive threshold THR.

F. wavread

1) Syntax: y = wavread('filename')

[y,Fs,bits] = wavread('filename')

2) Description: wavread supports multichannel data, with up to 32 bits per sample, and supports reading 24- and 32-bit .wav files.

y = wavread('filename') loads a WAVE file specified by the string filename, returning the sampled data in y. The .wav extension is appended if no extension is given. Amplitude values are in the range [-1, +1].

G. wavwrite

1) Syntax: wavwrite(y, 'filename')

wavwrite(y,Fs,'filename')

2) Description: wavwrite writes data to 8, 16, 24, and 32-bit .wav files. wavwrite(y, 'filename') writes the data stored in the variable y to a WAVE file called filename. The data has a sample rate of 8000 Hz and is assumed to be 16-bit. Each column of the data represents a separate channel. Therefore, stereo data should be specified as a matrix with two columns. Amplitude values outside the range [-1, +1] are clipped prior to writing.

V. FLOWCHARTS

A. Sound compression

This algorithm (Fig. 4) on the basis of the level of decomposition and percentage compression compresses the sound waveform and displays the compressed sound waveform. Compressed sound waveform is saved in the MATLAB file “compressedsound.mat”.

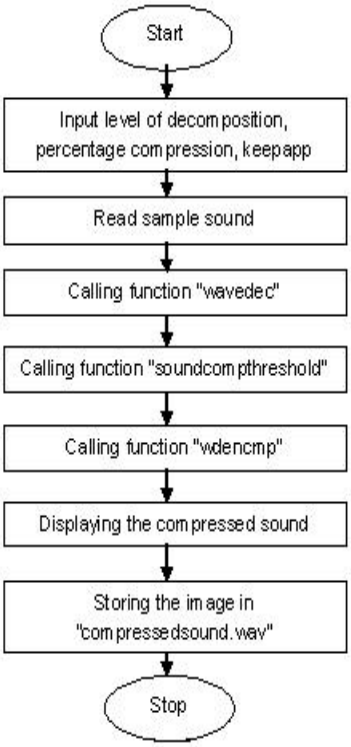

Fig. 4.

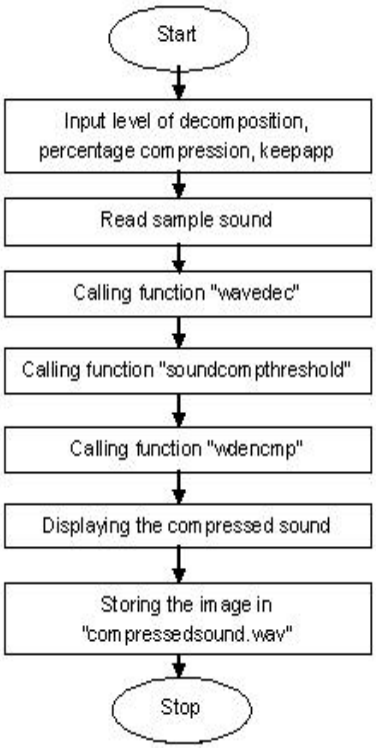

Fig. 5.

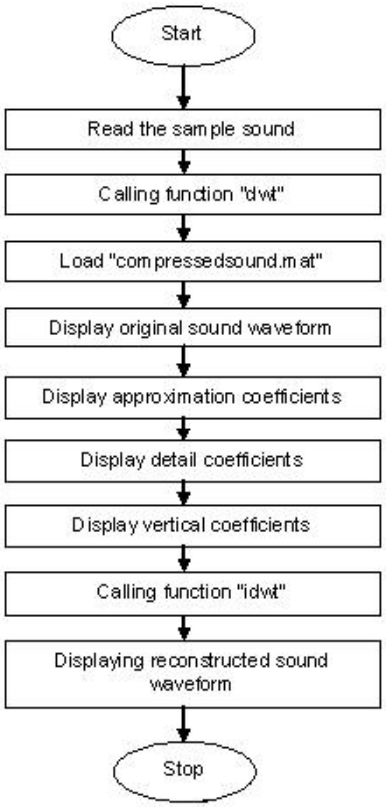

Fig. 6.

B. Sound Compress

This algorithm (Fig. 5) compresses the coefficients of the sound and stores them in the Matlab file named as "compressedsound.mat"

C. One-dimensional DWT

This algorithm (Fig. 6) performs the one-dimensional discrete wavelet transform on sound waveform using bi-orthogonal wavelet “bior 3.7” and displays the approximation and detail coefficients of the sound waveform.

D. Sound uncompress

This algorithm (Fig. 7) reconstructs the original sound from the compressed coefficients and stores the reconstructed sound as “reconstructedsound.wav”.

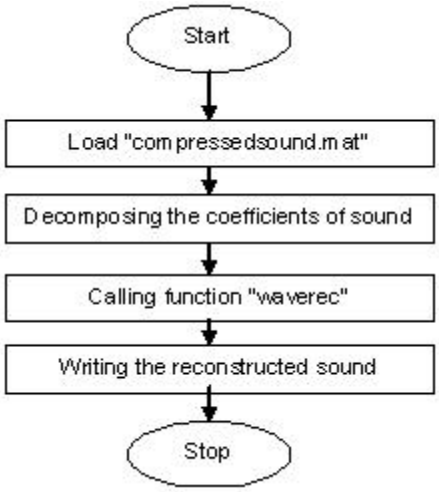

Fig. 7.

VI. RESULTS

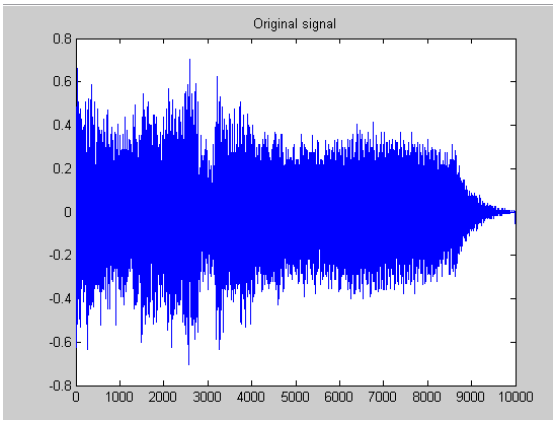

Fig. 8. Original Signal

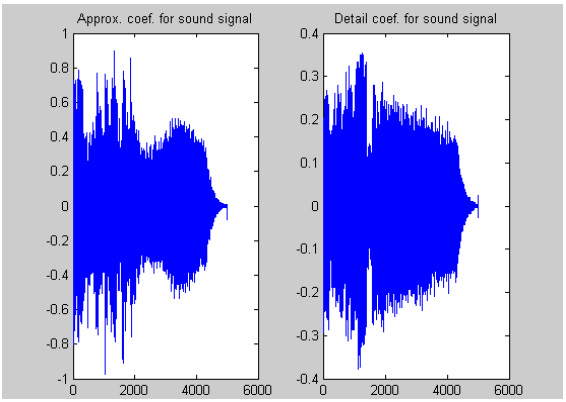

Fig. 9. Approximation & Detail Coefficients

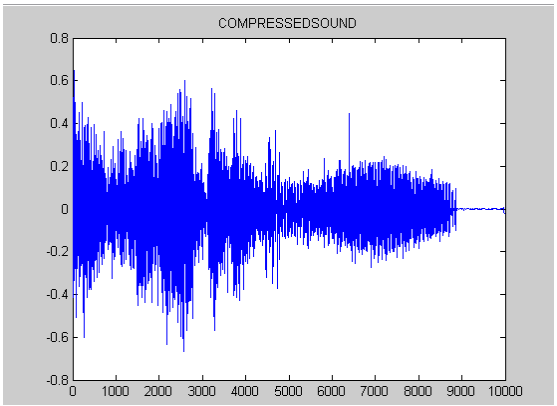

Fig. 10. Compressed Sound

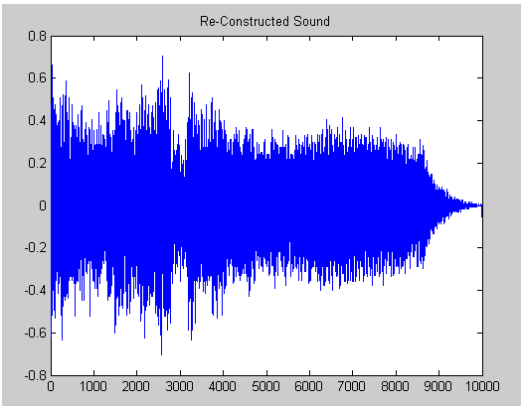

Fig. 11. Reconstructed Sound

VII. REFERENCES

[1]. Agbinya, J. I., "Discrete Wavelet Transform Techniques in Speech Processing", IEEE Tencon Digital Signal Processing Applications Proceedings, IEEE, New York, NY, 1996, pp 514-519.

[2]. Kinsner, W. and Langi, A. "Speech and Image Signal Compression with Wavelets", IEEE Wescanex Conference Proceedings, IEEE, New York, NY, 1993, pp. 368-375.

[3]. Graps, A., An Introduction To Wavelets, IEEE Computational Sciences and Engineering, Volume 2, Number 2, pp: 50-61, Summer 1995.

[4]. Mawla A., Najih M. A., Ramli A. R., Ibrahim A., and Syed A. R., "Comparing Speech Compression Using Wavelets With Other Speech Compression Schemes," IEEE Student Conference on Research and Development, Vol. 1, pp. 55-58, August 2003.

[5]. Amara Graps, "An Introduction to Wavelets",published by the IEEE computer society, vol. 2, no. 2, pp.50-61 Summer 1995.

[6]. R. Polikar, "The Wavelet Tutorial".

[7]. MATLAB7.0, www.mathworks.com.