Proc. of the 2nd International Conf. on Optics and Laser Applications
ICOLA'07, September 5-7, Yogyakarta, Indonesia

A Comparison of Discrete Cosine Transform and Discrete Wavelet Transform Techniques in Audio Compression

Endra

Department of Computer Engineering, University of Bina Nusantara
Jl K.H. Syahdan No.9 Kemanggisan / Palmerah, 11480, Jakarta
Tel. 021-53696930, fax. 021-5300244, email : endraoey@binus.ac.id

Abstract– **Transform coding is one of lossy audio compression methods which converts time domain sampled audio signal into a transform coefficients. In this paper, a comparison of *Discrete Cosine Transform* (*DCT*) and *Discrete Wavelet Transform* (*DWT*) coding compression performance has done. *WAV* audio format that contains uncompressed audio is used as audio signals that will be compressed. The average of coefficients absolute value is used as a threshold value.**

Experimental results show that *DCT* and *DWT* coding have almost the same compression performance. Audio signal that has many peaks and transience decreases the compression performance, both for *DCT* and *DWT* coding. *DCT* coding has a better compression performance than *DWT* coding for higher threshold value.

Keywords– Transform coding, DCT, DWT, WAV audio format

I. INTRODUCTION

Data compression or source coding is the process of encoding information using fewer bits than an unencoded representation would use through use of specific encoding schemes. Audio compression is a form of data compression designed to reduce the size of audio file. The methods of audio compression algorithms have been classified as either lossless or lossy. Lossless methods restore the compressed data to exactly the same form as the original, while lossy methods only generate an approximation [1]. Transform coding is one of lossy audio compression methods which converts time domain sampled audio signal into a transform coefficients. This transform is used to determine what information in an audio signal is perceptually irrelevant, because the information in transform domain is statistically concentrated in just a few transform coefficients. In this paper, I compare the compression performance of *DCT* and *DWT* that commonly used in transform coding algorithm.

WAV audio format that contains uncompressed audio in the *pulse-code modulation* (*PCM*) format is used as audio signals that will be compressed.

II. BASIC THEORY

A *DCT* is a Fourier-related transform similar to the *Discrete Fourier Transform* (*DFT*), but using only real numbers. The *DCT* is often used in lossy data compression, because it has a strong energy compaction property where most of the signal information tends to be concentrated in a few low-frequency components of the *DCT*. A *DCT*, implies an even extension of the original function, each symmetric extension gives rise to a different transform [2].

The *DCT*-II is the most commonly used form and is often simply referred to as "the *DCT*". It is defined by

$$y(k) = w(k)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{p(2n+1)(k)}{2N}\right),\ k = 0,1 \ldots N\text{-}1$$
$$w(k) = \begin{cases} \dfrac{1}{\sqrt{N}}, & k = 0 \\ \sqrt{\dfrac{2}{N}}, & 1 \le k \le N-1 \end{cases} \tag{1}$$

where *x(n)* is the *N*-vector describing the signal and *y(k)* is the *N*-vector describing the result of the transform and *N* is the length of signal. The inverse *DCT* is defined by

$$x(n) = \sum_{k=0}^{N-1} w(k)\,y(k)\cos\left(\frac{p(2n+1)(k)}{2N}\right),\ n = 0,1 \ldots N\text{-}1 \tag{2}$$

w(k) is defined by Eq (1).

DWT refers to wavelet transforms for which the wavelets are discretely sampled. The *DWT* coefficients of a signal is calculated by passing it through a series of analysis/decomposition low-pass filter and high-pass filters, the filter outputs then downsampled by 2. The outputs of *DWT* giving the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). The decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled [3].

In order to get the signal back (doing inverse *DWT*), detail coefficients and approximation coefficients upsampled by 2 and then passing through synthesis low-pass filter and high-pass filter [4]. Coefficients of analysis and synthesis filter are determined by the kinds of wavelet filter that are used.

III. EXPERIMENTAL RESULTS

These are the steps of compression algorithm that is used in this paper, first a *WAV* audio file is read and returning the sampled data to a vector *x*. Applied *DCT* or *DWT* to *x* produces as many coefficients *C* as there are samples in the audio signal. Then, every *C* smaller in magnitude than a threshold value is completely neglected, that is, replaced by zero produces C_t. The threshold value is calculated using the average of coefficients absolute value. After that, C_t is quantized into 16 bits using Eq (3) produces C_q.

$$C_q = fix\left(\frac{\left(2^{15}-1\right)C_t}{\max\left|C_t\right|}\right) \tag{3}$$

fix and *max* in Eq (3) mean round towards zero and maximum value. Then, S_q are run-length encoded, where each time a zero is encountered in the input data, *two* values are written to the output file. The first of these values is a zero, a flag to indicate that run-length compression is beginning. Finally, the run-length encoded coefficients are written to a binary file as a result of compression. The steps for decompression algorithm are the reverse of the steps of compression algorithm, those are : read binary file, run-length-decoding, dequantization, inverse *DCT* or inverse *DWT* and write to a WAV audio file to get reconstructed signal. *Energy Ratio (ER)*, *Compression Ratio (CR)* and *Power Signal to Noise Ratio (PSNR)* that defined by Eq (4), (5) and (6) are parameters that are used to measure compression performance.

$$ER = \left(\frac{\sum x_d^2}{\sum x^2}\right) x100\,\% \tag{4}$$

$$CR = \left(\frac{\text{Input file size - Compressed file size}}{\text{Input file size}}\right) x100\,\% \tag{5}$$

$$PSNR = 10\log\left(\frac{\sum x_d^2}{\sum e^2}\right) dB \tag{6}$$

x_d is a reconstructed/decompressed signal and *e* is error signal that is a difference between original signal and reconstructed signal.

Fig. 1. shows graph of uncompressed *WAV* audio signals (*A*, *B*, *C*, *D* and *E*). Table 1 shows the compression results of *DCT* coding.

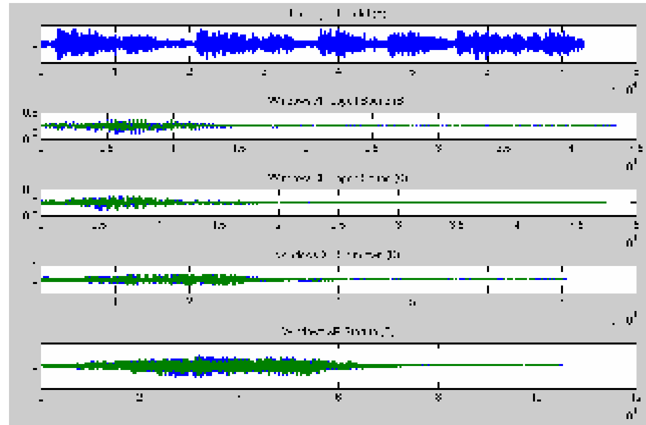

Fig. 1. Uncompressed WAV audio signals.

The used wavlet filters are *Daubechies* (*db20*), *Coiflets* (*coif5*) and discrete *Meyer* (*dmey*). Table 2, 3 and 4 show the compression result of *DWT* coding using maximum decomposition level for wavelet filters : *db20*, *coif5* and *dmey*.

Table 1. Compression results of *DCT* coding

Input File, Size (Kb)	Threshold value	*ER (%)*	*CR (%)*	*PSNR(dB)*
A, 143	0.0964	96.28	51.75	14.16
B, 176	0.0056	99.64	82.95	29.34
C, 186	0.0063	99.68	83.33	29.28
D, 276	0.0142	99.81	80.80	28.84
E, 415	0.0145	99.36	78.55	27.57

Table 2. Compression results of *DWT* coding, *db20*

Input File, Size (Kb)	Threshold value	*ER (%)*	*CR (%)*	*PSNR(dB)*
A, 143	0.1140	95.22	51.75	13.01
B, 176	0.0068	99.86	82.95	28.85
C, 186	0.0077	99.89	84.95	29.99
D, 276	0.0183	99.43	84.06	28.58
E, 415	0.0206	99.74	81.69	26.44

Table 3. Compression results of *DWT* coding, *coif5*

Input File, Size (Kb)	Threshold value	*ER (%)*	*CR (%)*	*PSNR(dB)*
A, 143	0.1145	95.23	51.05	13.01
B, 176	0.0070	98.53	82.95	26.13
C, 186	0.0080	99.84	84.95	29.66
D, 276	0.0186	99.85	84.06	28.90
E, 415	0.0208	99.75	81.69	26.28

Table 4. Compression results of *DWT* coding, *dmey*

Input File, Size (Kb)	Threshold value	*ER (%)*	*CR (%)*	*PSNR(dB)*
A, 143	0.1135	95.27	51.05	13.10
B, 176	0.0068	99.76	82.95	28.97
C, 186	0.0074	99.58	83.87	29.79
D, 276	0.0180	99.28	84.06	28.17
E, 415	0.0203	99.72	82.17	26.51

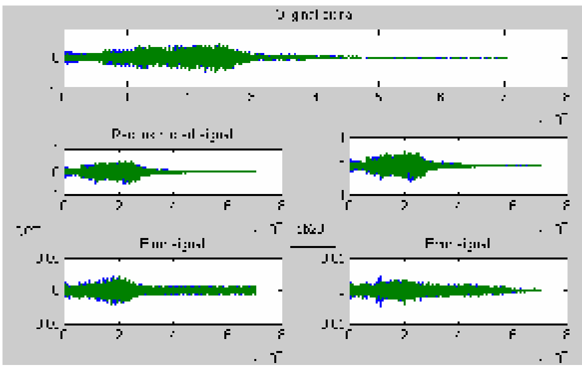

Fig. 2. Original, reconstructed and error signal of "Windows XP Shutdown" WAV audio file for *DCT* and *DWT* (*db20*).

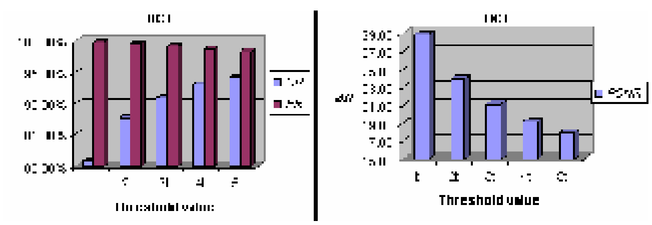

Fig. 3. The effect of threshold value on compression performance for *DCT* coding.

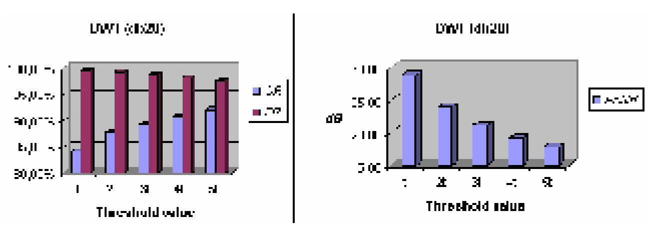

Fig. 4. The effect of threshold value on compression performance for *DWT* (*db20*) coding.

Table 1 until 4 show that *DCT* and *DWT* (*db20, coif5* and *dmy*) coding have almost the same compression performance. Both transform coding have *ER* greater than 95 %, *CR* greater than 78 % and *PSNR* greater than 26 *dB*, except for "Hallelujah Handel" *WAV* audio file that only give *CR* about 51 % and *PSNR* about 13 *dB* until 14 *dB*.

Fig.1 shows that "Hallelujah Handel" *WAV* audio signal has many peaks and transience that cause the compression performance are worse than other *WAV* audio file. Fig.2. shows the original, reconstructed and error signal of "Windows XP Shutdown" *WAV* audio file for *DCT* and *DWT* (*db20*) coding.

Fig.3 shows the effect of threshold value on compression performance for *DCT* coding ,where *t* is the average of coefficients absolute value. The increasing of threshold value from *t* to 5*t* makes *CR* increases 13.41 % but *ER* and *PSNR* decreases 1.49 % and 11 %.

Fig. 4 shows the effect of threshold value on compression performance for *DWT* (*db20*) coding. The increasing of threshold value from *t* to 5*t* makes *CR* increases 7.97 % but *ER* and *PSNR* decreases 1.94 % and 11.87 %.

IV. CONCLUSIONS

Experimental results show that *DCT* and *DWT* coding have almost the same compression performance. *WAV* audio signal that has many peaks and transience ("Hallelujah Handel" *WAV* audio file) decreases the compression performance about 27 % for ***compression ratio*** and about 12 ***dB*** for ***PSNR*** both for ***DCT*** and ***DWT*** coding .

DCT coding has a better *compression ratio* increment about 5 % than *DWT* coding (*db20*), but both has almost the same *energy ratio* and *PSNR* decrement for the increasing of threshold value from *t* to *5t*.

REFERENCES

[1] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, California (1999).
[2] Porat, Boaz, *A Course In Digital Signal Processing*, John Willey & Sons, Inc., New York (1997).
[3] Stéphane Mallat, *A Wavelet Tour of Signal Processing*, Academic Press (1999).
[4] N. F. Chamberlain, *Introduction to Wavelets v1.7*, South Dakota School of Mines and Technology (2002).