Signal Compression Using the Discrete Wavelet Transform and the Discrete Cosine Transform

Robert J. Barsanti, and Athanasios Athanason
Department of Electrical and Computer Engineering
The Citadel
Charleston, SC, 29407
e-mail: robert.barsanti@citadel.edu

Abstract—Signal compression using the wavelet transform and discrete cosine transform is investigated on short duration synthetic signals. The signal fidelity as measured by peak signal to noise ratio (PSNR) is compared for an assortment of wavelet basis functions. Plots of PSNR versus the signal compression are provided. The signal entropy is computed in each basis as an indicator of the performance of the compression process. Simulations demonstrating the effectiveness of each basis are presented and contrasted.

Key Words: Wavelet Analysis, Signal Compression, Discrete Cosine Transform

I. INTRODUCTION

The compression of time series data for the purpose of reducing storage and transmission speed is of interest to a wide variety of entertainment, business and scientific disciplines. As shown in Fig 1, most compression schemes use a linear transform to reduce redundancy and identify sparsity in the signal characteristics before quantizing and coding the resulting coefficients. Although a large number of compression algorithms can be found in the literature, this paper focuses on a comparison of the advantage gained through the use of one linear transform over another. The important issues of source encoding and decoding are not addressed in this work. Instead we choose to focus on the capabilities of the discrete wavelet transform (DWT), and the discrete cosine transform (DCT) to condense signal information. This paper will apply the wavelet-based compression techniques similar to that of [1] and [2] to synthetic signals, and extend these methods to the discrete frequency domain using the DCT. A basic premise explored in the paper is that the entropy [3] of the signal in the transform domain can be computed to confirm the compression performance of the decomposing basis.

Fig. 1. Block diagram of signal compression scheme

The remainder of this paper is divided into the following sections; Section II describes the DCT and DWT data transforms. Section III describes the basics of data compression and introduces signal entropy and information cost functions. Section IV explains the simulations and presents results, and section V contains a summary.

II. DATA TRANSFORMS

Transformations used for signal compression should possess two important properties. The first is that it should translate the signal energy into only a few transform coefficients. The second is that the resulting coefficients should be uncorrelated. The Karhunen-Loeve transform (KLT), which derives the basis vectors from the eigenvalue decomposition of the signal, provides the ideal decomposition for this purpose. The KLT is data depended and inefficient to compute [4]. The DCT performs closely to the KLT on many practical signals and is simpler to compute. Likewise, the DWT has been used in many data compression applications and is used to compare and contrast in this study.

A. The Discrete Cosine Transform

The discrete cosine transform (DCT) of a sequence $x(n)$ is given by [5],

$$
\begin{align*}
\text{DCT}(y(k)) &= \sum_{n=0}^{N-1} x(n) \cos \left( \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right) \\
&= \sum_{n=0}^{N-1} x(n) \cos \left( \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right)
\end{align*}
$$

for $k = 0, 1, \ldots, N-1$, where $N$ is the length of the signal and $y(k)$ is the DCT coefficient at frequency $k$. The DCT is a linear transform that maps a sequence of real numbers into another sequence of real numbers. It is widely used in image and audio compression because it can effectively represent the data in a compact form, with most of the energy concentrated in a few coefficients. The DCT has a number of properties that make it useful for compression, including orthogonality and energy compaction.
\[ C(k) = p(k) \sum_{n} x(n) \cos \left( \frac{\pi k (2n+1)}{2N} \right), \quad (1) \]

\[ p(k) = \frac{\sqrt{N}}{\sqrt{N}} \text{ for } k = 0 \]

\[ p(k) = \sqrt{1/2N} \text{ for } k = 1, 2, \ldots, N - 1, \]

where the sequence \([C(1), C(2), \ldots, C(N-1)]\) is the transform sequence. This set contains all information necessary to reconstruct the original signal by use of the inverse DCT (IDCT). The DCT is not merely the real part of the DFT. However, as shown in [6] the DCT and DFT are related and fast algorithms can be used to compute the DCT. In many cases the DCT will outperform the DFT in signal compression [7].

B. The Discrete Wavelet Transform

The DWT of a sequence \(x(n)\) is given by [8],

\[ W(J, b) = \frac{1}{n \sqrt{2^J}} \sum_{n} x(n) \Psi^{*} \left( \frac{n - b}{2^J} \right), \quad (2) \]

where \(\Psi\) represents wavelet function, which is dilated and contracted by the integer scale factor \(J\), and delayed in time by parameter \(b\). For an \(N\) point sequence the scale factor \(J\) assumes the values \(J = 0, 1, \ldots, \log_2(N)\), producing a decomposition of the input into octave bands. The delay values \(b\) are related to the scale by \(b = K \cdot 2^J\) for \(K\) an integer. Therefore, the DWT output is decimated by a factor of two at each successive octave \(J\). The DWT requires an input sequence length that is an even power of two, (i.e., \(N = 2^p\), with \(p = \text{integer}\)) and produces an equal number of wavelet coefficients. [8, 9, 10]

Eq. 2 produces a DWT output \(W\) which is set of \(N\) coefficients that represent the data in the wavelet domain. This set contains the information necessary to reconstruct the original signal from the corresponding wavelet function via the inverse wavelet transform (IDWT). The coefficients represent the correspondence between the input signal and the decomposing wavelet function at each particular delay \(b\), and scale \(J\). For simplicity and ease of display, the discrete wavelet coefficients can be represented as a vector \((W)\) by summing over the scales. [9, 10]

\[ W = [w_1, w_2, w_3, \ldots, w_N] . \quad (3) \]

This formulation allow for plotting of the DWT output as shown in the bottom plot of Fig. 2. A sample signal along with the DCT and DWT coefficients are displayed in Fig 2.

III. DATA COMPRESSION

A. Basics

The data compression procedure contains three steps as displayed in the block diagram of Fig 3. Step 1 transforms the signal \(x(n)\) into the transform domain via the DFT or DWT. Step 2 applies a threshold to the transform coefficients (to remove data), and (3) perform the inverse transform on modified coefficients to produce the compressed signal \(y(n)\). [5, 10]

The steps of quantization and coding (typically performed prior to storage or transmission) of the compressed signal were not performed in this study. Since these steps would only add complexity to the desired direct comparison of transform basis functions.

The step 2 compression threshold can be applied in a variety of ways. In this paper the coefficients were sorted by magnitude, and iteratively set to zero from smallest to largest to obtain a predetermined compression ratio or alternatively to achieve given level of fidelity.

\[ x(n) \rightarrow \text{DWT or DCT} \rightarrow \text{Threshold Coefficients} \rightarrow \text{IDWT or IDCT} \rightarrow y(n) \]

Fig. 3. Block diagram of compression scheme.

Any number of compression scores can be found in the literature. In this paper we will measure the signal compression as the ratio of original data length to transform data length after setting coefficients to zero.
Define signal compression percentage SC [11]

\[ SC = \left( \frac{\text{# of zeros in decomposition}}{\text{# original signal data points}} \right) \times 100\%. \quad (4) \]

The mean squared error (MSE) defined as

\[ MSE = \frac{1}{N} \sum_{n} (x(n) - y(n))^2. \quad (5) \]

The symbol \( x(n) \) represents the original signal and \( y(n) \) represents the compressed signal, as shown in Fig. 3.

The measure used to compare the relative performance of the transforms the peak signal to noise ratio (PSNR) is defined as [4]

\[ PSNR = 10 \cdot \log_{10} \left( \frac{\max(\sum x(n)^2)}{MSE} \right). \quad (6) \]

Fig. 4 shows a signal and its compressed version for comparison. The compressed signal was obtained using only 15% of the original signal coefficients, meaning 85% of the coefficients were set to zero, thus from Eq. 4 \( SC = 85\% \).

![Original Signal and Compressed Signal with 85% Zeros](image)

**B. The Signal Entropy and Information Cost**

The method of signal compression employed in this study depends upon the ability to represent the data with a small number of coefficients in the transforming basis. Therefore the ‘best basis’ is the one that most compactly represents the data by concentrating the signal information into the fewest significant coefficients. An information cost function \( Q \) can be used to quantify this idea. A popular cost function is the Shannon Entropy which is defined as [3]:

\[ Q(x) = \sum_{i} |x_i|^2 \log(1/|x_i|^2). \quad (7) \]

In signal processing the information gained from observing a single element \( x_i \) of a signal \( x = \{x_i\} \) can be found from the expression [3]:

\[ I(x_i) = \log(1/p_i), \quad I = 0 \quad \text{for} \quad p_i = 0, \quad (8) \]

where \( p_i = |x_i|^2 / ||x||^2 \) is the normalized energy of the \( i^{th} \) element of the signal. The entropy of the signal \( x \) is then defined as the expected value of \( I(x) \) over the length of the signal and is given by [3]:

\[ H(x) = E[I(x_i)] = \sum_{i} p_i I(x_i) = \sum_{i} p_i \log(1/p_i). \quad (9) \]

\( H(x) \) is the entropy of the signal, and is a measure of the average information content per symbol of the sequence \( x \). For example, if two signals contain equal energy but different entropy, the signal with the lower entropy has its energy concentrated in fewer elements [3]. It is this attribute of \( H(x) \) that makes it useful for comparing compression domains.

The entropy \( H(x) \) is bounded such that:

\[ 0 \leq H(x) \leq \log(N) \quad (10) \]

where \( N \) is number of elements of the sequence \( x \). Note that, \( H(x) = 0 \) only if the probability \( p_i = 1 \) for one \( i \), and all remaining probabilities are zero. Also the upper bound is achieved when the signal energy is evenly distributed among the coefficients [3]. Thus the best basis for compression is the decomposition with the smallest entropy.

Since \( H(x) \) is independent of rearrangement of the sequence elements, this justifies the simpler vector formulation of \( W \) given by Eq. 3.

**IV. SIMULATIONS AND RESULTS**

The data compression techniques discussed in section III were applied to a variety of signals using computer simulations and Matlab® software. An overview of the simulation is shown in Fig. 5. The DCT and the DWT were evaluated for their ability to maintain signal fidelity (measured by PSNR) as the number of transform coefficients set to zero was increased. A representative selection of the results obtained from the simulations is described in the following paragraphs.

Three simulated signal waveforms were generated using \( 2^{10} \) data points. For each signal the DCT and DWT coefficients were computed and stored. The DWT was applied using two different wavelet basis functions, the Haar 4, and the Symmlet 4 [5, 11]. The entropy of the signal coefficients in each of the basis’s were compute using Eq. 9 and are tabulated in table 1. The transform coefficients were sorted and then recursively set to zero (smallest to largest) to produce the compressed set of
signal coefficients. For each iteration, the next smallest coefficient was set to zero, the inverse transform was applied, and both the MSE and PSNR computed. The PSNR vs. the percentage of zeros are plotted in Fig 6, 7, and 8. Each figure also displays the original signal along with the “compressed” signal, which was constructed using the transform with the highest PSNR at the 90% signal compression.

![Block diagram of compression simulation](image)

**Fig. 5.** Block diagram of compression simulation.

Comparison of table 1 with Fig. 6, 7, and 8 reveals that in each case the domain with the lowest entropy also produced the best compression. Therefore, the signal entropy can be used to indicate which domain will perform the best.

The results for signal 1 (a pure sinusoid) are shown in Fig 6, the DCT performed best. This result was expected since the DCT basis efficiently concentrates the signal 1 energy in its domain. For signal 2, and 3 (see Fig. 7, and 8), the DWT provided better compression results than that of the DCT. In the case of signal 2, the DCT was shown to be a poor basis for this spikey signal. The DWT using the Symmlet 4 basis achieved the best PNSR. In the case of signal 3, the square wave basis of the DWT using the Haar 4 wavelet provided the best compression performance.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Signal 1</th>
<th>Signal 2</th>
<th>Signal 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT</td>
<td>0.002</td>
<td>4.570</td>
<td>3.154</td>
</tr>
<tr>
<td>Haar</td>
<td>2.877</td>
<td>4.024</td>
<td><strong>3.029</strong></td>
</tr>
<tr>
<td>Symmlet</td>
<td>2.139</td>
<td><strong>3.743</strong></td>
<td>3.087</td>
</tr>
</tbody>
</table>

Table 1. Computed values of the Information cost $H(x)$ for signals 1, 2, and 3 in each domain using Eq.7

![Fig. 6. Plot of 1) PSNR vs. threshold for signal 1, and 2) comparison of the original signal and signal with 90% of the transform coefficients set to zero.](image)

![Fig. 7. Plot of 1) PSNR vs. threshold for signal 2, and 2) Comparison of the original signal and signal with 90% of the transform coefficients set to zero.](image)
V. SUMMARY

This paper compared the performance of the DCT, and DWT threshold compression on select variety of simulated signals. The details of method were developed, and simulations of the performance of the proposed technique were presented. The paper also discussed signal entropy in the cosine and wavelet domain and related it to the algorithm performance. Future work will include the performance on more realist signals e.g., audio, or communications signals, and study of wavelet compression on still digital images.

VI. ACKNOWLEDGEMENT

The authors would like to acknowledge The Citadel Foundation for the grant that supported this research.

VII. REFERENCES