A COMPARISON OF THE DISCRETE HERMITE TRANSFORM AND
WAVELETS FOR IMAGE COMPRESSION

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Christopher John Bellis

May, 2012
The Discrete Hermite Transform is a relatively recent technique for signal and image processing. Early examples of applications of the Discrete Hermite Transform range from artifact removal from ballistocardiograms to a more pure signal processing application such as two-dimensional correlation. As such, exploring the ability to use this technique for the compression of medical images could provide alternatives to current methods. Better or more simple methods will hopefully allow for the expansion of the application to image compression hence reducing computer resources and increasing information sharing.

After applying wavelets and the Discrete Hermite Transform to a set of images, it was found that there is a significant correlation among several of the test cases and through observation of images, that the discrete Hermite Transform compared favorably.
ACKNOWLEDGEMENTS

For many large endeavors, a person cannot rely solely on oneself to be successful. My graduate studies and thesis are no exception and I would like to take this opportunity to thanks those that have helped.

First and foremost I would like to thank my Advisor, Dr. Dale Mugler. He is not only the person who introduced me to both of the mathematical techniques used here, but was his patient guidance and understanding through my unusual schedule that made completing my degree possible.

I would also like to thank my Thesis committee members Dr. Sheffer and Dr. Reedy. Their confidence and pragmatic approach to investigating this topic provided a reassuring assessment that it is worthwhile.

For the tremendous support and constant curiosity concerning my studies I would like to thank my family and friends. Their understanding during this project lessened the sting of missed activities. Without the support of my family, I would have never started such a ambitions project and it was my family that kept me on track.

Lastly, I would like to thank my wife. I cannot express my thanks for your understanding of how important completing my degree was to me, the sacrifice of not
spending time together and the unconditional support in the last few semesters. I will forever be in your debt for lending me your ear while I verbally worked through the many ideas in my head while putting them to paper.
DEDICATION

In Memory of

Erin Michelle Dillon
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
</tbody>
</table>

## CHAPTER

I. INTRODUCTION ......................................................... 1
   1.1 Purpose of Study ............................................. 1
   1.2 Significance of Study ....................................... 2
   1.3 Specific Aims .................................................. 2
   1.4 Statement Of Hypothesis ..................................... 2

II. LITERATURE REVIEW .................................................. 4
   2.1 Hermite Transform Description ............................. 4
   2.2 Measures of Image Fidelity ................................ 7

III. MATERIALS AND METHODS ........................................... 15
   3.1 Images Used ................................................... 15
   3.2 Wavelet Processing .......................................... 16
   3.3 Hermite Processing ........................................... 16
   3.4 Statistical Analysis ......................................... 17

IV. RESULTS ............................................................. 19
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Performance measures of Wavelet Compression</td>
<td>22</td>
</tr>
<tr>
<td>4.2 Performance measures of Wavelet Compression (Continued)</td>
<td>22</td>
</tr>
<tr>
<td>4.3 Performance measures of Wavelet Compression (Continued)</td>
<td>23</td>
</tr>
<tr>
<td>4.4 Performance measures of Wavelet Compression (Continued)</td>
<td>23</td>
</tr>
<tr>
<td>4.5 Performance measures of Wavelet Compression - Calculated</td>
<td>24</td>
</tr>
<tr>
<td>4.6 Performance measures of Wavelet Compression - Calculated (Continued)</td>
<td>24</td>
</tr>
<tr>
<td>4.7 Performance measures of Wavelet Compression - Calculated (Continued)</td>
<td>25</td>
</tr>
<tr>
<td>4.8 Performance measures of Wavelet Compression - Calculated (Continued)</td>
<td>25</td>
</tr>
<tr>
<td>4.9 Performance measures of Hermite Compression</td>
<td>26</td>
</tr>
<tr>
<td>4.10 Performance measures of Hermite Compression (Continued)</td>
<td>26</td>
</tr>
<tr>
<td>4.11 Performance measures of Hermite Compression (Continued)</td>
<td>27</td>
</tr>
<tr>
<td>4.12 Performance measures of Hermite Compression (Continued)</td>
<td>27</td>
</tr>
<tr>
<td>4.13 Performance measures of Hermite Compression - Calculated</td>
<td>28</td>
</tr>
<tr>
<td>4.14 Performance measures of Hermite Compression - Calculated (Continued)</td>
<td>28</td>
</tr>
<tr>
<td>4.15 Performance measures of Hermite Compression - Calculated (Continued)</td>
<td>29</td>
</tr>
<tr>
<td>4.16 Performance measures of Hermite Compression - Calculated (Continued)</td>
<td>29</td>
</tr>
<tr>
<td>4.17 Statistical Correlation Results - Select Cases</td>
<td>31</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>21</td>
</tr>
<tr>
<td>A.1</td>
<td>40</td>
</tr>
<tr>
<td>A.2</td>
<td>41</td>
</tr>
<tr>
<td>A.3</td>
<td>42</td>
</tr>
<tr>
<td>A.4</td>
<td>43</td>
</tr>
<tr>
<td>A.5</td>
<td>44</td>
</tr>
<tr>
<td>A.6</td>
<td>45</td>
</tr>
<tr>
<td>A.7</td>
<td>46</td>
</tr>
<tr>
<td>A.8</td>
<td>47</td>
</tr>
<tr>
<td>A.9</td>
<td>48</td>
</tr>
<tr>
<td>A.10</td>
<td>49</td>
</tr>
<tr>
<td>A.11</td>
<td>50</td>
</tr>
<tr>
<td>A.12</td>
<td>51</td>
</tr>
<tr>
<td>A.13</td>
<td>52</td>
</tr>
<tr>
<td>A.14</td>
<td>53</td>
</tr>
<tr>
<td>A.15</td>
<td>54</td>
</tr>
</tbody>
</table>
A.16 Results for thigh.bmp. .................................................. 55
A.17 Results for thorax.bmp. .................................................. 56
A.18 Results for PrintTest.bmp. .............................................. 57
A.19 Results for Tree.bmp. .................................................... 58
CHAPTER I
INTRODUCTION

Medicine today is quickly adopting digital information techniques for everything from record keeping to imaging. Having useful and inexpensive methods to minimize the storage of information and increase the ease of data exchange will aid to progress such efforts.

1.1 Purpose of Study

One-dimensional Hermite transforms for signal analysis have been applied to many signal processing cases. Examples of the applications of the one-dimensional Discrete Hermite Transform (DHmT) have been applied for ballistocardiogram artifact removal [1] and for heart arrhythmia analysis [2]. In a paper by Alexandre Refregier [3], two-dimensional continuous Hermite transforms were used in several image analysis applications and compression of galaxy images obtained from the Hubble Space Telescope [3]. Applying the two-dimensional DHmT to image compression is explored here as an alternative to current compression techniques such as those utilizing wavelets.
1.2 Significance of Study

Applying the two-dimensional DHmT will provide a new alternative method for image compression compared to techniques that rely upon wavelets. The original image will be compared to compressed versions created utilizing wavelets and Discrete Hermite Transforms. By using Mean Squared Error (MSE) and a new indicator of signal fidelity based upon structural similarity referred to as Structural Similarity Index (SSIM) [4], it will be investigated if there is a practical difference between utilizing wavelets and Discrete Hermite Transforms for medical image compression.

1.3 Specific Aims

The specific aims of the study were:

1. To develop methods for applying the Discrete Hermite Transform for the purpose of image compression

2. To compare the performance of such methods to established compression.

1.4 Statement Of Hypothesis

Hypotheses were formulated after initial literature review. From previous work several hypotheses were created for the investigation of the DHmT as an image compression technique.
1.4.1 Research Hypothesis

Dilated Discrete Hermite transforms can be used for the basis of algorithms that compress medical images with improvement of mean square error and structural similarity index compared to algorithms utilizing wavelets transforms.

1.4.2 Null Hypothesis

Utilizing dilated Discrete Hermite Transforms for medical image compression will not provide a means of medical image compression that is significantly different, via mean squared error and structural similarity index, than current wavelet techniques.
CHAPTER II
LITERATURE REVIEW

In the sections that follow, the background of Hermite functions are introduced. Both continuous and discrete Hermite functions are presented in detail for the understanding of the dilated Discrete Hermite Transform. Discussion of the Discrete Hermite Transform concludes the first section of this chapter. The second section describes the positive and negative features of Mean Squared Error when applied to image comparison for similarity. A newer technique, SSIM Index, for measuring image similarity is presented. This chapter ends with a description of the wavelet method applied for image comparison that was used in this study.

2.1 Hermite Transform Description

As Hermite functions are the basis of the analysis technique presented, this section describes, in detail, the theory of the continuous Hermite functions (CHf), the discrete Hermite functions (DHf), and their properties.

2.1.1 Continuous Hermite Functions

Continuous Hermite Functions originate from Hermite polynomials which are a set of orthogonal polynomials and can be defined in two different ways [5]. The first is a monic set of polynomials given by
\[ H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} \quad (2.1) \]

or the other form that results in the leading coefficient being a power of two:

\[ H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} \quad (2.2) \]

The Hermite polynomials satisfy a three-term recurrence relation of form

\[ H_{n+1}(x) = 2xH_n(x) - 2xH_{n-1}(x) \quad (2.3) \]

where \( H_0 = 1 \) and \( H_{-1} = 0 \). The CHf are defined as a normalized Gaussian multiple of the corresponding \( H_n(x) \) and in particular the CHf are defined for \( n \geq 0 \) as [5]:

\[ h(x) = \frac{1}{\pi^{1/4} \sqrt{2^n n!}} e^{-x^2} H_n(x) \quad (2.4) \]

The multiplication of the Hermite polynomials by the normalized Gaussian results in the CHf being of finite support, but as \( n \) is increased so does the length of the support. “These functions are orthonormal in the sense that” [5]

\[ \int_{-\infty}^{+\infty} h_n(x)h_m(x)dx = \delta(n - m) \quad (2.5) \]

“and that every L2 function has an expansion in terms of the CHf.” The last property of CHf to be considered is they are eigenfunctions of the Fourier transform as defined as \( F(h_n(x)) = (-i)^n h_n(x) \) where \( F \) is the integral Fourier transform.
2.1.2 Discrete Hermite Functions

Discrete Hermite functions (DHf) have analogous properties to the continuous functions. DHf are eigenvectors of the centered Fourier matrix such that

\[ F_c(h_k) = -i^k h_k \tag{2.6} \]

for \( k \geq 0 \) where \( F_c \) is the centered Fourier matrix and \( h_k \) is the \((k + 1)\)st eigenvector. Also, they form an orthonormal set of eigenvectors since they are eigenvectors of a related symmetric tridiagonal matrix [3]. This permits that every vector of length \( N \) can be expressed as a linear combination of the DHf [5] “This means that for a vector \( x \) of length \( N \), there is a representation

\[ x[n] = \sum_{k=0}^{N-1} c_k h_k[n], \tag{2.7} \]

for \( 0 \leq n \leq N - 1 \). Bracket notion is used here to indicate a discrete function of index [5].”

For multiscale applications used in signal and image processing [5], a dilated form can be generated and used. With the initial DHf being a discrete Gaussian, a parameter \( s \) can be used to define the dilation (with 1 used for the undilated case) and interpreted as the width of the Gaussian being similar to the standard deviation.

2.1.3 2D Discrete Hermite Transform

Image processing with two dimensional Discrete Hermite Transforms was proposed by J.B Martens. He suggested that Hermite polynomials could be multiplied by a Gaussian window in a similar fashion as he described for the one-dimensional case.
As such:

\[ u_{n-m,m}((x, y), \sigma) = g_{n-m}\left(-\frac{x}{\sigma}\right) g_{m}\left(-\frac{y}{\sigma}\right) \omega^2((x, y), \sigma) \]  

(2.8)

where

\[ g_n = \frac{1}{\sqrt{2^n n!}} H_n(x) \]  

(2.9)

when \((x, y)\) represent the two-dimensional coordinates, \(H_n(x)\) represent the hermite polynomials of order \(n\) and the Gaussian window defined as:

\[ \omega((x, y), \sigma) = \frac{1}{\sigma \sqrt{\pi}} e^{\frac{x^2+y^2}{2\sigma^2}} \]  

(2.10)

and \(\sigma\) represents the standard deviation of the Gaussian. Therefore, the two-dimensional representation of any signal given as \(f(x, y)\) is:

\[ f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij}(\sigma) u_{ij}((x, y), \sigma) \]  

(2.11)

where \(c_{ij}(\sigma)\) are the Hermite coefficients [6].

### 2.2 Measures of Image Fidelity

Determining the quality of image compression or signal integrity can be a subjective practice. In order to remove perceptual bias, that is to say that the viewer of an image might influence the measure of image quality, a standard measure and a new measure will be utilized. Mean Square Error (MSE) being a widely accepted standard measure for signal quality and used in the field of statistics is presented first as it applies to image comparison. Explanation of the a new measure, SSIM index, is presented after the discussion of MSE.
2.2.1 Mean Square Error

Mean Square Error (MSE) for signal integrity remains popular due to its history and ease of calculation [4]. The MSE is defined as the average of the square of the difference between two signals. Specifically:

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2$$  \hspace{1cm} (2.12)

or in the general form $l_p$ norm:

$$d_p(x, y) = \left( \sum_{i=1}^{N} |e_i|^p \right)^{\frac{1}{p}}$$  \hspace{1cm} (2.13)

and in image processing, this can take the the form of peak signal-to-noise ratio (PNSR):

$$PSNR = 10 \log_{10} \frac{L^2}{MSE}$$  \hspace{1cm} (2.14)

where $L$ is the dynamic range of allowable image pixel intensities.

MSE offers the following properties [4]:

1. It is simple, parameter free and easy to compute. It is also memoryless (calculated for each sample with needed other values)

2. Allows for consistent and direct interpretations of similarity:

   - Nonnegativity: $dp(x, y) \geq 0$
   - Identity: $dp(x, y) = 0$ if and only if $x = y$
   - Symmetry: $dp(x, y) = dp(y, x)$
   - Triangular inequality: $dp(x, z) \leq dp(x, y) + dp(y, z)$
3. MSE is energy preserving in the transform domain (if orthogonal linear transform is used) and indicates a clear physical meaning.

4. It is ideal in the context of optimization since it possesses the properties of convexity, symmetry, and differentiability.

5. When the difference is replaced with statistical expectation, it results in a form that the contribution for each source of distortion can be analyzed independently.

6. It has been established as a convention and is widely used for filter design, compression, restoration, denoising, reconstruction and classification. It is also often compared to other competing algorithms.

In Figure 2.1 on page 10, we see an example of how widely MSE can vary for several different types of image distortion: [4] image (a) is the original (b) mean contrast stretch, (c) luminance shift, (d) Gaussian noise, (e) Impulsive noise, (f) JPEG compression, (g) Blurring, (h) spatial scaling (zooming out), (i) spatial shift right, (j) spatial shift left, (k) rotation counter-clockwise and (l) rotation clockwise. “Note that the MSE values [relative to the original image(a)] are nearly identical [images (b) -(g)] even though the same images present dramatically different visual quality. Also notice that images that undergo small geometrical modification [images (h)- (i)] may have large MSE values relative to the original, yet show a negligible loss of perceived quality” [4].
However, as demonstrated by Bovik and Wang, MSE does not always coincide with the perception of visual quality. MSE has several implicit assumptions that do not hold up in the context of signal fidelity. MSE assumes implicitly [4] that:

1. Signal fidelity is independent of temporal and spatial relationships between samples of the original.

Figure 2.1: MSE and SSIM Index Comparison
2. Signal fidelity is independent of any relationship between the original signal and the error signal.

3. Signal fidelity is independent of the signs of the error signal samples.

4. And all samples are equally important to signal fidelity.

2.2.2 Measure of Structural Similarity of Images

There are alternatives to MSE for image fidelity; specifically the SSIM index. This measure stems from the fact that images are highly structured and possess strong neighbor dependencies which carry important information about the visual scene. SSIM was created based on the fact that the human visual system is highly adapted to extract structural information [Wang, Bovik] The SSIM index is computed by measuring the local luminances \( l(x, y) \), the similarity of local contrasts \( c(x, y) \), and the similarity of local structures \( s(x, y) \), among patches of two images being compared. These three computed statistics are combined to form the local SSIM. More formally the SSIM index \( S(x, y) \) is stated as [4]:

\[
S(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) = (\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}) \cdot (\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}) \cdot (\frac{2\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3}) \tag{2.15}
\]

where \( \mu_x \) and \( \mu_y \) are the local sample means of \( x \) and \( y \), \( \sigma_x \) and \( \sigma_y \) are the local sample standard deviations and \( \mu_{xy} \) is the sample cross correlation of \( x \) and \( y \) after removing their means. The constant terms \( C_1, C_2 \) and \( C_3 \) are used to stabilize each term in the event that sample means, variances and correlations are near zero.
Two properties of the SSIM index are that the index is symmetric and bounded. The symmetry provides that the same index value will be provided regardless of their ordering and being bound will produce a value between -1 and 1. The SSIM is computed by sliding a window pixel by pixel across the images producing an index map. The SSIM score is then averaged across the entire map.

One shortcoming of the SSIM index is that it is sensitive to relative translations, scaling and rotations. This is overcome by utilizing the complex wavelet SSIM (CW-SSIM). Here, two sets of wavelet coefficients are extracted at the same spatial location and then the SSIM index is calculated. The formula then becomes \[ \text{[4]} \]:

\[
\tilde{S}(c_x, c_y) = \tilde{m}(c_x, c_y) \cdot \tilde{p}(c_x, c_y)
\]

\[
= \frac{2 \sum_{i=1}^{N} |c_{x,i}| |c_{y,i}| + K}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + K} \cdot \frac{2 \sum_{i=1}^{N} c_{x,i}^* c_{y,i} + K}{2 \sum_{i=1}^{N} |c_{x,i}^* c_{y,i}| + K}
\]  \[ \text{[2.18]} \]

where \(c_x = \{c_{x,i}| i = 1, 2, \ldots, N\}\) and \(c_y = \{c_{y,i}| i = 1, 2, \ldots, N\}\) are the extracted coefficients of the two images being compared at the same spatial location and the same wavelet subband. \(c^*\) is the complex conjugate of \(c\) and \(K\) is a small positive constant used as a stabilizer.

\(\tilde{m}(c_x, c_y)\) represents the SSIM index calculated from the magnitudes of the wavelet coefficients and are 1 if and only if “\(|c_{x,i}| = |c_{y,i}|\).”

\(\tilde{p}(c_x, c_y)\) is determined by the consistency of changes between \(c_x\) and \(c_y\)” \[4\] the maximum value for \(\tilde{p}(c_x, c_y)\) occurs when the phase difference between \(c_{x,i}\) and \(c_{y,i}\) is constant for all \(i\).
2.2.3 Wavelet Compression

One of the applications of Wavelets has been for compressing images. Two very common methods of image compression that utilize wavelets are JPEG (Joint Photographic Experts Group) and the newer JPEG2000 and ASWDR [7]. There are several important features to wavelet compression and the above mentioned algorithms take advantage of these properties [7].

1. Progressive transmission and reconstructions can be performed on the compressed image. This is defined as the ability to transmit or reconstruct on parts of the image. The encoding and decoding process can be interrupted and resumed without loss of information. This is useful for analyzing only a portion of an image.

2. Can be lossless - the techniques used can create a smaller version for viewing quickly. Then the entire image can be recreated without loss of information.

3. Allows for the selection of a Region of Interest from the compressed image, then a detailed version can be recreated. This is due to the fact that the precise location information is recorded during compression.

Wavelet compression is an iterative process that produces a set of numbers that are less than the number of items in the original set. These numbers are what are stored, and used to recreate the image. Both the JPEG and the ASWDR methods utilize the Daub 5/3 wavelet for lossless compression. The JPEG standards used a
5-level (the process is repeated 5 times) Daub 9/7 wavelet transform during lossy compression. The Daub 9/7 will be detailed next.

Daub 9/7 analysis signals are defined by the numbers \( \alpha_1 \) through \( \alpha_9 \) and \( \beta_1 \) through \( \beta_7 \). From these numbers the scaling signal can be created as:

\[
V_k^1 = (0, \ldots, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, 0, \ldots, 0) \quad (2.19)
\]

and \( V_{k+1}^1 \) is a translation by two time units. The trend values are then calculated by

\[
A_k = f \cdot V_k^1 \quad (2.20)
\]

The Analysis Wavelet is defined as

\[
W_k^1 = (0, \ldots, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, 0, \ldots, 0) \quad (2.21)
\]

and \( W_{k+1}^1 \) is a translation by two time units. The fluctuations values are then calculated by

\[
d_k = f \cdot W_k^1 \quad (2.22)
\]

The resulting trend and fluctuation values are stored as the image. This wavelet is applied as if it were orthogonal, even though it is biorthogonal. On average, it is nearly energy preserving with only a 2 percent difference. This allows thresholding techniques to be applied in a similar manner to the lossless cases [7].
CHAPTER III
MATERIALS AND METHODS

This study was conducted to compare compression using the Discrete Hermite Transform with the compression using the widely accepted Wavelet transform for image compression for medical images.

3.1 Images Used

This study used images created by functional Magnetic Resonance Imaging (fMRI) obtain from the fMRI Data center website maintained by the University of California, Santa Barbara [8]. It also used MRI images obtained from the Visible Human project website maintained by the U.S. National Library of Medicine of the National Institutes of Health [9]. Lastly, several non-medical images were used in the comparison.

Image formats were selected so that the compression techniques under investigation in this study were not being applied to images that had been already compressed. Source images were of the bitmap format to reduce any unwanted effects of pre-processed images. The TIFF file format was used when saving images that had been reconstructed after processing.
3.2 Wavelet Processing

Images were imported into MatLab. Functions within MatLab were then used to transform the image, calculate a threshold, threshold the coefficients and reconstruct the image based on the subset of coefficients. The compression ratio, the mean square error and structural similarity index were then calculated.

3.2.1 Wavelet Standard processing

For this compression method, the threshold was calculated by finding the median of the absolute value of the details of the wavelet decomposition. This threshold calculation is one of the options in of the toolset provided by MatLab.

3.2.2 Wavelet Calculated Threshold

For the images presented in Appendix A labeled 'Calculated Threshold’, the threshold was obtained by finding the median of the absolute value of the all the coefficients of the wavelet decomposition. The purpose was to have a constant compression ratio similar to what was found for Hermite compression with calculated thresholds.

3.3 Hermite Processing

Images were imported into MatLab. Similar to the wavelet case, functions within MatLab were then used to transform the image, calculate a threshold, threshold the coefficients and reconstruct the image based on the subset of coefficients. The
Compression Ratio, the mean square error and structural similarity index were then calculated.

3.3.1 Hermite Constant Threshold

Based on the results of processing img10.bmp with wavelet compression, a threshold was found that produced a similar compression ratio when Hermite Processing was used. This threshold was then applied on all images in the study. The aim for selecting a threshold in this manner was strictly for aiding comparison between the two compression techniques.

3.3.2 Hermite Calculated Threshold

The wavelet method threshold was calculated by taking the median of a particular portion of coefficients. Similarly, the formula

\[ \text{Hermite}_{\text{thr}} = \text{median}(\text{abs(coefficients)}) \]  

was used for each image to obtain the threshold and then as stated above, the threshold was applied to the coefficients and the new subset of coefficients were used to reconstruct the image.

3.4 Statistical Analysis

Statistical analysis was performed on the results to determine how closely the two compression techniques performed. Compression based on the discrete Hermite transform performed was compared to wavelet based compression. Spearman Rho correlation was applied to two measures for each of two compression cases.
3.4.1 Wavelet Standard Processing vs. DHmT constant Threshold

The results obtained from applying wavelet standard processing (THR calculated by finding median of the absolute value of the details of the wavelet decomposition) and the discrete Hermite process utilizing a constant threshold were correlated using Spearman Rho correlation. Both mean square error (MSE) and structural similarity index (SSIM) were compared. MSE and SSIM values for all images were used in the correlation. Once the statistical analysis was performed, comparison of the P-VAL obtained to a value of less than 0.05 was used to determine if a significant correlation existed.

3.4.2 Wavelet Calculated Threshold vs. DHmT Calculated Threshold

Both MSE and SSIM values were correlated using Spearman Rho correlation obtained when the wavelet calculated threshold (the threshold was obtained by finding the median of the absolute value of all the wavelet decomposition coefficients) and the DHmT Calculated Threshold method (threshold calculated by finding the median of the absolute value of all the DHmT coefficients) were applied. MSE and SSIM values for all images that produced a resulting compression ratio of two (2) were used in the correlation. Once the statistical analysis was performed, comparison of the P-VAL obtained to a value of less than 0.05 was used to determine if a significant correlation existed.
CHAPTER IV
RESULTS

Results of processing the group of nineteen images over the four test cases as described in section 3.2.1 for Wavelet Standard, section 3.2.2 for Wavelet Calculated Threshold, section 3.3.1 for Hermite with Constant Threshold and section 3.3.2 for Hermite with Calculated Threshold are presented in tables at the end of this section. In the tables, CR is the abbreviation for Compression Ratio, MSE is the abbreviation for Mean Square Error, SSIM is the abbreviation used for Structure Similarity Index and THR is the abbreviation used to indicate the threshold used during coefficient threshold. As a guide, lower mean square error values indicate better performance and a SSIM closer to one indicates better performance. The resulting images are presented in Appendix A for comparison.

4.1 Trends Resulting Due to Image Features

Looking across the tables, as the amount of black space around the periphery of the image decreased, MSE increased, SSIM decreased and the compression ratio suffered. This held true for both Wavelet and Hermite Compression as presented in tables 4.1 through 4.4 for wavelets and tables 4.9 through 4.12 for Hermite compression. This is evident in images at the ends of the head scan sequence (images img10.bmp to
img30.bmp and imag90.bmp to img120.bmp) but when applied the middle images (img40 through img80) performed well and similarly.

4.2 Comparison of Constant Compression Ratio

Looking at tables 4.5 through 4.8 and tables 4.12 through 4.16, for images that resulted in a compression ratio of two (2), the wavelet compression outperforms the Hermite compression in both MSE and SSIM. This is evident in images img40.bmp through img90.bmp, abdomen.bmp, feet.bmp, pelvis.bmp, thigh.bmp, thorax.bmp, PrintTest.bmp and Tree.bmp.

4.3 Visual Image Quality

While it is left to the reader to truly determine any differences that might occur among the various compression methods, visually comparing the groups of five images reveals that all techniques produced similar results. As an example, all images for img10.bmp are provided in Figure 4.1 on page 21:
Figure 4.1: Results for img10.bmp.
### 4.4 Wavelet Compression

#### Table 4.1: Performance measures of Wavelet Compression

<table>
<thead>
<tr>
<th>File Name</th>
<th>img10.bmp</th>
<th>img20.bmp</th>
<th>img30.bmp</th>
<th>img40.bmp</th>
<th>img50.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>7.8075</td>
<td>5.9497</td>
<td>5.3997</td>
<td>4.7970</td>
<td>4.2413</td>
</tr>
<tr>
<td>MSE</td>
<td>12.9428</td>
<td>25.4700</td>
<td>43.9699</td>
<td>44.6884</td>
<td>648.7539</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9574</td>
<td>0.9394</td>
<td>0.9161</td>
<td>0.9025</td>
<td>0.9066</td>
</tr>
<tr>
<td>THR</td>
<td>8.6750</td>
<td>10.4500</td>
<td>12.7500</td>
<td>12.1750</td>
<td>12.1250</td>
</tr>
</tbody>
</table>

#### Table 4.2: Performance measures of Wavelet Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img60.bmp</th>
<th>img70.bmp</th>
<th>img80.bmp</th>
<th>img90.bmp</th>
<th>img100.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>3.9828</td>
<td>3.9238</td>
<td>4.7795</td>
<td>4.8588</td>
<td>5.2997</td>
</tr>
<tr>
<td>MSE</td>
<td>40.9040</td>
<td>41.3615</td>
<td>50.3573</td>
<td>49.1208</td>
<td>44.7519</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9191</td>
<td>0.9208</td>
<td>0.8873</td>
<td>0.9028</td>
<td>0.9095</td>
</tr>
<tr>
<td>THR</td>
<td>10.8500</td>
<td>10.8750</td>
<td>12.7500</td>
<td>12.7500</td>
<td>12.7500</td>
</tr>
</tbody>
</table>
Table 4.3: Performance measures of Wavelet Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img110.bmp</th>
<th>img120.bmp</th>
<th>abdomen.bmp</th>
<th>feet.bmp</th>
<th>pelvis.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>6.4188</td>
<td>7.9064</td>
<td>2.1235</td>
<td>1.5872</td>
<td>1.8872</td>
</tr>
<tr>
<td>MSE</td>
<td>30.334</td>
<td>9.4418</td>
<td>6.0568</td>
<td>0.1963</td>
<td>1.6354</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9200</td>
<td>0.9573</td>
<td>0.9774</td>
<td>0.9973</td>
<td>0.9892</td>
</tr>
<tr>
<td>THR</td>
<td>11.6000</td>
<td>7.5250</td>
<td>3.0000</td>
<td>0.5000</td>
<td>1.5000</td>
</tr>
</tbody>
</table>

Table 4.4: Performance measures of Wavelet Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>thigh.bmp</th>
<th>thorax.bmp</th>
<th>PrintTest.bmp</th>
<th>Tree.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.1191</td>
<td>3.1310</td>
<td>5.4805</td>
<td>1.8987</td>
</tr>
<tr>
<td>MSE</td>
<td>1.4892</td>
<td>4.2515</td>
<td>25.58193</td>
<td>225.1384</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9892</td>
<td>0.9796</td>
<td>0.9692</td>
<td>0.9031</td>
</tr>
<tr>
<td>THR</td>
<td>1.5000</td>
<td>2.5000</td>
<td>10.3500</td>
<td>18.0000</td>
</tr>
</tbody>
</table>
Table 4.5: Performance measures of Wavelet Compression - Calculated

<table>
<thead>
<tr>
<th>File Name</th>
<th>img10.bmp</th>
<th>img20.bmp</th>
<th>img30.bmp</th>
<th>img40.bmp</th>
<th>img50.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>3.2737</td>
<td>2.5013</td>
<td>2.0823</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>16.266e-21</td>
<td>1.3042e-20</td>
<td>1.72e-20</td>
<td>3.1823e-4</td>
<td>0.0041</td>
</tr>
<tr>
<td>SSIM</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9999</td>
</tr>
<tr>
<td>THR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2192</td>
<td>0.6059</td>
</tr>
</tbody>
</table>

Table 4.6: Performance measures of Wavelet Compression - Calculated (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img60.bmp</th>
<th>img70.bmp</th>
<th>img80.bmp</th>
<th>img90.bmp</th>
<th>img100.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0004</td>
<td>2.0000</td>
<td>2.0187</td>
</tr>
<tr>
<td>MSE</td>
<td>0.0038</td>
<td>0.0075</td>
<td>0.0063</td>
<td>4.2598e-4</td>
<td>1.6286e-20</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>THR</td>
<td>0.5787</td>
<td>0.7231</td>
<td>0.6860</td>
<td>25.1900</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 4.7: Performance measures of Wavelet Compression - Calculated (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img110.bmp</th>
<th>img120.bmp</th>
<th>abdomen.bmp</th>
<th>feet.bmp</th>
<th>pelvis.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.5374</td>
<td>3.0832</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>1.1610e-20</td>
<td>4.5617e-21</td>
<td>0.9279</td>
<td>0.0813</td>
<td>0.3662</td>
</tr>
<tr>
<td>SSIM</td>
<td>1</td>
<td>1</td>
<td>0.9945</td>
<td>0.9986</td>
<td>0.9962</td>
</tr>
<tr>
<td>THR</td>
<td>0</td>
<td>0</td>
<td>2.6836</td>
<td>0.7616</td>
<td>1.6700</td>
</tr>
</tbody>
</table>

Table 4.8: Performance measures of Wavelet Compression - Calculated (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>thigh.bmp</th>
<th>thorax.bmp</th>
<th>PrintTest.bmp</th>
<th>Tree.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>0.2393</td>
<td>0.7380</td>
<td>3.9185e-4</td>
<td>54.4101</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9971</td>
<td>0.9933</td>
<td>1</td>
<td>0.9755</td>
</tr>
<tr>
<td>THR</td>
<td>1.3470</td>
<td>2.2686</td>
<td>0.1299</td>
<td>19.5234</td>
</tr>
</tbody>
</table>
4.5 Hermite Compression

Table 4.9: Performance measures of Hermite Compression

<table>
<thead>
<tr>
<th>File Name</th>
<th>img10.bmp</th>
<th>img20.bmp</th>
<th>img30.bmp</th>
<th>img40.bmp</th>
<th>img50.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>8.6073</td>
<td>5.1799</td>
<td>3.7428</td>
<td>3.4366</td>
<td>3.0532</td>
</tr>
<tr>
<td>MSE</td>
<td>20.5488</td>
<td>22.4555</td>
<td>22.8512</td>
<td>22.9824</td>
<td>22.4457</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.8575</td>
<td>0.8302</td>
<td>0.8148</td>
<td>0.8221</td>
<td>0.8167</td>
</tr>
<tr>
<td>THR</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
</tbody>
</table>

Table 4.10: Performance measures of Hermite Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img60.bmp</th>
<th>img70.bmp</th>
<th>img80.bmp</th>
<th>img90.bmp</th>
<th>img100.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>3.2028</td>
<td>3.1452</td>
<td>3.2717</td>
<td>3.3230</td>
<td>3.5153</td>
</tr>
<tr>
<td>MSE</td>
<td>22.5372</td>
<td>22.7827</td>
<td>22.7408</td>
<td>22.7344</td>
<td>422.4543</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.8194</td>
<td>0.8272</td>
<td>0.8240</td>
<td>0.8174</td>
<td>0.8042</td>
</tr>
<tr>
<td>THR</td>
<td>0.0050</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
</tbody>
</table>
Table 4.11: Performance measures of Hermite Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img110.bmp</th>
<th>img120.bmp</th>
<th>abdomen.bmp</th>
<th>feet.bmp</th>
<th>pelvis.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>4.9536</td>
<td>9.9372</td>
<td>2.9057</td>
<td>4.3621</td>
<td>3.1760</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.8199</td>
<td>0.8677</td>
<td>0.9508</td>
<td>0.9276</td>
<td>0.9465</td>
</tr>
<tr>
<td>THR</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
</tbody>
</table>

Table 4.12: Performance measures of Hermite Compression (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>thigh.bmp</th>
<th>thorax.bmp</th>
<th>PrintTest.bmp</th>
<th>Tree.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>3.7026</td>
<td>2.8324</td>
<td>1.7435</td>
<td>1.3610</td>
</tr>
<tr>
<td>MSE</td>
<td>14.4398</td>
<td>11.4870</td>
<td>15.5499</td>
<td>13.7147</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9327</td>
<td>0.9470</td>
<td>0.9215</td>
<td>0.9940</td>
</tr>
<tr>
<td>THR</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
</tbody>
</table>
Table 4.13: Performance measures of Hermite Compression - Calculated

<table>
<thead>
<tr>
<th>File Name</th>
<th>img10.bmp</th>
<th>img20.bmp</th>
<th>img30.bmp</th>
<th>img40.bmp</th>
<th>img50.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.000</td>
</tr>
<tr>
<td>MSE</td>
<td>1.5515</td>
<td>2.8759</td>
<td>4.8208</td>
<td>5.6692</td>
<td>6.9268</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9587</td>
<td>0.9335</td>
<td>0.9123</td>
<td>0.9006</td>
<td>0.8959</td>
</tr>
<tr>
<td>THR</td>
<td>0.0135</td>
<td>0.0208</td>
<td>0.0263</td>
<td>0.0281</td>
<td>0.0311</td>
</tr>
</tbody>
</table>

Table 4.14: Performance measures of Hermite Compression - Calculated(Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img60.bmp</th>
<th>img70.bmp</th>
<th>img80.bmp</th>
<th>img90.bmp</th>
<th>img100.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>6.3199</td>
<td>6.6385</td>
<td>6.1225</td>
<td>5.9460</td>
<td>5.2699</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9043</td>
<td>0.9010</td>
<td>0.0982</td>
<td>0.9052</td>
<td>0.9052</td>
</tr>
<tr>
<td>THR</td>
<td>0.0299</td>
<td>0.0304</td>
<td>0.0292</td>
<td>0.0289</td>
<td>0.0275</td>
</tr>
</tbody>
</table>
Table 4.15: Performance measures of Hermite Compression - Calculated (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>img110.bmp</th>
<th>img120.bmp</th>
<th>abdomen.bmp</th>
<th>feet.bmp</th>
<th>pelvis.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>3.2242</td>
<td>1.2405</td>
<td>2.2431</td>
<td>1.5987</td>
<td>1.7330</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9279</td>
<td>0.9652</td>
<td>0.9904</td>
<td>0.9828</td>
<td>0.9892</td>
</tr>
<tr>
<td>THR</td>
<td>0.0218</td>
<td>0.0138</td>
<td>0.0194</td>
<td>0.0149</td>
<td>0.0159</td>
</tr>
</tbody>
</table>

Table 4.16: Performance measures of Hermite Compression - Calculated(Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>thigh.bmp</th>
<th>thorax.bmp</th>
<th>PrintTest.bmp</th>
<th>Tree.bmp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>MSE</td>
<td>1.9108</td>
<td>2.2380</td>
<td>32.7358</td>
<td>105.6741</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9892</td>
<td>0.9640</td>
<td>0.8727</td>
<td>0.9581</td>
</tr>
<tr>
<td>THR</td>
<td>0.1640</td>
<td>0.0185</td>
<td>0.0726</td>
<td>0.1088</td>
</tr>
</tbody>
</table>
4.6 Statistical Correlation

Statistical analysis was performed to identify possible correlation among wavelet and DHmT processing techniques for two cases each.

The first case investigates if a correlation exists between wavelet standard Processing (THR calculated by finding median of the absolute value of the details of the wavelet decomposition) and the discrete Hermite process utilizing a constant threshold. MSE and SSIM values for all images were used.

The second case identifies if a correlation between the wavelet calculated threshold (the threshold was obtained by finding the median of the absolute value of all the wavelet decomposition coefficients) with the DHmT Calculated Threshold method (threshold calculated by finding the median of the absolute value of all the DHmT coefficients).

Spearman-Rho correlation was used and the results are listed in table 4.17 on page 31.
<table>
<thead>
<tr>
<th>Parameter Correlated</th>
<th>P-VAL</th>
<th>Statistical Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet Std vs DHmT Const. THR - MSE</td>
<td>0.0141</td>
<td>Significant Correlation</td>
</tr>
<tr>
<td>Wavelet Std vs DHmT Const. THR - SSIM</td>
<td>0.0052</td>
<td>Significant Correlation</td>
</tr>
<tr>
<td>Wavelet vs DHmT w/ Calc. THR - MSE</td>
<td>0.3733</td>
<td>No Significant Correlation</td>
</tr>
<tr>
<td>Wavelet vs DHmT w/ Calc. THR - SSIM</td>
<td>0.0026</td>
<td>Significant Correlation</td>
</tr>
</tbody>
</table>
CHAPTER V
DISCUSSION AND CONCLUSIONS

With this being a first study in utilizing the discrete Hermite Transform for image compression, the choice to compare DHmT to the widely accepted wavelet technique was an obvious choice. Since digital images are now the most prevalent, the idea of using a discrete Hermite transform ([1] Mahadevan et al) is much more important than the continuous Hermite functions used in ([3] Refregier). The 2D DHmT also allows the ability to analyze and compress larger images than in [3]. However there were a few shortcomings when existing measures are applied for comparing images. A measure of structure similarity was used as another reference point beyond mean square error to aid in evaluating the effectiveness of each compression method. Statistical correlation between wavelet and DHmT was performed to help identify if the two methods did produce similar results. And as a check against the numerical and statistical analysis, images are presented for visual comparison.

Source images were carefully selected for format. Images in a format that were compressed as little as possible were used to limit the possible influence, either positive or negative, to the compression techniques being compared in this study.

Images were selected to demonstrate performance of the compression techniques over a range of image structures. The images used range from ones containing
many transitions of light to dark (image tree.bmp and the various body images) to ones that are substantially single colored (head images img10.bmp to img40.bmp).

Methods of calculating the threshold to reduce sets of coefficients were chosen to be as simple as possible. From the tools used, a simple one for wavelets was the one primarily used in this study. This calculation was used for the Hermite processing to be consistent. And an additional test case was evaluated that utilized the same threshold calculation for both wavelets and DHmT processes. This case uses the median of the absolute value of all coefficients produced by each technique.

Wavelets, however are not limited to just using such a thresholding technique. The JPEG standard utilizes complex stages surrounding the wavelet decomposition of the image. For example, pre-processing for tiling (breaking into smaller regions), quantization (scaling the coefficients) and coding (quantized coefficients within each subband are grouped into rectangular blocks (codeblocks), which are coded independently by an adaptive binary arithmetic coder and Tier-2 coding, where the coded data is organized in packets using a rate control). [10]. In the paper by Cobas and his team, they also investigate performance compression and to overcome this they use peak signal-to-noise ratio as a quality measurement (a measure that utilized MSE), and a probability distribution also based on MSE to help identify how far to compress an image without sacrificing image quality.

With the complexity and efficiency already established by the JPEG standard one would naturally ask why investigate another compression method. However, there are many applications where image compression is used but can hinder performance.
One such application is indexing of images for web-indexing such as described by J. Jiang [11] where a new algorithm was investigated that allowed for only partial reconstruction of the image prior to indexing. The aim was to improve performance by not performing the entire reconstruction and still have strong indexing performance. By establishing DHmT for image compression and in conjunction with the strong ability to perform correlation as described in the thesis by Srinivasan [12], DHmT could be used for both the compression and indexing applications.

The significance of this study lies in the fact that no compression technique is perfect for all applications. Despite being a first study investigating the viability of applying the discrete Hermite transform to image compression, the findings of the study demonstrate that DHmT could be used for image compression. This will hopefully inspire other investigators to seek out other applications for DHmT in the realm of signal and image processing.

5.1 Conclusion

Overall, the Hermite based compression compared favorably to the accepted Wavelet method. This can be seen by the trend of the Hermite techniques that followed the performance of the wavelet techniques. The results of statistical correlation reveal that for the standard wavelet method correlated to the constant threshold DHmT method there were significant correlations for both the MSE and SSIM measures. However, for wavelet and DHmT with calculated threshold, there was a significant correlation for MSE but not for SSIM. This demonstrates that wavelets and DHmT
compare favorably. Since wavelet and DHmT utilizing similar thresholding methods were correlated across two measures and resulted in a significant correlation, it can be concluded that the performance of wavelets and DHmT trend, or follow across the image set used, in a manner that is very similar to each other.

Further evidence that wavelets and DHmT will produce similar and effective results can be seen when the wavelet performance decreased (indicated by a higher MSE and a lower SSIM) the Hermite based compression techniques followed that behavior. When the parameter of compression ratio was held constant, wavelet techniques did outperform the Hermite techniques, but after visually comparing all techniques, both techniques produced acceptable results.

5.2 Future Scope

Much effort has been expended in perfecting Wavelet-based compression techniques that Hermite based techniques have yet to benefit from. An investigation to see if a refined threshold selection process for Hermite based compression utilizing some of the complex statistical processes employed by wavelets (evidenced by the Joint Photographic Expert Group standard (JPEG)) would bring Hermite compression closer to JPEG performance would be a next logical step.

Dilation parameters were held constant for all Hermite compression performed in this study. With the successful results obtained in this study, an exploration and development in selecting the best dilation parameter to achieve the best compression results could also be an area to be explored.
BIBLIOGRAPHY


37
APPENDIX A

IMAGES
Figure A.1: Results for img10.bmp.
Figure A.2: Results for img20.bmp.
Figure A.3: Results for img30.bmp.
Figure A.4: Results for img40.bmp.
Figure A.5: Results for img50.bmp.
Figure A.6: Results for img60.bmp.
Figure A.7: Results for img70.bmp.
Figure A.8: Results for img80.bmp.
Figure A.9: Results for img90.bmp.
Figure A.10: Results for img100.bmp.
Figure A.11: Results for img110.bmp.
Figure A.12: Results for img120.bmp.
Figure A.13: Results for abdomen.bmp.
Figure A.14: Results for feet.bmp.
Figure A.15: Results for pelvis.bmp.
Figure A.16: Results for thigh.bmp.
Figure A.17: Results for thorax.bmp.
Figure A.18: Results for PrintTest.bmp.
Figure A.19: Results for Tree.bmp.
APPENDIX B

SOURCE CODE

HermCompr2.m

function [ thr2, I3, ssim1, hermiteSz, origSz, CR, MSE, img0] =
    HermCompr2(X, outfile, thr)
% HermCompr2 is a function for performing the two-dimensional discrete
% Hermite Transform on a grayscale image of size 256 x 256.
% The function
% then reduces the number of coefficients by applying a threshold. With the
% reduced set of coefficients, the image is reconstructed and displayed
% next to the original and reconstructed images is saved.
% Compression
% Ratio, MSE, and a measure of image fidelity is calculated.
%
59
% USAGE 1
%
%  [I3, ssim1, hermiteSz, origSz, CR, MSE, img0] = HermCompr2(X, outfil, thr)
%    Uses the user specified value for thresholding.
%
% USAGE 2
%
%  [I3, ssim1, hermiteSz, origSz, CR, MSE, img0] = HermCompr2(X, outfil)
%    Calculates the threshold using the formula:
%    thr = median(abs(Hermite Coefficients))
%
% INPUTS
%
%  X: is the input file to be compressed.
%  outfil: is the name of the reconstructed image to be saved.
%  thr (optional): value to be used as the threshold.
%    If not specified, then the value is calculated.
%
% OUTPUTS
%
%  I3: Reconstructed image values
%  ssim1: SSIM Index between original and I3
% hermiteSz: Number of non-zero values of Hermite coefficients after THR
% origSz: Number of elements in original image
% CR: Compression Ratio = 1/(hermiteSz/origSz)
% MSE: Mean Square Error
% img0: Original Image elements.

img0=imread(X);

% some constants
n=256; sig=1.2; s=sig; % sigma relates to FHT width
img=im2double(img0);

% Compute the 2-D DHnT

Hcs=genhchssig(n,s); % Generates "half" length

% The 2-D transform is rows first, replace-then column transforms
I2 = img;

% columns first
for j = 1:n
    I2(:,j) = fHcs(img(:,j), Hcs); % see fhtv for fast Hermite transform
end

for k = 1:n % for row transforms
    I3(k,:) = fHcs(I2(k,:), Hcs);
end

C2 = I3;

I4 = zeros(n, n);

% Set threshold and filter out the "noise"
switch nargin
    case 3
        thr2 = thr;
    case 2
        thr2 = median(abs(reshape(C2, 1, numel(C2))));
    otherwise
        thr2 = 0;
end

for j = 1:n
for k=1:n
    if abs(C2(j,k))>thr2
        I4(j,k)=C2(j,k);
    end
end
end

C3=I4;

save('DHmtCoeff.mat','C3');

% Inverse transform filtered version
I2=img;

% columns first
for j=1:n
    I2(:,j)=ifHcs(I4(:,j),Hcs);
end
for k=1:n % for row transforms
    I3(k,:)=ifHcs(I2(k,:),Hcs);
end
% C2=I3;

figure;

subplot(1,2,1); imshow(img0,[]);
axis square;
title('Hermite Original Image');
subplot(1,2,2); imshow(I3,[]);
axis square;
title('Hermite Compressed Image');

i5=abs(I4)>0.25;
i5mod=i5.*i5;

figure(3);
surfc(i5mod); % figure(gcf) % note position of the larger transform values
title('Hermite Coefficients');

imwrite(I3, outfile);
I5=im2uint8(I3);

origSz = numel(img0);
hermiteSz = nnz(C3);
CR=1/(hermiteSz/origSz);

% Calculate MSE (mean square error)
mseImage = (double(img0) - double(I5)) ^ 2;
MSE = (sum(mseImage(:))) / origSz;
[ssim1 map]=ssim_index(img0,I5);
function [ Xrecon , thr , origSz , CthSz , CR , Cth , MSE , ssim1 , img , C ] = wavecomp2 ( X , lv1 )

% wavecomp2 is a funcion for performing the two–dimensional discrete
% Hermite Transform on a grayscale image of size 256 x 256.
% The function
% then reduces the number of coefficients by apply a threshold. With the
% reduced set of coefficients, the image is reconstructed and displayed
% next to the original and reconstructed images is saved.
% Compression
% Ratio, MSE, and a measure of image fidelity is calculated.
% Source is
% from MATLAB wavelet toolbox help for 2–d discrete wavelet.

% USAGE

% [Xrecon, thr, origSz, CthSz, CR, Cth, MSE, ssim1, img, C] = wavecomp2(X, lv1)

%
% INPUTS:

% X – this is the image file to be processed
% lvl – Number of levels of wavelet decomposition to be performed

% OUTPUTS

% Xrecon – reconstructed image values
% thr – Threshold value used
% origSz – Size of original image data
% CthSz – Size of thresholded coefficients
% CR – Compression Ratio
% Cth – Thresholded coefficients
% MSE – Mean Square Error
% ssim1 – SSIM Index between original and reconstructed
% img
% C

% analysis

img=imread(X);
wavemode=dwtmode('per');

% multi-level wavelet decomposition
[C,L] = wavedec2(img,lvl,'bior4.4');

% calculate threshold by thr = median(abs(detail at level 1))
[thr,sorh,keepapp] = ddencmp('cmp','wv',img);

% Hard threshold and compression based on above calculation
[xt,dxt,ldx,perf0,perf12] = wdencomp('gbl',C,L,'bior4.4',lvl,thr,'h',1);
Cth = wthresh(C,'s',thr);

% reconstruct the image after thresholding has been applied
Xrecon = waverec2(Cth,L,'bior4.4');
figure(1);
colormap(gray);
imagesc(Xrecon);

%
% calculate Number of original size, coefficients still remaining and ratio
% original size is needed to maintain image

CR = 1/(CthSz/origSz);

% Calculate MSE, mean square error.

MSE = (sum(mseImage(:))) / origSz;

[ssim1 map] = ssim_index(img, Xrecon);
function [Xrecon, thr, origSz, CthSz, CR, Cth, MSE, ssim1, img, C] = wavecomp3(X, lv1)

wavecomp2 is a function for performing the two-dimensional discrete
Hermite Transform on a grayscale image of size 256 x 256. The function
then reduces the number of coefficients by applying a threshold. With the
reduced set of coefficients, the image is reconstructed and displayed.
This function uses all coefficients to calculate threshold. Compression
Ratio, MSE, and a measure of image fidelity is calculated.
Source is from MATLAB wavelet toolbox help for 2-d discrete wavelet.

%USAGE

[Xrecon, thr, origSz, CthSz, CR, Cth, MSE, ssim1, img, C] = 
wavecomp2(X, lv1)
%

% INPUTS:

% X – this is the image file to be processed

% lvl – Number of levels of wavelet decomposition to be performed

%

% OUTPUTS

% Xrecon – reconstructed image values

% thr – Threshold value used

% origSz – Size of original image data

% CthSz – Size of thresholded coefficients

% CR – Compression Ratio

% Cth – Thresholded coefficients

% MSE – Mean Square Error

% ssim1 – SSIM Index between original and reconstructed Image

% img – imported image data

% C – Coefficients of Wavelet decomposition

%
% analysis

img=imread(X);
wavemode=dwtmode('per');

% multi-level wavelet decomposition

[C,L] = wavedec2(img,lvl,'bior4.4');

% calculate threshold

thr = median(abs(C));%myTHR

% Hard threshold and compresion based on above calculation

%Cth = wthresh(C,'s',thr);

n=length(C);
Cth=zeros(1,n);
for j=1:n
    if abs(C(j))>thr
        Cth(j)=C(j);
    end
end

% reconstruct the image after thresholding has been applied

Xrecon = waverec2(Cth,L,'bior4.4');
figure(1);
colormap(gray);
imagesc(Xrecon);

%
%calculate Number of original size, coefficients still remaining and ratio
% original size is needed to maintain image
%
origSz = numel(img);
CthSz = sum(Cth~=0);
CR=1/(CthSz/origSz);

% Calculate MSE, mean square error.
mseImage = (double(img) - double(Xrecon)).^2;
MSE = (sum(mseImage(:)) / origSz);
[ssim1 map]=ssim_index(img,Xrecon);
% genhchssig.m

% Generation of the hermite cosine and sine functions, built on genhchsb

% dec 14 '07  Output is the Hcs matrix with Hc and Hs submatrices whose
% columns are the Hermite cosine/sine vectors.
% Value of n to use is the double one

function Hcs=genhchssig(n,sigma)

% sigma is the dilation parameter
r=1/sigma^2; %Ta=zeros(n,n);
xset=n/2:n-1;
x=-2*cos(pi*r)*sin(pi*r/n*xset).*sin(pi*r-pi*(xset+1)*r/n);
yset=n/2+1:n-1;
y=sin(pi*yset*r/n).*sin(pi*(n-yset)*r/n);
Ta=diag(x)+diag(y,-1)+diag(y,1);
Tc=Ta; Ts=Ta;  % Ta is the lower right corner of the general
T matrix, and it is

% used for both cosine and sine vectors.
% for Tc generation
\[ T_c(1,1) = \text{Ta}(1,1) + (\sin(\pi/(2*\sigma^2)))^2; \]

\[ [V_c, D_c] = \text{eig}(T_c); \quad d_c = \text{diag}(D_c); \]

\[ [v_c, I_c] = \text{sort}(d_c); \quad d_c = d_c(I_c); \]

\[ V_{c2} = V_c(:, I_c); \]

\[ m = n/2; \]

\[ \text{for j = 1:m} \]
\[ \quad \text{if } \max(V_{c2}(:, j)) + \min(V_{c2}(:, j)) < 0 \]
\[ \quad \quad V_{c2}(:, j) = -V_{c2}(:, j); \]
\[ \quad \text{end} \]
\[ \text{end} \]

\[ H_c = \text{fliplr}(V_{c2}); \quad \%/\sqrt{2}; \]

% for Ts generation
\[ T_s(1,1) = \text{Ta}(1,1) - (\sin(\pi/(2*\sigma^2)))^2; \]

\[ [V_s, D_s] = \text{eig}(T_s); \]

\[ d_s = \text{diag}(D_s); \]

\[ [v_s, I_s] = \text{sort}(d_s); \]

\[ d_s = d_s(I_s); \]

\[ V_{s2} = V_s(:, I_s); \]
for  j=1:m
    if  max(Vs2(:,j))+min(Vs2(:,j))<0
        Vs2(:,j)=-Vs2(:,j);
    end
end
Hs=fliplr(Vs2);  %/sqrt(2);
Hcs=[Hc, Hs];
ifHcs.m

% ifHcs.m built on ifht.m. Incorporates Hcs=[Hc,Hs]
% a 'fast' inverse Hermite transform
% dec 13 07 (but we'd like to make it faster)

function x=ifHcs(c,Hcs)
n=length(c); x=zeros(n,1); srt=sqrt(2);
fHc=flipud(Hcs(:,1:n/2)); fHs=flipud(Hcs(:,n/2+1:n));

% cp=c(1:2:n); cm=c(2:2:n);
xp=fHc*cp; xp=xp/srt;
XM=fHs*cm; XM=XM/srt;

xa=(xp-XM);
xb=(xp+XM);

x=[xa; flipud(xb)];

% Note: The sqrt(2) divider is to make the V matrix
% vectors of same value as Hc,Hs.
function [mssim, ssim_map] = ssim_index(img1, img2, K, window, L)

% SSIM Index, Version 1.0
% Copyright (c) 2003 Zhou Wang
% All Rights Reserved.
%
% The author was with Howard Hughes Medical Institute, and Laboratory
% for Computational Vision at Center for Neural Science and Courant
% Institute of Mathematical Sciences, New York University, USA. He is
% currently with Department of Electrical and Computer Engineering,
% University of Waterloo, Canada.
%
78
Permission to use, copy, or modify this software and its documentation for educational and research purposes only and without fee is hereby granted, provided that this copyright notice and the original authors’ names appear on all copies and supporting documentation.

This program shall not be used, rewritten, or adapted as the basis of a commercial software or hardware product without first obtaining permission of the authors. The authors make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
This is an implementation of the algorithm for calculating the Structural SIMilarity (SSIM) index between two images. Please refer to the following paper:


Kindly report any suggestions or corrections to zhouwang@ieee.org

Input: (1) img1: the first image being compared
(2) img2: the second image being compared
(3) K: constants in the SSIM index formula (see the above reference). Default value: \( K = [0.01 \ 0.03] \)
(4) window: local window for statistics (see the above reference). Default window is Gaussian given by
\[
\text{window} = \text{fspecial('gaussian', 11, 1.5)};
\]
(5) L: dynamic range of the images. Default: \( L = 255 \)

Output: (1) mssim: the mean SSIM index value between 2 images.

If one of the images being compared is regarded as perfect quality, then mssim can be considered as the quality measure of the other image.

If \( \text{img1} = \text{img2} \), then mssim = 1.

(2) ssim_map: the SSIM index map of the test image. The map
has a smaller size than the input images. The actual size:

\[ \text{size}(\text{img1}) - \text{size}(\text{window}) + 1. \]

% Default Usage:
% Given 2 test images img1 and img2, whose dynamic range is 0–255
%
% [mssim ssim_map] = ssim_index(img1, img2);
%
% Advanced Usage:
% User defined parameters. For example
%
% K = [0.05 0.05];
% window = ones(8);
% L = 100;
% [mssim ssim_map] = ssim_index(img1, img2, K, window, L);
%
% See the results:
%
% mssim % Gives the mssim value
if (nargin < 2 | nargin > 5)
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end

if (size(img1) ~= size(img2))
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end

[M N] = size(img1);

if (nargin == 2)
if ((M < 11) | (N < 11))
    mssim = -Inf;
    ssim_map = -Inf;
    return
end

window = fspecial('gaussian', 11, 1.5); %
K(1) = 0.01;
    % default settings
K(2) = 0.03;
    %
L = 255; %
end

if (nargin == 3)
    if ((M < 11) | (N < 11))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
    window = fspecial('gaussian', 11, 1.5);
    L = 255;
end
if (length(K) == 2)
    if (K(1) < 0 | K(2) < 0)
        mssim = -Inf;
        ssim_map = -Inf;
        return;
    end
else
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end
end

if (nargin == 4)
    [H W] = size(window);
    if ((H*W) < 4 | (H > M) | (W > N))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
end
L = 255;
if (length(K) == 2)
    if (K(1) < 0 | K(2) < 0)
        mssim = -Inf;
        ssim_map = -Inf;
        return;
    end
else
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end

if (nargin == 5)
    [H W] = size(window);
    if ((H*W) < 4 | (H > M) | (W > N))
        mssim = -Inf;
        ssim_map = -Inf;
        return
    end
    if (length(K) == 2)
if (K(1) < 0 | K(2) < 0)
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end
else
    mssim = -Inf;
    ssim_map = -Inf;
    return;
end
end

C1 = (K(1)*L)^2;
C2 = (K(2)*L)^2;
window = window/sum(sum(window));
img1 = double(img1);
img2 = double(img2);

mu1  = filter2(window, img1, 'valid');
mu2  = filter2(window, img2, 'valid');
mu1_sq = mul.*mul;
\[
\begin{align*}
\mu_2 \cdot \mu_2 &= \mu_2 \cdot \mu_2; \\
\mu_1 \cdot \mu_2 &= \mu_1 \cdot \mu_2; \\
\sigma_1 \cdot \sigma_2 &= \text{filter2}(\text{window}, \text{img1} \cdot \text{img1}, \text{'valid'}) - \mu_1 \cdot \mu_2; \\
\sigma_2 \cdot \sigma_2 &= \text{filter2}(\text{window}, \text{img2} \cdot \text{img2}, \text{'valid'}) - \mu_2 \cdot \mu_2; \\
\sigma_1 \cdot \sigma_2 &= \text{filter2}(\text{window}, \text{img1} \cdot \text{img2}, \text{'valid'}) - \mu_1 \cdot \mu_2; \\
\text{if} \quad (C_1 > 0 \& C_2 > 0) \\
\quad \text{ssimMap} &= ((2 \cdot \mu_1 \cdot \mu_2 + C_1) \cdot (2 \cdot \sigma_1 \cdot \sigma_2 + C_2)) / ((\mu_1 \cdot \mu_2 + \mu_2 \cdot \mu_2 + C_1) \cdot \sigma_1 \cdot \sigma_2 + C_2); \\
\text{else} \\
\quad \text{numerator1} &= 2 \cdot \mu_1 \cdot \mu_2 + C_1; \\
\quad \text{numerator2} &= 2 \cdot \sigma_1 \cdot \sigma_2 + C_2; \\
\quad \text{denominator1} &= \mu_1 \cdot \mu_2 + \mu_2 \cdot \mu_2 + C_1; \\
\quad \text{denominator2} &= \sigma_1 \cdot \sigma_2 + \sigma_2 \cdot \sigma_2 + C_2; \\
\quad \text{ssimMap} &= \text{ones}(\text{size}(\mu_1)); \\
\quad \text{index} &= (\text{denominator1} \cdot \text{denominator2} > 0); \\
\quad \text{ssimMap} \cdot \text{index} &= (\text{numerator1} \cdot \text{index}) \cdot (\text{numerator2} \cdot \text{index}) \\
\quad &/ (\text{denominator1} \cdot \text{index}) \cdot (\text{denominator2} \cdot \text{index}); \\
\quad \text{index} &= (\text{denominator1} \neq 0) \& (\text{denominator2} == 0); \\
\quad \text{ssimMap} \cdot \text{index} &= \text{numerator1} \cdot \text{index} / \text{denominator1} \cdot \text{index}; \\
\text{end}
\end{align*}
\]
mssim = mean2(ssim_map);

return
function [StdmmseRho, StdmmsePVAL, StdsisimRho, StdsisimPVAL, 
THRmseRho, THRmsePVAL, THRssimRho, THRssimPVAL] = 
myPearsonRho(stdmsein, stdssimin, THRmse, THRssim) 

% myPearsonRho is a function for performing correlation among 
% two datasets 

% % USAGE 
% [CorrStdmmse, CorrStdsisim, CorrTHRmse, CorrTHRssim] = 
% myPearsonRho('stdmsein', 'stdssimin', 'THRmse', 'THRssim') 
% 
% % INPUTS: m x 2 ASCII tab-delimited data files containing: 
% % stdmsein – mse values for technique 1 
% % stdssimin – SSIM values for technique 1 
% % THRmse – mse values for technique 2 
% % THRssim – SSIM values for technique 2 
% 
% % OUTPUTS 
% % StdmmseRho – Rho values for technique 1 
% % StdmmsePVAL – P-VAL values for technique 1
% StdsisimRho - Rho for technique 1
% StdsisimPVAL - P-VAL for technique 1
% THRmseRho - Rho for technique 2
% THRmsePVAL - P-VAL for technique 2
% THRssimRho - Rho for technique 2
% THRssimPVAL - P-VAL for technique 2

% Load the data sets
stdmseinL = load('stdmsein');
stdssiminL = load('stdssimin');
THRmseL = load('THRmse');
THRssimL = load('THRssim');

% Compute Rho and P-VAL
[StdmmseRho, StdmmsePVAL] = corr(stdmseinL,'type','spearman');
[StdssimRho, StdssimPVAL] = corr(stdssiminL,'type','spearman');
[THRmseRho, THRmsePVAL] = corr(THRmseL,'type','spearman');
[THRssimRho, THRssimPVAL] = corr(THRssimL,'type','spearman');
%The End