Wavelet Image Compression

Myung-Sin Song

This paper is dedicated to my advisor.

Abstract. Wavelets provide a powerful and remarkably flexible set of tools for handling fundamental problems in science and engineering, such as audio de-noising, signal compression, object detection and fingerprint compression, image de-noising, image enhancement, image recognition, diagnostic heart trouble and speech recognition, to name a few. Here, we are going to concentrate on wavelet application in the field of Image Compression so as to observe how wavelet is implemented to be applied to an image in the process of compression, and also how mathematical aspects of wavelet affect the compression process and the results of it. Wavelet image compression is performed with various known wavelets with different mathematical properties. We study the insights of how wavelets in mathematics are implemented in a way to fit the engineering model of image compression.

1. Introduction

Wavelets are functions which allow data analysis of signals or images, according to scales or resolutions. The processing of signals by wavelet algorithms in fact works much the same way the human eye does; or the way a digital camera processes visual scales of resolutions, and intermediate details. But the same principle also captures cell phone signals, and even digitized color images used in medicine. Wavelets are of real use in these areas, for example in approximating data with sharp discontinuities such as choppy signals, or pictures with lots of edges.

While wavelets is perhaps a chapter in function theory, we show that the algorithms that result are key to the processing of numbers, or more precisely of digitized information, signals, time series, still-images, movies, color images, etc. Thus, applications of the wavelet idea include big parts of signal and image processing, data compression, fingerprint encoding, and many other fields of science and engineering. This thesis focuses on the processing of color images with the use of custom designed wavelet algorithms, and mathematical threshold filters.

Although there have been a number of recent papers on the operator theory of wavelets, there is a need for a tutorial which explains some applied tends from

1991 *Mathematics Subject Classification.* Primary 42C40.

Key words and phrases. Wavelet Analysis, Image Compression, Harmonic Analysis.

scratch to operator theorists. Wavelets as a subject is highly interdisciplinary and it draws in crucial ways on ideas from the outside world. We aim to outline various connections between Hilbert space geometry and image processing. Thus, we hope to help students and researchers from one area understand what is going on in the other. One difficulty with communicating across areas is a vast difference in lingo, jargon, and mathematical terminology.

With hands-on experiments, our paper is meant to help create a better understanding of links between the two sides, math and images. It is a delicate balance deciding what to include. In choosing, we had in mind students in operator theory, stressing explanations that are not easy to find in the journal literature.

Our paper results extend what was previously known, and we hope yields new insight into scaling and of representation of color images; especially, we have aimed for better algorithms.

The paper concludes with a set of computer generated images which serve to illustrate our ideas and our algorithms, and also with the resulting compressed images.

1.1. Overview.

How wavelets work in image processing is analogous to how our eyes work. Depending on the location of the observation, one may perceive a forest differently. If the forest was observed from the top of a skyscraper, it will be observed as a blob of green; if it was observed in a moving car, it will be observed as the trees in the forest flashing through, thus the trees are now recognized. Nonetheless, if it is observed by one who actually walks around it, then more details of the trees such as leaves and branches, and perhaps even the monkey on the top of the coconut tree may be observed. Furthermore, pulling out a magnifying glass may even make it possible to observe the texture of the trees and other little details that cannot perceived by bare human eyes. See [**Mac01**], [**Mar82**].

Wavelet Image Processing enables computers to store an image in many scales of resolutions, thus decomposing an image into various levels and types of details and approximation with different-valued resolutions. Hence, making it possible to zoom in to obtain more detail of the trees, leaves and even a monkey on top of the tree. Wavelets allow one to compress the image using less storage space with more details of the image.

The advantage of decomposing images to approximate and detail parts as in 3.3 is that it enables to isolate and manipulate the data with specific properties. With this, it is possible to determine whether to preserve more specific details. For instance, keeping more vertical detail instead of keeping all the horizontal, diagonal and vertical details of an image that has more vertical aspects. This would allow the image to lose a certain amount of horizontal and diagonal details, but would not affect the image in human perception.

As mathematically illustrated in 3.3, an image can be decomposed into approximate, horizontal, vertical and diagonal details. N levels of decomposition is done. After that, quantization is done on the decomposed image where different quantization maybe done on different components thus maximizing the amount of needed details and ignoring ‘not-so-wanted’ details. This is done by thresholding where some coefficient values for pixels in images are ‘thrown out’ or set to zero or some ‘smoothing’ effect is done on the image matrix. This process is used in JPEG2000.

1.2. Motivation. In many papers and books, the topics in wavelets and image processing are discussed in mostly in one extreme, namely in terms of engineering aspects of it or wavelets are discussed in terms operators without being specifically mentioned how it is being used in its application in engineering. In this paper, the author adds onto [**Sko01**], [**Use01**] and [**Vet01**] more insights about mathematical properties such as properties from Operator Theory, Functional Analysis, etc. of wavelets playing a major role in results in wavelet image compression. Our paper aims in establishing if not already established or improve the connection between the mathematical aspects of wavelets and its application in image processing. Also, our paper discuss on how the images are implemented with computer program, and how wavelet decomposition is done on the digital images in terms of computer program, and in terms of mathematics, in the hope that the communication between mathematics and engineering will improve, thus will bring greater benefits to mathematicians and engineers.

2. Wavelet Color Image Compression

2.1. Methods. The whole process of wavelet image compression is performed as follows: An input image is taken by the computer, forward wavelet transform is performed on the digital image, thresholding is done on the digital image, entropy coding is done on the image where necessary, thus the compression of image is done on the computer. Then with the compressed image, reconstruction of wavelet transformed image is done, then inverse wavelet transform is performed on the image, thus image is reconstructed. In some cases, zero-tree algorithm [**Sha93**] is used and it is known to have better compression with zero-tree algorithm but it was not implemented here.

2.1.1. *Forward Wavelet Transform.* Various wavelet transforms are used in this step. Namely, Daubechies wavelets, Coiflets, biorthogonal wavelets, and Symlets. These various transforms are implemented to observe how various mathematical properties such as symmetry, number of vanishing moments and orthogonality differ the result of compressed image. Advantages short support is that it preserves locality. The Daubechies wavelets used are orthogonal, so do Coiflets. Symlets have the property of being close to symmetric. The biorthogonal wavelets are not orthogonal but not having to be orthogonal gives more options to a variety of filters such as symmetric filters thus allowing them to possess the symmetric property.

MATLAB has a subroutine called wavedec2 which performs the decomposition of the image for you up to the given desired level (N) with the given desired wavelet(wname). Since there are three components to deal with, the wavelet transform was applied componentwise. "wavedec" is a two-dimensional wavelet analysis function. [C,S] = wavedec2(X,N,'wname') returns the wavelet decomposition of the matrix X at level N, using the wavelet named in string 'wname'. Outputs are the decomposition vector C and the correspondingbookkeeping matrix S [**MatlabUG**]. Here the image is taken as the matrix X.

2.1.2. *Thresholding.* Since the whole purpose of this project was to compare the performance of each image compression using different wavelets, fixed thresholds were used.

Soft threshold was used in this project in the hope that the drastic differences in gradient in the image would be noted less apparently. The soft and hard thresholdings T_{soft}, T_{hard} are defined as follows:

$$T_{soft}(x) = \begin{cases} 0 & \text{if } |x| \leq \lambda \\ x - \lambda & \text{if } x > \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases} \tag{2.1}$$

$$T_{hard}(x) = \begin{cases} 0 & \text{if } |x| \leq \lambda \\ x & \text{if } |x| > \lambda \end{cases} \tag{2.2}$$

where $\lambda \in \mathbb{R}_+$. As it could be observed by looking at the definitions, the difference between them is related to how the coefficients larger than a threshold value λ in absolute values are handled. In hard thresholding, these coefficient values are left alone. Unlike in hard thresholding, the coefficient values area decreased by λ if positive and increased by λ if negative [**Waln02**].

MATLAB has this subroutine called wthrmngr which computes the global threshold or level dependent thresholds depending on the option and method. The options available are global threshold and level dependent threshold, and the global threshold is used in the program. However, a fixed threshold values were used so as to have the same given condition for every wavelet transform to compare the performances of different conditions. Here, fixed thresholds 10 and 20 were used. For the lossless compression 0 is used as the threshold for an obvious reason.

2.1.3. *Entropy Encoding.* Entropy defined as

$$H(s) = -\sum_{i=1}^{q} P(s_i) log_2(P(s_i)),$$

where s_i are codewords and S is the message. Entropy coding uses codewords with varying lengths, here codewords with short length are used for values that have to be encoded more often, and the longer codewords are assigned for less encoded values. H(S) measures the amount of information in the message, ie. the minimal number of bits needed to encode one word of the message. Unfortunately, the entropy encoding was not implemented on the codes for the color image compression using wavelets. However, Shannon entropy which is defined below was used in the code for the image compression with wavelet packets. See [**Son04**] and [**Bra02**].

The Shannon entropy functional is defined by

$$M(c\{b_j\}) = -\sum_{n=1} M|\langle c, b_j\rangle|^2 log|\langle c, b_j\rangle|^2 \tag{2.3}$$

Also, entropy could be viewed as a quantity that measures the amount of uncertainty in a probability distribution, or equivalently of the amount of information obtained from one sample from the probability space.

2.1.4. *Reconstruction of Wavelet Transformed Image.* At this step, the significance map is taken and with the amplitudes of the non-zero valued wavelet coefficients, the wavelet transformed image is reconstructed.

2.1.5. *Inverse Wavelet Transformation.* The wavelet parameters are converted back into an image almost identical to the original image. How much identical they are will be dependent upon whether the compression was lossy or lossless.

2.2. Wavelets. Compactly supported wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function $f(x)$ as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single prototype wavelet called the mother wavelet $\psi(x)$, by dilations or scaling and translations. Wavelet bases are very good at efficiently representing functions that are smooth except for a small set of discontinuities.

For each $n, k \in \mathbb{Z}$, define $\psi_{n,k}(x)$ by

$$\psi_{n,k}(x) = 2^{n/2}\psi(2^n x - k) \tag{2.4}$$

Constructing the function $\psi(x)$, L^2 on $\mathbb{R}$, such that $\{\psi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is an orthonormal basis on $\mathbb{R}$. As mentioned before $\psi(x)$ is a wavelet and the collection $\{\psi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is a wavelet orthonormal basis on $\mathbb{R}$; this framework for constructing wavelets involves the concept of a multiresolution analysis or MRA.

2.2.1. *Multiresolution Analysis.* Multiresolution analysis is a device for computation of basis coefficients in $L^2(\mathbb{R}) : f = \sum\sum c_{n,k}\psi_{n,k}$. It is defined as follows, see [**Kei04**]: Define

$$V_n = \{f(x)|f(x) = 2^{n/2}g(2^n x), g(x) \in V_0\},$$

where

$$f(x) = \sum_{n\in\mathbb{Z}} \langle f, \phi(\cdot - n)\rangle \phi(x - n).$$

Then a multiresolution analysis on $\mathbb{R}$ is a sequence of subspaces $\{V_n\}_{n\in\mathbb{Z}}$ of functions L^2 on $\mathbb{R}$, satisfying the following properties:

(a) For all $n, k \in \mathbb{Z}, V_n \subseteq V_{n+1}$.
(b) If $f(x)$ is C_c^0 on $\mathbb{R}$, then $f(x) \in \overline{span}\{V_n\}_{n\in\mathbb{Z}}$. That is, given $\epsilon > 0$, there is an $n \in \mathbb{Z}$ and a function $g(x) \in V_n$ such that $\|f - g\|_2 < \epsilon$.
(c) $\bigcap_{n\in\mathbb{Z}} V_n = \{0\}$.
(d) A function $f(x) \in V_0$ if and only if $2^{n/2}f(2^n x) \in V_n$.
(e) There exists a function $\phi(x)$, L^2 on $\mathbb{R}$, called the scaling function such that the collection $\phi(x - n)$ is an orthonormal system of translates and $V_0 = \overline{span}\{\phi(x - n)\}$.

Definition 2.1. Let $\{V_J\}$ be an MRA with scaling function $\phi(x)$ which satisfies (2.14) and scaling filter $h(k)$, where $h(k) = \langle 2^{-1/2}\phi(\frac{x}{2}), \phi(x-k)\rangle$. Then the wavelet filter $g(k)$ is defined by

$$g(k) = (-1)^k\overline{h(1-k)}$$

and the wavelet by

$$\psi(x) = \sum_{k\in\mathbb{Z}} g(k)\sqrt{2}\phi(2x - k).$$

See [**Kei04**].

Then $\{\psi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is a wavelet orthonormal basis on $\mathbb{R}$.

Definition 2.2. The orthogonal projection of an arbitrary function $f \in L^2$ onto V_n is given by

$$P_n f = \sum_{k\in\mathbb{Z}} \langle f, \phi_{n,k}\rangle \phi_{n,k}$$

[**Kei04**].

As k varies, the basis functions $\phi_{n,k}$ are shifted in steps of 2^{-n}, so P_nf cannot represent any detail on a scale smaller than that. We say that the functions in V_n have the resolution 2^{-n} or scale 2^{-n}. Here, P_nf is called an approximation to f at resolution 2^{-n}. For a given function f, an MRA provides a sequence of approximations P_nf of increasing accuracy [**Kei04**]. We include the following known proof for the benefit of the readers.

Theorem 2.3. [**Waln02**] *For all $f(x) \in C^0_c(\mathbb{R})$, $\lim_{n\to\infty} \|P_nf - f\|_2 = 0$*

Proof. Let $\epsilon > 0$. Then there exists $N \in \mathbb{Z}$ and $g(x) \in V_j$ such that $\|f-g\|_2 < \epsilon/2$. By 2.1, $g(x) \in V_n$ and $P_ng(x) = g(x)$ for all $n \leq N$. Thus,

$$\begin{aligned}\|f - P_nf\|_2 &= \|f - g + P_ng - P_nf\| \\ &\leq \|f-g\|_2 + \|P_n(f-g)\|_2 \\ &\leq 2\|f-g\|_2 \\ &< \epsilon\end{aligned}$$

where Minkowski's and Bessel's inequalities are applied. Since this holds for all $n \geq N$ the proof is complete. □

Definition 2.4. The difference between the approximations at resolution 2^{-n} and 2^{-n-1} is called the fine detail at resolution 2^{-n} which is as follows:

$$Q_nf(x) = P_{n+1}f(x) - P_nf(x)$$

or

$$Q_nf = \sum_{k\in\mathbb{Z}} \langle f, \psi_{n,k}\rangle \psi_{n,k}$$

Q_n is also an orthogonal projection and its range W_n is orthorgonal to V_n where the following holds:

$$V_n = \{f | P_nf = f\} \tag{2.5}$$

$$W_n = \{f | Q_nf = f\} \tag{2.6}$$

$$V_n \oplus W_n = V_{n+1} \tag{2.7}$$

$$\psi \in V_{-1} \ominus V_0 = \{f | f \in V_{-1}, f \perp V_0\} = W_0 \tag{2.8}$$

Theorem 2.5. [**Dau92**], [**Waln02**] *There are choices of the numbers h and g in 2.1 such that $\{\psi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is a wavelet orthonormal basis on $\mathbb{R}$.*

Proof. We must show othornomality and completeness. As for completenes, we have

$$\bigcap_{n\in\mathbb{Z}} V_n = 0 \tag{2.9}$$

and

$$\overline{\bigcup_n V_n} = L^2(\mathbb{R}). \tag{2.10}$$

Then we have $\{\psi_{n,k} | k \in \mathbb{Z}\} = W_n = V_n \ominus V_{n-1}$. Hence $\{\psi_{n,k}\}_{n,k\in\mathbb{Z}}$ is complete if and only if $\sum_n W_n = L^2(\mathbb{R})$ holds, and this is true if and only if (2.9) and (2.10) holds. Since we have those conditions for $\{\psi_{n,k}\}_{n,k\in\mathbb{Z}}$, it is complete.

Now, as for the orthonomality,

$$\begin{aligned}\langle \psi_{n,k}, \psi_{n,l}\rangle &= \langle 2^{n/2}\psi(2^n x - k), 2^{n/2}\psi(2^n x - l)\rangle \\ &= \langle \psi_{0,k}, \psi_{0,l}\rangle = \delta(k-l).\end{aligned}$$

To prove orthonomality between scales, let $n, n' \in \mathbb{Z}$ with $n' < n$, and let $k, k' \in \mathbb{Z}$ be arbitrary. Since $\psi(x) \in V_1$, $\psi_{0,k'}(x) \in V_1$ also. Then we have $\psi_{n',k'} \in V_{n'+1}$. Since $\langle \psi_{0,k}, \psi_{0,l}\rangle = 0$ for all $k, l \in \mathbb{Z}$, it follows that $\langle \psi_{n,k}, \phi_{n,l}\rangle = 0$, for all $n, k, l \in \mathbb{Z}$. Given $f(x) \in V_n$ we know that $f(x) = \sum_k \langle f, \phi_{n,k}\rangle \phi_{n,k}(x)$. Hence for $f(x) \in V_n$,

$$\begin{aligned}\langle \psi_{n,l}, f\rangle &= \langle \psi_{n,l}, \sum_k \langle f, \phi_{n,k}\rangle \phi_{n,k}\rangle \\ &= \sum_k \overline{\langle f, \phi_{n,k}\rangle} \langle \psi_{n,l}, \phi_{n,k}\rangle = 0\end{aligned}$$

Since, $n' < n$, $V_{n'+1} \subseteq V_n$ and since $\psi_{n',k'} \in V_{n'+1}$, $\psi_{n',k'} \in V_n$ also. Hence $\langle \psi_{n,k}, \psi_{n',k'}\rangle = 0$. Therefore $\{\psi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is a wavelet orthonormal basis on $\mathbb{R}$. See [**Waln02**]. □

2.2.2. *Symmetry.* Symmetric filters are preferred for they are most valuable for minimizing the edge effects in the wavelet representation of discrete wavelet transform(DWT) of a function; large coefficients resulting from false edges due to periodization can be avoided.

Since orthogonal filters in exception to Haar-filter cannot be symmetric, biorthogonal filters are almost always selected for image compression application [**Waln02**].

2.2.3. *Vanishing Moments.* Vanishing Moments are defined as follows: From the definition of multiresolution analysis(MRA), any wavelet $\psi(x)$ that comes from an MRA must satisfy

$$\int_{\mathbb{R}} \psi(x)dx = 0 \tag{2.11}$$

The integral (2.11) is referred to as the zeroth moment of $\psi(x)$, so that if (2.11) holds, we say that $\psi(x)$ has its zeroth moment vanishing. The integral $\int_{\mathbb{R}} x^k \psi(x)dx$ is referred to as the k^{th} moment of $\psi(x)$ and if $\int_{\mathbb{R}} x^k \psi(x)dx = 0$, we say that $\psi(x)$ has its k^{th} moment vanishing [**Waln02**].

We may encounter a situation where having different number of vanishing moments on the analysis filters than on the reconstruction filters. As a matter of fact, it is possible to have different number of vanishing moments on the analysis filters than on the reconstruction filters. Vanishing moments on the analysis filters are desired for small coefficients in the transform as a result, whereas vanishing moments on the reconstruction filter results in fewer blocking artifacts in the compressed image thus is desired. Thus having sufficient vanishing moments which maybe different in numbers on each filters are advantageous.

2.2.4. *Size of the filters.* Long analysis filters results in greater computation time for the wavelet or wavelet packet transform. Long reconstruction filters can create unpleasant artifacts in the compressed image for the following reason. The reconstructed image is made up of the superposition of only a few scaled and shifted reconstruction filters. So features of the reconstruction filters such as oscillations or lack of smoothness, can be obvious noted in the reconstructed image. Smoothness can be guaranteed by requiring a large number of vanishing moments in the reconstruction filter. However, such filters tend to be oscillatory [**Waln02**]. Also, see [**Vet01**].

2.3. Various Wavelets. For later use in computations, we recall here the specific wavelets needed:

2.3.1. *Haar Wavelet.* Haar wavelet is the only known wavelet that is compactly supported, orthogonal and symmetric [**WaEx05**].

The Haar system is defined as follows in [**Waln02**]: Define

$$I_{n,k} = [2^{-n}k, 2^{-n}(k+1))].$$

Let $p(x) = \chi_{[0,1)}(x)$, and for each $n, k \in \mathbb{Z}$, define $p_{n,k}(x) = 2^{n/2}p(2^n x - k)$. The collection $\{\phi_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is referred to as the system of Haar scaling functions.

(a) For each $n, k \in \mathbb{Z}$, $p_{n,k}(x) = 2^{n/2}\chi_{I_{n,k}}(x)$, so that $p_{n,k}(x)$ is supported on the interval $I_{n,k}$ and does not vanish on that interval. There for we refer to the scaling function $p_{n,k}(x)$ as being associated with the interval $I_{n,k}$.

(b) For each $n, k \in \mathbb{Z}$, $\int_{\mathbb{R}} p_{n,k}(x)dx = \int_{I_{n,k}} p_{n,k}(x)dx = 2^{n/2}$ and

$$\int_{\mathbb{R}} |p_{n,k}(x)|^2 dx = \int_{I_{n,k}} |p_{n,k}(x)|^2 dx = 1.$$

Definition 2.6. Let $h(x) = \chi_{[0,1/2)}(x) - \chi_{[1/2,1)}(x)$, and for each $n, k \in \mathbb{Z}$ define $h_{n,k}(x) = 2^{n/2}h(2^n x - k)$. The collection $\{h_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is referred to as the Haar system on $\mathbb{R}$. For each $n \in \mathbb{Z}$, the collection $\{h_{n,k}(x)\}_{n,k\in\mathbb{Z}}$ is referred to as the system of scale j Haar functions.

Definition 2.7. Given $J, N \in \mathbb{N}$ with $J < N$ and a finite sequence $c_0 = \{c_0\}_{k=0}^{2^N-1}$, the discrete Haar transform of c_0 is defined by $\{d_j(k) | 1 \le j \le J; 0 \le k \le 2^{N-j} - 1\} \cup \{c_J(k) \| 0 \le k \le 2^{N-j} - 1\}$ where

$$c_j(k) = \frac{1}{\sqrt{2}}c_{j-1}(2k) + \frac{1}{\sqrt{2}}c_{j-1}(2k+1) d_j(k) = \frac{1}{\sqrt{2}}c_{j-1}(2k) + \frac{1}{\sqrt{2}}c_{j-1}(2k+1) \tag{2.12}$$

$$c_{j-1}(2k) = \frac{1}{\sqrt{2}}c_j(k) + \frac{1}{\sqrt{2}}d_j(k) c_{j-1}(2k+1) = \frac{1}{\sqrt{2}}c_j(k) - \frac{1}{\sqrt{2}}d_j(k) \tag{2.13}$$

Haar wavelets are basically same as Daubechies wavelet db1 (in MATLAB) or Daub4. Haar wavelets are example of compactly supported wavelets. The compact support of the Haar wavelets enables the Haar decomposition to have a good time localization. Specifically, this means that the Haar coefficients are effective for locating jump discontinuities and also for the efficient representation of signals with small support. However, the fact that they have jump discontinuities(sect5.4.3), in particular in the poorly decaying Haar coefficients of smooth functions and in the blockiness of images reconstructed from subsets of the Haar coefficients [**Waln02**].

2.3.2. *Daubechies Wavelets Constructions.* In order to construct compactly-supported, orthogonal wavelets, we first look at the dilation equation

$$\phi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} h(k)\phi(2x - k) \tag{2.14}$$

and the wavelet equation

$$\psi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} g(k)\phi(2x-k) \tag{2.15}$$

Notice from these two equations that the compactness of the support of ϕ and ψ can be achieved if we choose the number of nonvanishing coefficients $\{h(k)\}$, that is, the filter length, to be finite. This implies that $m_0(\omega) = \sum_j h(j)e^{-in\omega}/\sqrt{2}$ is a trigonometric polynomial.

One can see that once we know ϕ at the integers, the values of ϕ at the dyadic points $k/2^n$ then can be obtained recursively using the scaling equation. Once we find ϕ we can use (2.15) to generate ψ.

So Daubechies' approach to finding ϕ and ψ is to first determine the finite number of filter coefficients h(j) such that orthogonality and smoothness or moment conditions are guaranteed. To find h(j) we start from the Fourier domain where the orthogonality condition for the scaling function ϕ is.

$$|m_0(\omega)|^2 + |m_0(\omega+\pi)|^2 = 1. \tag{2.16}$$

The condition that the first N moments of vanish is

$$\int_{\mathbb{R}} x^k\psi(x)dx = 0 \text{for } k = 0, 1, ..., N-1. \tag{2.17}$$

To satisfy the moment condition (2.17), $m_0(\omega)$ has to assume the following form:

$$m_0(\omega) \propto (\frac{1+e^{i\omega}}{2})^N \tag{2.18}$$

Now, define

$$M_0(\omega) = |m_0(\omega)|^2,$$

where $M_0(\omega)$ is a polynomial in $cos(\omega)$, the moment-vanishing condition implies

$$M_0(\omega) = (cos^2\omega/2)^N L(\omega), \tag{2.19}$$

where $L(\omega)$ is a polynomial in $cos(\omega)$, and the orthogonality condition gives

$$M_0(\omega) + M_0(\omega+\pi) = 1, \tag{2.20}$$

By the half angle identity, we can write $L(\omega) = P(sin^2\omega/2)$. Now, it leaves us to find the form of the polynomial P, "take its square root" to get $m_0(\omega)$, and identify the coefficients $\{h(j)\}$. Let $y = sin^2\omega/2$, using (2.19) and (2.20), we see that P satisfies

$$(1-y)^N P(y) + y^N P(1-y) = 1, \tag{2.21}$$

or

$$P(y) = (1-y)^{-N}(1 - y^N P(1-y)). \tag{2.22}$$

It turns out that the lowest-degree polynomial that satisfies (2.21) is $N-1$. So we can find the form of the polynomial P with degree $N-1$ explicitly from (2.22) by expanding $(1-y)^{-N}$ in a Taylor series and retaining terms up to order $N-1$:

$$P(y) = (1-y)^{-N}(1 - y^N P(1-y)) = \sum_{k=0}^{N-1} \binom{N+k-1}{k} y^k. \tag{2.23}$$

Notice that $P(y) \geq 0$ for $y \in [0,1]$.

Summarizing what we have done, we have established

$$|m_0(\omega)|^2 = M_0(\omega) = (cos^2\omega/2)^N P(sin^2\omega/2), \tag{2.24}$$

where P is given by (2.23), such that $m_0(\omega)$ satisfies the required moment-vanishing condition and the orthogonality condition. Now it leaves us to "take the square root" of P to obtain $m_0(\omega)$ and it will be done by spectral factorization.

We write,

$$A(\omega) = P(\frac{1 - cos(\omega)}{2}) = \alpha \prod_{j=0}^{N-1} (cos(\omega) - c_j)$$

where we have regarded P as a polynomial of degree $N - 1$, and expressed it in terms of its roots $\{c_j\}$ (α is a constant). Since the polynomial has real coefficients, the roots c_j either are real or occur in complex conjugate pairs.

On the other hand, we can also write the original polynomial P in terms of $z = e^{i\omega}$,

$$A(\omega) = \alpha \prod_{j=0}^{N-1} (\frac{z + z^{-1}}{2} - c_j) = P(\frac{1 - (z + z^{-1}/2)}{2}) = P(z). \tag{2.25}$$

The zeros of $p(z)$ appear in quadruplets $\{z_j, \bar{z_j}, z_j^{-1}, \bar{z_j}^{-1}\}$ if $z_j \in \mathbb{C}$, and in doublets $\{r_j, r_j^{-1}\}$ if $z_j = r_j \in \mathbb{R}$. So we can write

$$p(z) = \alpha z^{-(N-1)} \prod_{j=0}^{N-1} (\frac{z^2}{2} - c_j z + \frac{1}{2})$$

$$= \alpha' z^{-(N-1)} \prod_j (z - z_j)(z - \bar{z_j})(z - z_j^{-1})(z - \bar{z_j}^{-1}) \prod_k (z - z_k)^2((z - \bar{z_k})^2 \prod_l (z - r_l)(z - r_l^{-1}).$$

Earlier, we have separated the case $z_j = e^{i\alpha j}$, where $z_j = \bar{z_j}^{-1}$ and the quadruplet reduces to a doublet of degeneracy 2 [**WaEx05**].

The Daubechies wavelets are orthogonal wavelets which is energy or norm preserving. There are a number of Daubechies wavelets, DaubJ, where $J = 4, 6, , 20$. The easiest way to understand this transform is just to treat them as simple generations of the Daub4 transform with the scaling and translation factors. The most apparent difference between each of them is the length of the supports of their scaling signals and wavelets. Daub4 wavelet is the same as the Haar wavelet. Daub4 wavelet preserves the energy due to its orthogonality and the proof of this could be found in p.39-40 of [**Wal99**]. Daub4 transform is suitable for identifying features of the signal that are related to turning points in its graph [**Wal99**]. Now, one might wonder why we have so many different DaubJs and their advantages, and disadvantages. Daub6 often produces smaller size fluctuation values than those produced by Daub4 transform. The types of signals for which this occurs are the ones that are obtained from the sample of analog signals that are at least three times continuously differentiable. These kinds of signals are approximated better, over a large proportion of their values by quadratic approximations. The curve graphs of quadratic functions enable then to provide superior approximations to the parts of the signal that are near to the turning points in its graph. So for signal compression Daub6 transform generally does a better job. However, the fact that Daub4 being better in approximating signals better approximated by linear approximation [**Wal99**].

If $H = H'$ and $G = G'$ in a biorthogonal set [**Dau92**] of quadrature filters, then the pair H, G is called an orthogonal quadrature filter pair which is a pair of operators and is defined as follows:

$$Hu(i) = \sum_{j=-\infty}^{\infty} h(2i-j)u(j), i \in \mathbb{Z} \tag{2.26}$$

$$Gu(i) = \sum_{j=-\infty}^{\infty} g(2i-j)u(j), i \in \mathbb{Z} \tag{2.27}$$

In addition, following conditions hold:

- Self-duality: $H'H^* = G'G^* = I$
- Independence: $GH^* = HG^* = 0$
- Exact reconstruction: $H^*H + G^*G = I$
- Normalization: $H\mathbf{1} = \sqrt{2}\mathbf{1}$

H is the low-pass filter and G is the high-pass filter.

The first two conditions may be expressed in terms of the filter sequences h, g which respectively define H, G:

$$\sum_k h(k)\bar{h}(k+2n) = \delta(n) = \sum_k g(k)\bar{g}(k+2n) \tag{2.28}$$

$$\sum_k g(k)\bar{h}(k+2n) = 0 = \sum_k h(k)\bar{g}(k+2n) \tag{2.29}$$

See [**Wic94**].

2.3.3. *Coiflets.* Coiflets are designed so as to maintain a close match between the trend values and the original signal values. All of the coiflets, CoifI, $I = 6, 12, 18, 24, 30$ are defined in a similar way as Daubechies wavelets but they have some different properties. Coif6 transform produces a much closer match between trend subsignals and the original signal values than the match that any of the DaubJ transforms can produce. This means that the . CoifI wavelets have nearly symmetric graphs [**Wal99**].

Coifman wavelet systems are similar to Daubechies wavelet systems (in rank 2) in that they have a maximal number of vanishing moments, but the vanishing of moments are equally distributed between the scaling function and the wavelet function. In contrast to the case for Daubechies wavelets, there is no formula for Coiflets of arbitrary genus, and there is no formal proof of their existence for arbitrary genus at this time. There are numerical solutions using Newton's method which work well until round-off error gives problems, up to about genus 20 (round-off error is also a problem in calculating the Daubechies scaling vector numerically beyond this same range with spectral factorization, even though the formulas are valid and give an existence theorem for every genus [**Res98**].

If we used Daubechies wavelets in the same way, one cannot get the same approximation results, except to low order.

It is very advantageous to have a high number of vanishing moments for ψ; it leads to high compressibility because the fine scale wavelets coefficients of a function would be essentially zero where the function is smooth. Since $\int_{\mathbb{R}} \phi(x) = 1$, the same thing can never happen for the $\langle f, \phi_{n,k} \rangle$. Nevertheless, if $\int_{\mathbb{R}} x^l \phi(x)dx = 0$ for $l = 1, ..., L$, then we can apply the same Taylor expansion argument and conclude that for N large, $\langle f, \phi_{-N,k}\rangle \approx 2^{N/2} f(2^{-N}k)$, with an error that is negligibly small where f is smooth. This means that we have a remarkably simple quadrature rule to go from the sample of f to its fine scale foefficients $\langle f, \phi_{-N,k}\rangle$. For this reason, R. Coifman suggested that it might be worthwhile to construct orthonormal wavelet bases with vanishing moments not only for ψ, but also for ϕ. See [**Dau92**].

2.3.4. *Biorthogonal Wavelets.* The biorthogonal wavelets have bases that are defined in a way that has weaker definition of the bases of orthogonal wavelet bases. Though the orthogonal wavelet's filter has self-duality only, the biorthogonal wavelet's filter has duality. Since the orthogonality of the filter makes the wavelet energy preserving as proven in [**Wal99**], the biorthogonal wavelets are not energy preserving.

Current compression systems use biorthogonal wavelet instead of orthogonal wavelets despite the fact that it is not energy preserving. The fact that biorthogonal wavelets are not energy preserving is not a big problem since there are linear phase biorthogonal filter coefficients which are "close" to being orthogonal [**Use01**]. The main advantage of the biorthogonal wavelet transform is that it permits the use of a much broader class of filters, and this class includes the symmetric filters. The biorthogonal wavelet transform is advantageous because it can use linear phase filters which gives symmetric outputs when presented with symmetric input. This transform is called the symmetric wavelet transform and it solves the problems of coefficient expansion and border discontinuities. See [**Use01**].

A quadruplet H, H', G, G' of convolution operators or filters is said to form a set of biorthogonal quadrature filters, if the filters satisfy the following conditions:

- Duality: $H'H^* = G'G^* = I = HH'^* = GG'^*$
- Independence: $G'H^* = H'G^* = 0 = GH'^* = HG'^*$
- Exact reconstruction: $H^*H' + G^*G' = I = H'^*H + G'^*G$
- Normalization: $H\mathbf{1} = H'\mathbf{1} = \sqrt{2}\mathbf{1}$ and $G\mathbf{1} = G'\mathbf{1} = 0$

H and H' are the low-pass filter and G and G' are the high-pass filter.

The first two conditions may be expressed in terms of the filter sequences h, h', g, g' which respectively define H, H', G, G':

$$\sum_k h'(k)\bar{h}(k+2n) = \delta(n) = \sum_k g'(k)\bar{g}(k+2n) \tag{2.30}$$

$$\sum_k g'(k)\bar{h}(k+2n) = 0 = \sum_k h'(k)\bar{g}(k+2n) \tag{2.31}$$

See [**Wic94**].

Notice the difference in (2.3.2) and (2.3.4) that self-duality no longer holds in 2.3.4 and the conditions are weakened.

$$\sum_k h(k) = \sqrt{2}; \tag{2.32}$$

$$\sum_k g(2k) = -\sum_k g(2k+1); \tag{2.33}$$

$$\sum_k h'(k) = \sqrt{2}; \tag{2.34}$$

$$\sum_k g'(2k) = -\sum_k g'(2k+1). \tag{2.35}$$

See [**Wic94**]. Having four operators gives plenty of freedom to construct filters with special properties, such as symmetry.

2.3.5. *Symlets.* The family of wavelets symlets are short of "symmetrical wavelets". They are not perfectly symmetrical, but they are designed in such a way that they have the least asymmetry and highest number of vanishing moments for a given compact support [**Gon02**]. Symlets are another family of Daubechies wavelets thus are constructed in the same way as Daubechies wavelets.

THEOREM 2.8. *The wavelet algorithms listed in this chapter can be realized as images; ie. the wavelets as matrices can be applied to image matrices for wavelet decomposition.*

The proof is implemented in the next section.

3. Digital Image Representation and Mathematics behind It

In this section we will explore the digital image representation and Mathematics behind it. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This enables formulating solutions to many technical computing problems, especially those involving matrix representations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fotran.

The name MATLAB stands for matrix laboratory. In university environments, MATLAB is the standard computational tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the computational tool of choice for research, development, and analysis. MATLAB is complemented by a family of application-specific solutions called toolboxes; here, Wavelet Toolbox is used [**Gon04**].

3.1. Digital Image Representation. An image is defined as a two-dimensional function ie. a matrix, $f(x, y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at the point. Color images are formed by combining the individual two-dimensional images. For example, in the RGB color system, a color images consists of three namely, red, green and blue individual component images. Thus many of the techniques developed for monochrome images can be extended to color images by processing the three component images individually. When x, y and the amplitude values of f are all finite, discrete quantities, the image is called a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels and pixels. Since pixel is the most widely used term, the elements will be denoted as pixels from now on.

An image maybe continuous with respect to the x- and y-coordinates, and also in amplitude. Digitizing the coordinates as well as the amplitude will take into effect the conversion of such an image to digital form. Here, the digitization of the coordinate values are called sampling; digitizing the amplitude values is called quantization. A digital image is composed of a finite number of elements, each of which has a particular location and value. The field of digital image processing refers to processing digital images by means of a digital computer. See [**Gon04**].

3.1.1. *Coordinate Convention.* Assume that an image $f(x, y)$ is sampled so that the resulting image has M rows and N columns. Then the image is of size $M \times N$. The values of the coordinates (x, y) are discrete quantities. Integer values are used for these discrete coordinates. In many image processing books, the image origin is set to be at $(x, y) = (0, 0)$. The next coordinate values along the first row of the image are $(x, y) = (0, 1)$. Note that the notation $(0, 1)$ is used to signify the second sample along the first row. These are not necessarily the actual values of physical coordinates when the image was sampled. Note that x ranges from 0 to $M-1$, and y from 0 to $N-1$, where x and y are integers. However, in the Wavelet Toolbox the notation (r, c) is used where r indicates rows and c indicates the columns. It could be noted that the order of coordinates is the same as the order discussed previously. Now, the major difference is that the origin of the coordinate system is at $(r, c) = (1, 1)$; hence r ranges from 1 to M, and c from 1 to N for r and c integers. The coordinates are referred to as pixel coordinates. See [**Gon04**].

3.1.2. *Images as Matrices.* The coordinate system discussed in preceding section leads to the following representation for the digitized image function:

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

The right side of the equation is a representation of digital image. Each element of this array(matrix) is called the pixel.

Now, in MATLAB, the digital image is represented as the following matrix:

$$\mathbf{f} = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & \vdots & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix} \tag{3.1}$$

where M = the number of rows and N = the number of columns Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. See [**Gon04**].

3.1.3. *Color Image Representation in MATLAB.* An RGB color image is an $M \times N \times 3$ array or matrix of color pixels, where each color pixel consists of a triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location. An RGB image may be viewed as a "stack" of three gray-scale images, that when fed into the red, green, and blue inputs of a color monitor, produce a color image on the screen. So from the "stack" of three images forming that RGB color image, each image is referred to as the red, green, and blue component images by convention. Now, the data class of the component images determine their range of values. If an RGB color image is of class double, meaning that all the pixel values are of type double, the range of values is $[0, 1]$. Likewise, the range of values is $[0, 255]$ or $[0, 65535]$ for RGB images of class uint8 or uint16, respectively. The number of bits used to represent the pixel values of the component images determines the bit depth of an RGB color image. See [**Gon04**].

The RGB color space is shown graphically as an RGB color cube. The vertices of the cude are the primary (red, green, and blue) and secondary (cyan, magenta, and yellow) colors of light. See [**Gon04**].

3.1.4. *Indexed Images.* An indexed image has two components: a data matrix of integers, X, and a colormap matrix, map. Matrix map is an $m \times 3$ array of class double containing floating-point values in the range $[0, 1]$. The length, m, of the map is equal to the number of colors it defines. Each row of map specifies the red, green, and blue components of a single color. An indexed image uses "direct mapping" of pixels intensity values of colormap values. The color of each pixel is determined by using the corresponding value of integer matrix X as a pointer into map. If X is of class double then all of its components with value 2 point to the second row, and so on. If X is of class unit 8 or uint16, then all components with value 0 point to the first row in map, all components with value 1 to point to the second row, and so on [**Gon04**].

3.1.5. *The Basics of Color Image Processing.* Color image processing techniques deals with how the color images are handled for a variety of image-processing tasks. For the purposes of the following discussion we subdivide color image processing into three principal areas: (1) color transformations (also called color mappings); (2) spatial processing of individual color planes; and (3) color vector processing. The first category deals with processing the pixels of each color plane based strictly on their values and not on their spatial coordinates. This category is analogous to the intensity transformations. The second category deals with spatial (neighborhood) filtering for individual color planes and is analogous to spatial filtering. The third category deals with techniques base on processing all components of a color image simultaneously. Since full-color images have at least three components, color pixels are indeed vectors. For example, in the RGB color images, the RGB system color point can be interpreted as a vector extending from the origin to that point in the RGB coordinate system.

Let c represent an arbitrary vector in RGB color space:

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

This equation indicates that the components of c are simply the RGB components of a color image at a point. Since the color components are a function of coordinates (x, y) by using the notation

$$\mathbf{c}(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

For an image of size $M \times N$, there are MN such vectors, $\mathbf{c(x,y)}$, for $x = 0, 1, , M-1$ and $y = 0, 1, , N-1$ [**Gon04**].

In order for independent color component and vector-based processing to be equivalent, two conditions have to be satisfied: (i) the process has to be applicable to both vectors and scalars. (ii) the operation on each component of a vector must be independent of the other components. The averaging would be accomplished by summing the gray levels of all the pixels in the neighborhood. Or the averaging could be done by summing all the vectors in the neighborhood and dividing each component of the average vector is the sum of the pixels in the image corresponding to that component, which is the same as the result that would be obtained if the averaging were done on the neighborhood of each component image individually, and then the color vector were formed [**Gon04**].

3.2. Reading Images. In MATLAB, images are read into the MATLAB environment using function called imread. The syntax is as follows: imread(filename) Here, filename is a string containing the complete name of the image file including any applicable extension. For example, the command line >> f = imread (lena.jpg); reads the JPEG image lena into image array or image matrix f.

Since there are three color components in the image, namely red, green and blue components, the image is broken down into the three distinct color matrices $\mathbf{f}_R$, $\mathbf{f}_G$ and $\mathbf{f}_B$ in the form 3.1. See [**Gon04**].

3.3. Wavelet Decomposition of an Image.

3.3.1. *Color Conversion.* In the process of image compression, applying the compression to the RGB components of the image would result in undesirable color changes. Thus, the image is transformed into its intensity, hue and color saturation components. The color transformation used in JPEG 2000 standard [**Sko01**] has been adopted. For the lossy compression, equations (3.2) and (3.3) were used in the program.

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0 & 1.402 \\ 1.0 & 0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} \tag{3.3}$$

Also, see [**Bha97**], [**Rao96**]. For more details about subjective quality evaluation of the different color spaces, plesae see [**JPG00**], [**Nad99**]. In YC_bC_r color space, Y is the single component that represents luminance. C_b and C_r store the color information where C_b stands for difference between the blue component and a reference value, and C_r is the difference between the red component and a reference value [**Gon04**]. In the case of lossless compression equations (3.4) and (3.5) were used.

$$\begin{bmatrix} Y_r \\ V_r \\ U_r \end{bmatrix} \begin{bmatrix} \lfloor \frac{R+2G+B}{4} \rfloor \\ R - G \\ B - G \end{bmatrix} \tag{3.4}$$

$$\begin{bmatrix} G \\ R \\ Br \end{bmatrix} = \begin{bmatrix} Y_r - \lfloor \frac{U_r+V_r}{4} \rfloor \\ V_r + G \\ U_r + G \end{bmatrix} \tag{3.5}$$

Here, Y is the luminance and U and V are chrominance values (light intensity and color intensity), the subscript r stands for reversible. The advantage of this color system is that the human perception for the Y component is substantially more sensitive than for fluctuations in the U or V components. This can practically be used to transform U and V components are transfered less. That is, of these components, it reduces the data set of these two components to 1/4 of the original amount to be worth transferring [**Han00**], [**Sko01**].

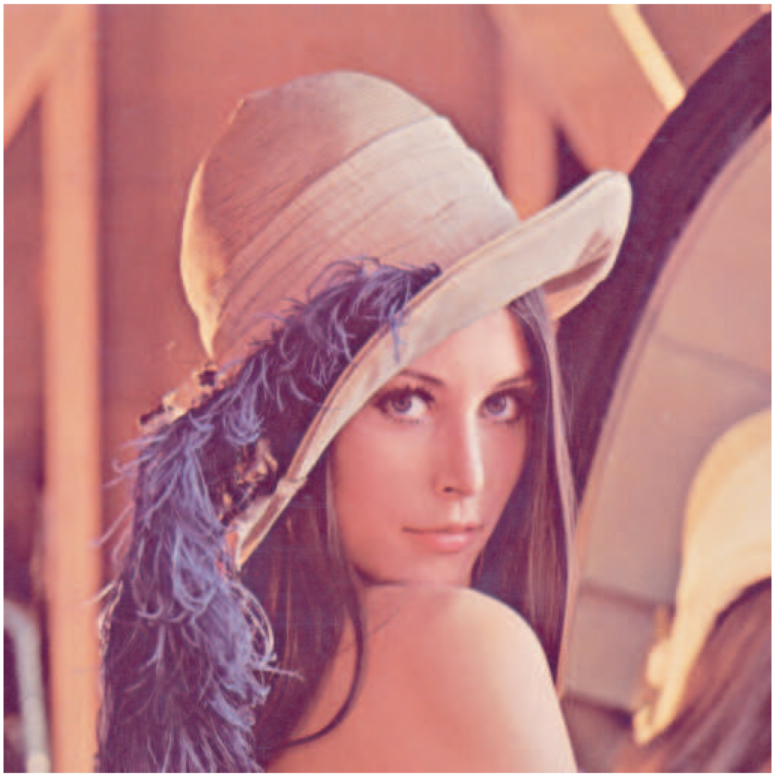

Figure 1. Original Lena Image.

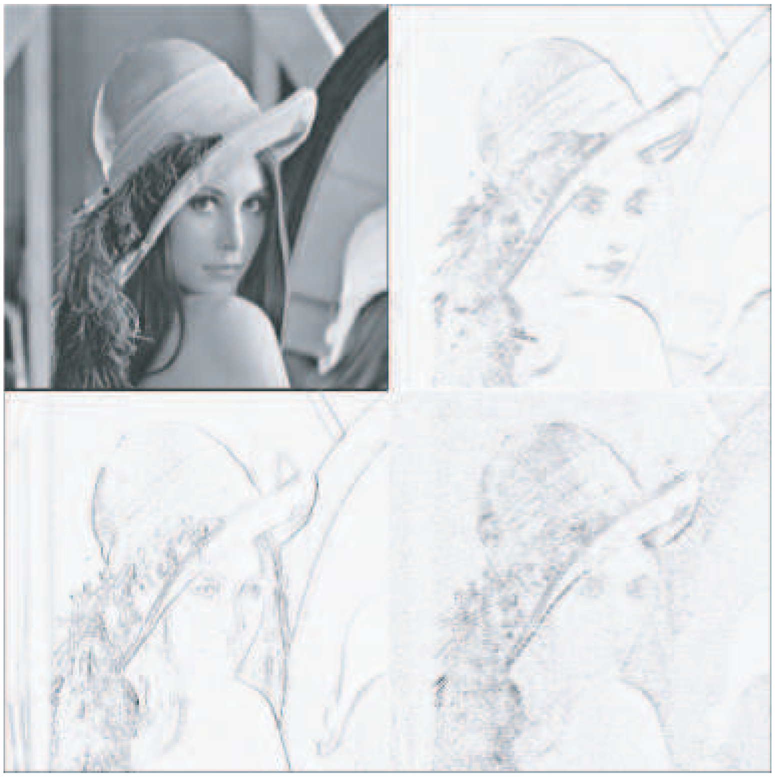

Figure 2. Wavelet Decomposition of an Image Component. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

A 1-level wavelet transform of an $N \times M$ image can be represented as

$$\mathbf{f} \mapsto \begin{pmatrix} \mathbf{a}^1 & | & \mathbf{h}^1 \\ -- & & -- \\ \mathbf{v}^1 & | & \mathbf{d}^1 \end{pmatrix} \tag{3.6}$$

$$\begin{array}{l} \mathbf{a}^1 = V_m^1 \otimes V_n^1 = \phi(x,y) = \phi(x)\phi(y) = \sum_i \sum_j h_i h_j \phi(2x-i)\phi(2y-j) \\ \mathbf{h}^1 = V_m^1 \otimes W_n^1 = \psi^H(x,y) = \psi(x)\phi(y) = \sum_i \sum_j g_i h_j \psi(2x-i)\phi(2y-j) \\ \mathbf{v}^1 = W_m^1 \otimes V_n^1 = \psi^V(x,y) = \phi(x)\psi(y) = \sum_i \sum_j h_i g_j \phi(2x-i)\psi(2y-j) \\ \mathbf{d}^1 = W_m^1 \otimes W_n^1 = \psi^D(x,y) = \psi(x)\psi(y) = \sum_i \sum_j g_i g_j \psi(2x-i)\psi(2y-j) \end{array} \tag{3.7}$$

where the subimages $\mathbf{h}^1, \mathbf{d}^1, \mathbf{a}^1$ and $\mathbf{v}^1$ each have the dimension of $N/2$ by $M/2$.

Here, $\mathbf{a}^1$ denotes the first averaged image, which consists of average intensity values of the original image. $\mathbf{h}^1$ denotes the first detail image of horizontal components, which consists of intensity difference along the vertical axis of the original image. $\mathbf{v}^1$ denotes the first detail image of vertical components, which consists of intensity difference along the horizontal axis of the original image. $\mathbf{d}^1$ denotes the first detail image of diagonal components, which consists of intensity difference along the diagonal axis of the original image. The original image is reconstructed from the decomposed image by taking the sum of the averaged image and the detail images and scaling by a scaling factor. See [**Wal99**].

Here wavelet decomposition of images was performed the number of times the image can be divided by 2 ie. (floor(log2(min(size of Image)))) times. The averaged image of the previous level is decomposed into the four subimages in each level of wavelet image decomposition.

Applying further wavelet decomposition on image in Figure 2 would result in images Figure 3 and Figure 4. Note that the image on the top left most corner get blurrier as it gets "averaged" out and also note the horizontal, vertical and diagonal components of the image. A better example where the horizontal, vertical and diagonal components are more explicitly shown in images Figure 6 and Figure 7. Notice that the horizontal, vertical and diagonal components in the rectangular duster in the picture.

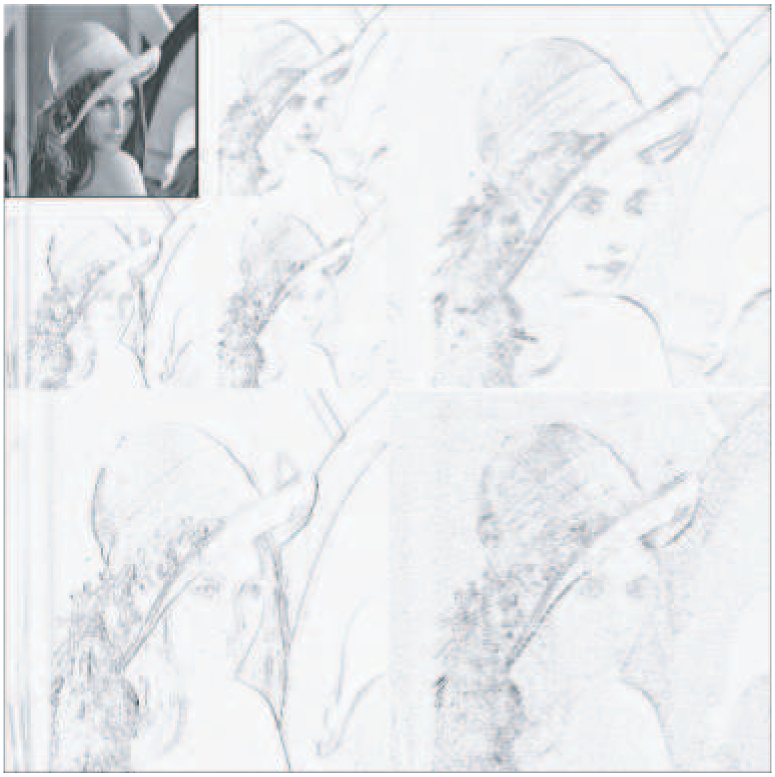

FIGURE 3. Wavelet Decomposition of an Image Component - 2nd Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

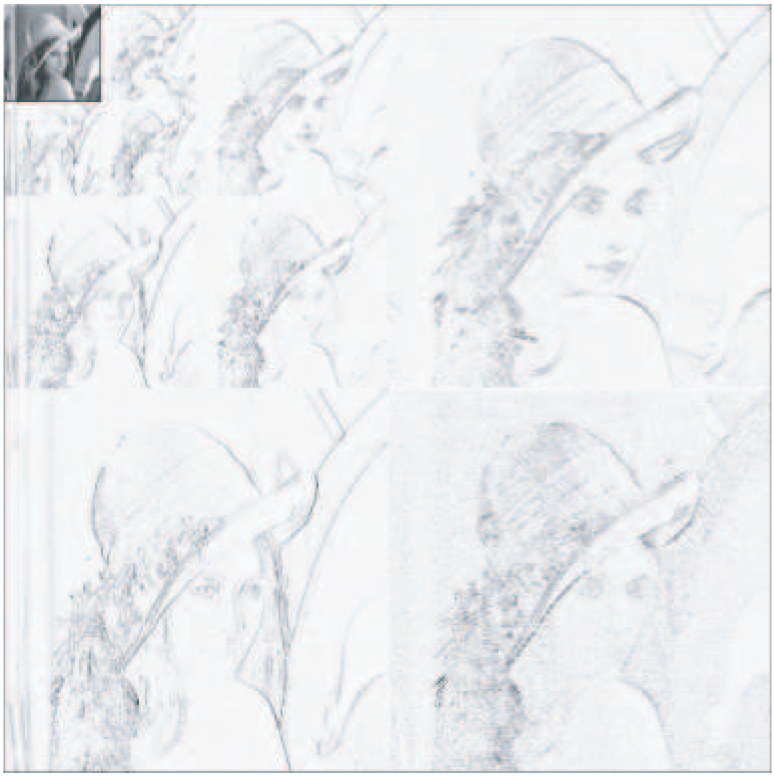

Figure 4. Wavelet Decomposition of an Image Component - 3rd Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

The following are some more examples that illustrate how the horizontal, vertical, diagonal and average components work. Take note of the frame in the picture.

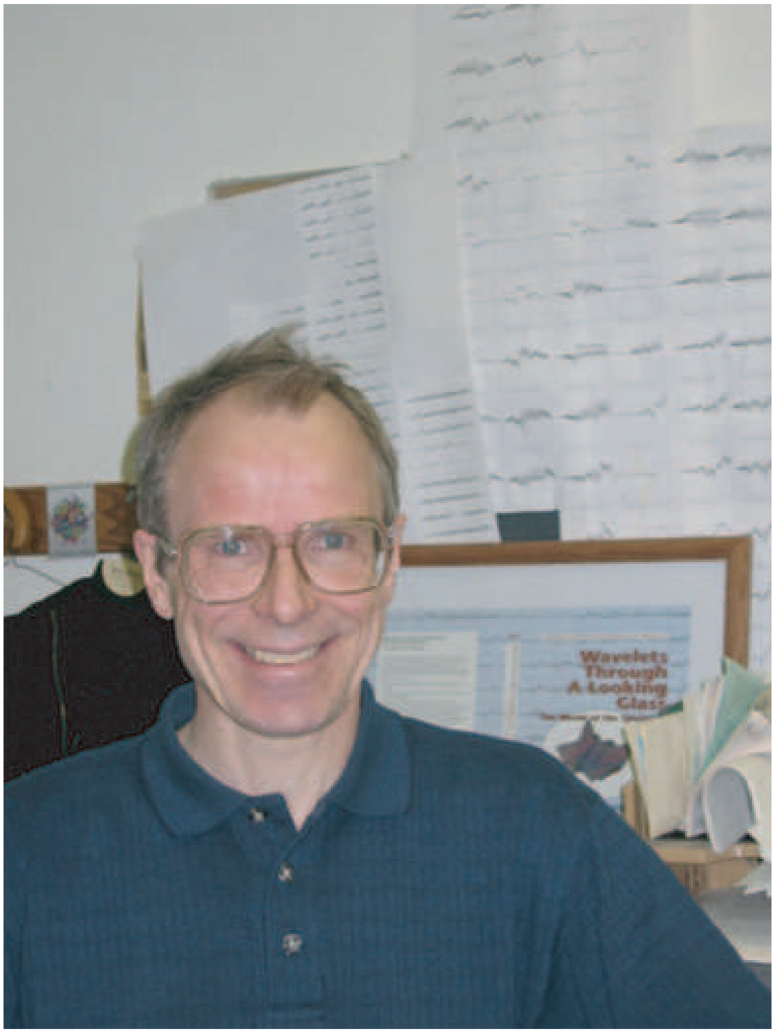

Figure 5. The Original Image Before the Wavelet Decomposition.

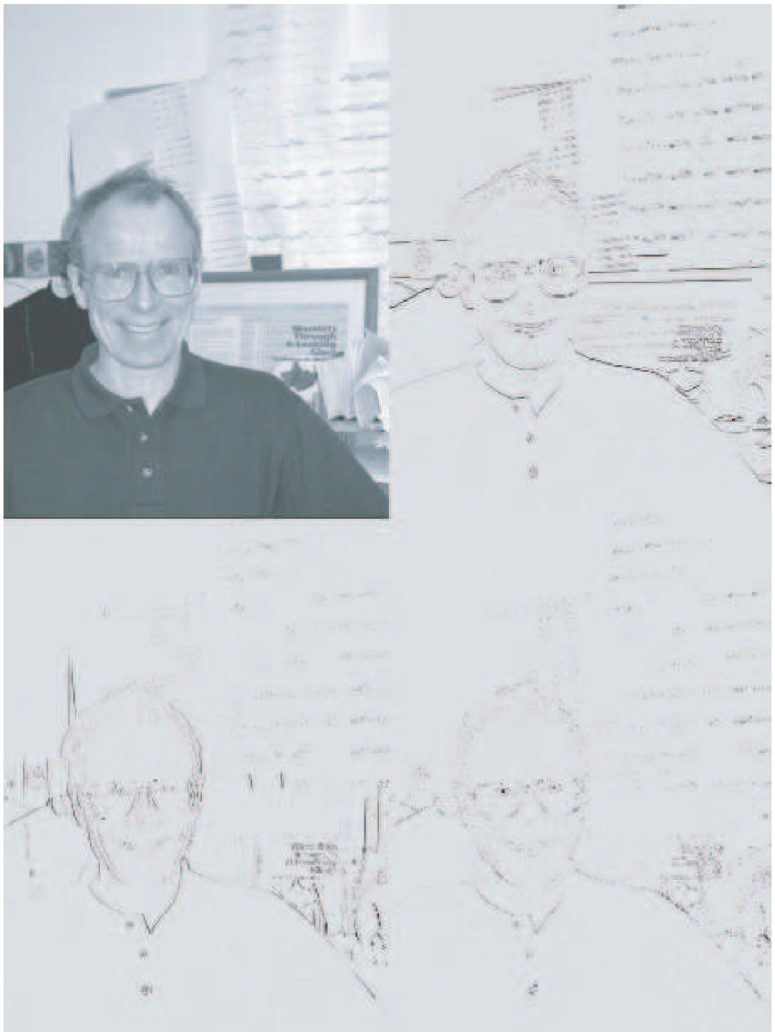

Figure 6. Wavelet Decomposition of an Image Component - 1st Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

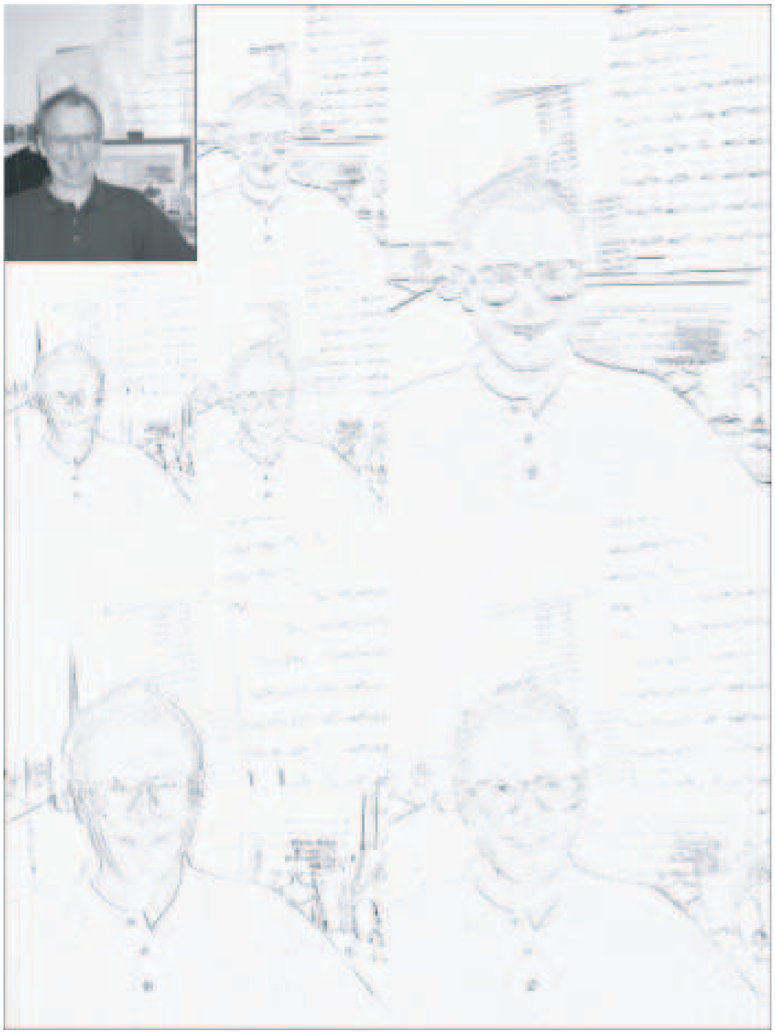

Figure 7. Wavelet Decomposition of an Image Component - 2nd Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

The following example is a simple example where the average, horizontal, diagonal and vertical components are explicitly depicted. As one can see, only the horizontal difference and some horizontalness are detected for the horizonal component and only the vertical difference and some verticalness are detected for the vertical component. As for the diagonal component, one can only see the diagonal difference and the average component carries the 'shape' of the original image throughout.

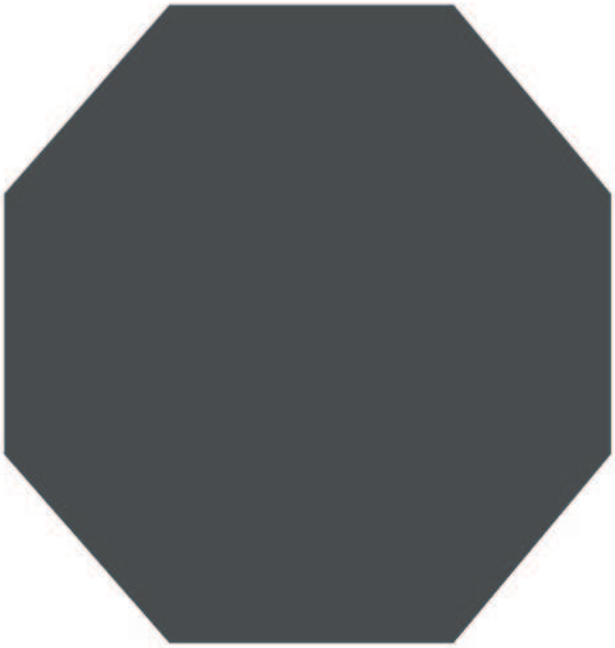

FIGURE 8. The Original Image Before the Wavelet Decomposition. The image has been lightened because of the constrains of the printing process.

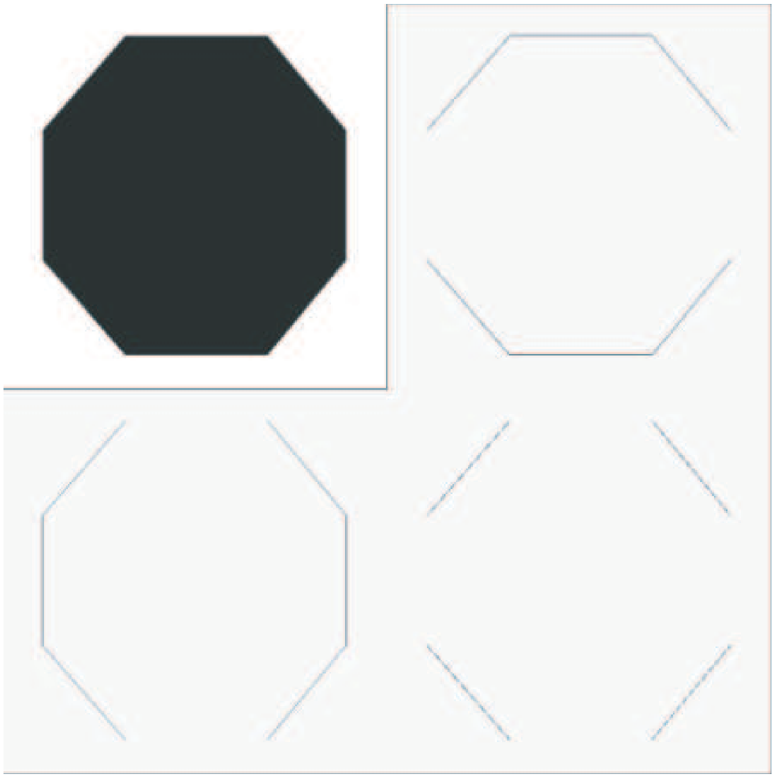

Figure 9. Wavelet Decomposition of an Image Component - 1st Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

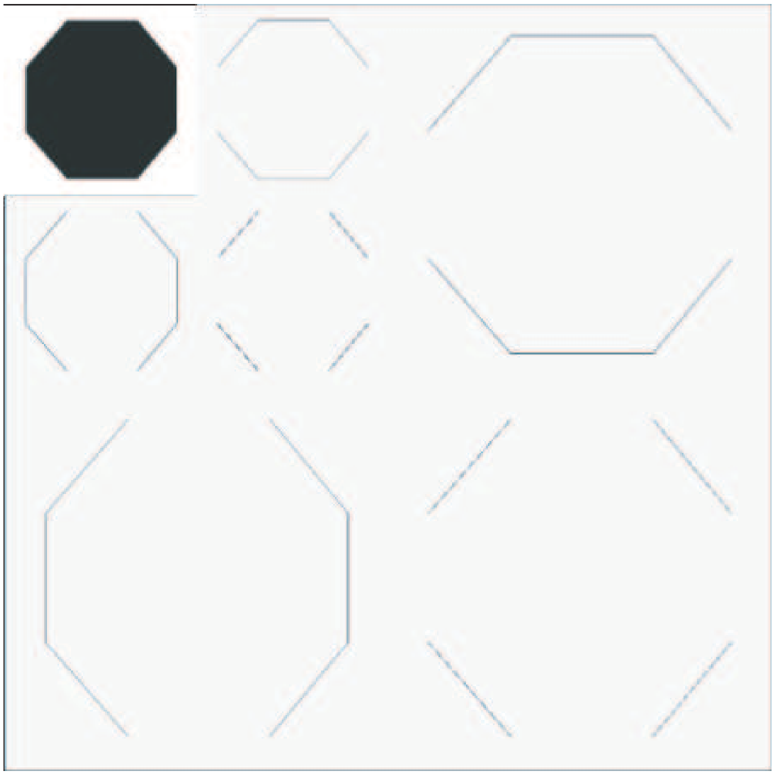

Figure 10. Wavelet Decomposition of an Image Component - 2nd Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

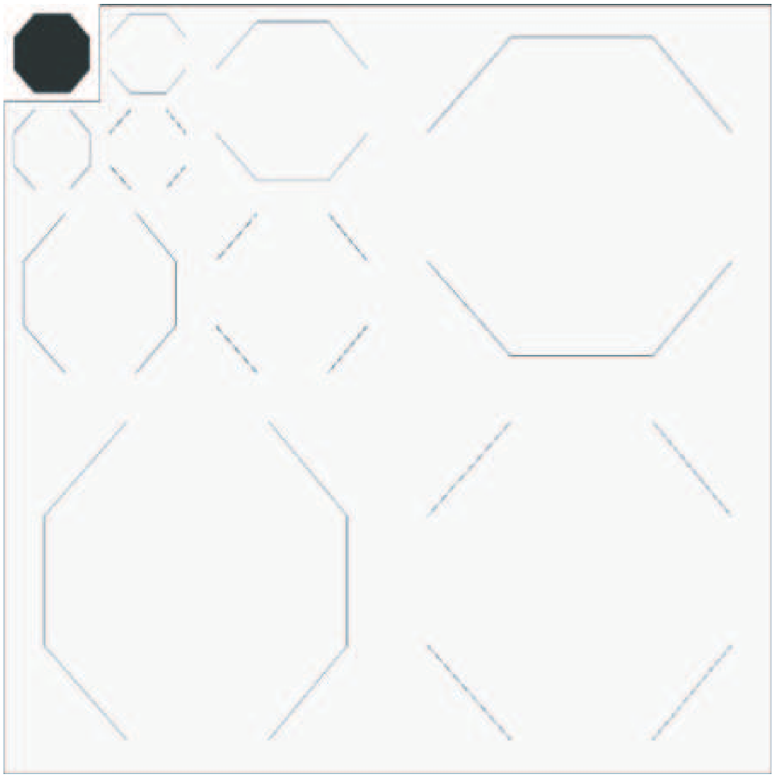

Figure 11. Wavelet Decomposition of an Image Component - 3rd Level Decomposition. The image has been modified: the average detail has been lightened and the horizontal, vertical and diagonal details are shown as negative images with a reversal of white and black, because of contraints of the printing process.

In the process of the computation using MATLAB, it keeps track of two matrices C and S; C is the coefficient decomposition vector:

$$C = \begin{bmatrix} a(n) & h(n) & v(n) & d(n) & h(n-1) & ... & v(1) & d(1) \end{bmatrix}$$

where a, h, v and d are columnwise vectors containing approximation, horizontal, vertical and diagonal coefficient matrices, respectively. C has 3n+1 sections where n is the number of wavelet decompositions. S is an (n+2)x2 bookkeeping matrix:

$$S = \begin{bmatrix} sa(n,:) & sd(n,:) & sd(n-1,:) & ... & sd(1,:) & sx \end{bmatrix}$$

Where sa is the approximation size entry and sd is detail size entry [**Gon04**].

The above process is performed mathematically as follows: f_R, f_G and f_B are treated as vectors of row vectors. For example, for fR we have

$$\mathbf{f}_R = \begin{bmatrix} f_{R_1} \\ f_{R_2} \\ \vdots \\ f_{R_M} \end{bmatrix} where f_i = (f_{i,1}, f_{i,2}, ..., f_{i,N}) \tag{3.8}$$

Then each row vectors in f_R go through the following operation:

$$\begin{bmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,N/2} \\ d_{i,1} \\ d_{i,2} \\ \vdots \\ d_{i,N/2} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ g_0 & g_1 & g_2 & g_3 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,N} \end{bmatrix} \tag{3.9}$$

After performing above operation on each row vector, form a matrix with the resulting row vectors. Multiplying the same matrix to the colomn vector of the resulting matrix would result in (3.6). It can be seen in [**Pre92**] that the above process is done by performing double loops of multiplication of the vector sequence with the matrix.

For the image compression, the quantization process takes place after the wavelet decomposition stage. That is, threshoding 2.1.2 of the matrix takes place thus resulting in data reduction. A more detailed description of this process can be found in [**Wal99**]. Also, see [**Bri03**] and [**Woh03**].

3.4. Mathematical Insights. In this section, we are going to make some changes. H and G in previous sections and chapters will now be denoted as S_0 and S_1. The projection operators S_0 and S_1 are operators in $L^2(\mathbb{R})$, since computers cannot compute infinite integrals we have to convert the mathematical decomposition process to something that can be expressed to computer algorithm. This is made possible by the property that in Hilbert space $L^2(\mathbb{R}) \cong L^2(\mathbb{T}) \cong l^2(\mathbb{Z})$. See [**Dou98**]. Recall that choices are involved in the isomorphisms, e.g. $\cong L^2(\mathbb{T}) \cong l^2(\mathbb{Z})$ is Fourier's choice. Thus an integral can be expressed in terms of sequences. The relationship between projection operators S_0 and S_1 can be expressed as the following matrices where the boxes are the coefficients h_j's as in (3.9).

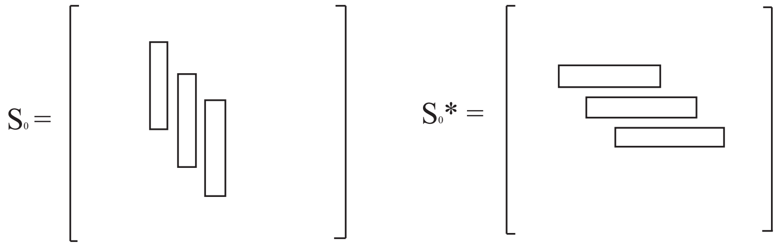

FIGURE 12. S_0 and S_0^*.

V_n and W_n defined in 2.2.1 can now be expressed as follows:

$$V_n = S_0 e_n$$

and

$$W_n = S_1 e_n$$

where $e_n = (0, ..., 0, 1, 0, ...0)$, 1 is in the n^{th} place.

We can compare the operations performed in two different spaces namely, $L^2(\mathbb{T})$ and $l^2(\mathbb{Z})$. See [**Jor05**] for more details about the operator notion.

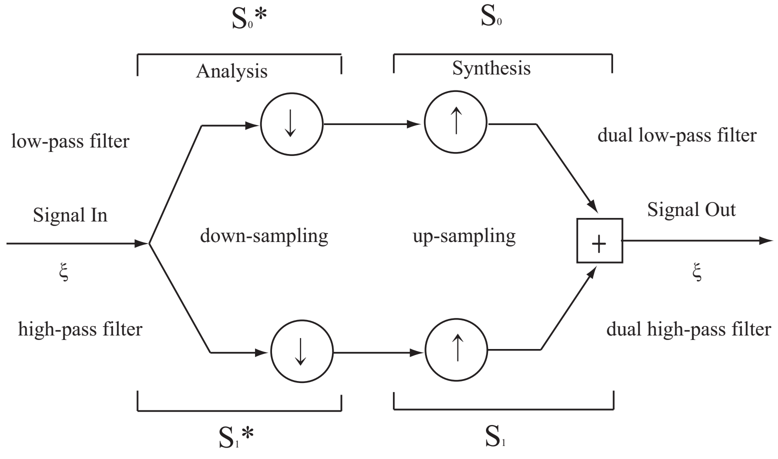

FIGURE 13. For $L^2(\mathbb{T})$.

Perfect reconstruction (Operator notation):

$$S_0 S_0^* + S_1 S_1^* = I$$

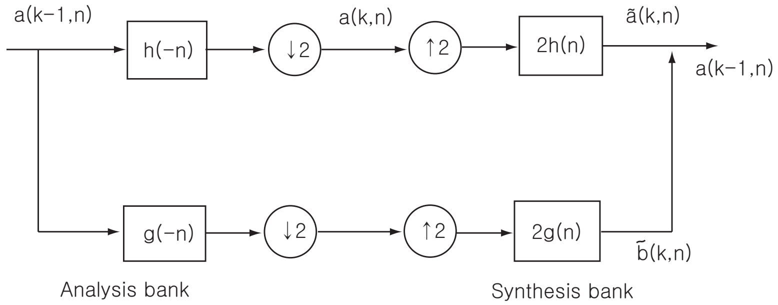

Figure 14. For $l^2(\mathbb{Z})$.

See (2.31), (3.9).

4. Results and Discussion

4.1. Implementation of the Program. The program was implemented using MATLAB with various subroutines that enables the wavelet transformation, image compression and threshold computation from the Wavelet Toolkit .

4.2. Discussion.

4.2.1. *Lossy Compression.* There are various factors that influence the image compression. As mentioned above in section 2, nonorthogonality of the wavelet may cause the compression to be lossy. When threshold is applied to the compression, some of the 'insignificant' coefficients are thrown out thus the resulting in lossy compression. Also, the number of levels the wavelet transform is applied would influence the compression quality. Although the lossiness caused by the nonorthogonal wavelet was not avoidable when certain wavelets were used, an attempt to minimize the lossiness was made for the number of levels part by going down all the way to the single pixel level when the wavelet transform was applied (floor(log2(min(size of Image)))). In addition various threshold values are applied to observe the lossiness.

A lossy compression method tend to produce inaccuracies in the decompressed image. Lossy compression method is used when these inaccuracies are so small that they are imperceptible. If those imperceptible inaccuracies are acceptable the lossy technique is advantageous compared to the lossless ones for higher compression ratios can be attained.

In order to support the claims made by comparison of the resulting images and the theoretical knowledge that we obtained from the texts, some numerical comparisons are made. They are the compression ratio, the root mean square error, rms, the relative two norm difference, D, and the peak signal to noise ratio, PNSR. The formulas used are as follows:

$$ratio = \frac{1}{\frac{X\times Y\times 3-(L^2 normrecoveryin\% X\times Y\times 3/100)}{X\times Y\times 3}} \tag{4.1}$$

$$rms = \sqrt{\frac{\sum_{n=1}^{3}\sum_{i=1}^{Y}\sum_{j=1}^{X}(f_{i,j,n}-g_{i,j,n})^2}{X\times Y\times 3}} \tag{4.2}$$

$$D = \sqrt{\frac{\sum_{n=1}^{3} \sum_{i=1}^{Y} \sum_{j=1}^{X} (f_{i,j,n} - g_{i,j,n})^2}{\sum_{n=1}^{3} \sum_{i=1}^{Y} \sum_{j=1}^{X} f_{i,j,n}^2}} \tag{4.3}$$

$$PSNR = 20 \log \frac{255}{rms} \tag{4.4}$$

See [**Wal99**]

Various wavelet transforms with two different thresholdings were used to compress the and 8-bit color image lena.png. The results are as follows:

One thing that could be noted right away by looking at the images is that the images compressed with smaller threshold value that is 10 look closer to the original lena.png compared to the images compressed with threshold value 20 overall.

Now, looking at the performances of each wavelet transforms given the same threshold value, bior 2.2 (Biorthogonal wavelet), sym5 (Symlet) and Coif3 (Coiflet) seem to have produced the less flawless compressed images compare to all the other wavelets.

Within the Daubechies wavelets db2 appears to have produced the least flawless compressed image; that agrees with what was discussed above in Daubechies wavelets that db2 is being better in signal compression than db1(Haar). Considering the errors and compression ratios as well as the perception of the image sym5 would be the best choice of wavelets, among the ones that was used for the image compression. So, in this case, sym5 being very close to symmetric wavelet did a better job in image compression. Also, having the extra properties as mentioned under the Coiflets section made Coif3 perform better in image compression. Having biorothogonal property also seem to result in better image compression. On the other hand the orthogonal Daubechies wavelets do not seem to perform better than coiflets, biorthogonal wavelets and symlets. See [**Wal99**]

Also, having longer support which is proportional to the order of the wavelet, appears to worsen the performance of the image compression.

With the threshold value 10, when a Daubechies wavelet, db1 was used the compression ratio was 34.2627 while db2 resulted in 38.4340. A Coiflet Coif1 resulted in compression ratio of 37.0173 whereas Coif 3 resulted in compression ratio of 26.8321. Biorthogonal wavelets bior1.1 and bior2.2 gave 34.2627 and 30.2723 for the compression ratio respectively. Symlets sym2 and sym5 resulted in compression ratios of 38.4340 and 34.3523 respectively. Now, with higher threshold value, since more date is being lost, the compression ratio increases. However, the quality of the image diminishes at the same time.

4.3. Conclusion.

Wavelet compression did show remarkable performance especially with smaller threshold value; it was not differentiable in between the original image and the compressed image for some cases.

However, more improvements can still be made. As it is mentioned in [**Sko01**] there is more room for improvement by adding more stages to the compression such as quantization, entropy encoding, etc. Also, we have not covered all the wavelets that is out there, that it cannot be decided as to which one performs the best image compression.

Mathematical aspects of wavelets play a very significant role in differing the results of engineering applications. I hope to study the mathematical properties of wavelets and their applications in various parts of engineering.

References

[Bha97] Bhaskaran V., Konstantinides, K., *Image and Video Compression Standards: Algorithms and Applications*, 2nd ed. Norwell, MA: Kluwer, 1997.

[Bra02] Bratelli, 0. and Jorgensen, P., *Wavelets Through a Looking Glass: The World of the Spectrum*, Birkhäuser, 2002.

[Bri03] Brislawn, C. M. and Wohlberg, B. E. and Percus, A. G., *Resolution scalability for arbitrary wavelettransforms in the JPEG-2000 standard* Visual Commun. & Image Process., ser. Proc. SPIE vol. 5150 **1** (Jul. 2003), 774–784.

[Dau92] Daubechies, I., *Ten Lectures on Wavelets*, SIAM, 1992.

[Dou98] Douglas, R. G., *Banach Algebra Techniques in Operator Theory, Second Edition*, Springer, 1998.

[Gon02] Gonzalez, R. C. and Woods, R. E. and Eddins, S. L., *Digital Image Processing Using MATLAB*, Prentice Hall, 2004.

[Gon04] Gonzalez, R. C. and Woods, R. E. and Eddins, S. L., *Digital Image Processing Using MATLAB*, Prentice Hall, 2004.

[Han00] Hansen, O., *Einführung in die Theorie und Anwendung der Wavelets*, Berlin: Logos-Verl., 2000.

[Jaf01] Jaffard, S. and Meyer, Y. and Ryan, R. D., *Jaffard, S. and Meyer, Y. and Ryan, R. D*, SIAM, 2001.

[Jor05] Jorgensen, P. E. T., *Analysis and Probability Wavelets, Fractals, and Dynamics*, Springer, 2005.

[JPG00] *JPEG-LS (14495) Final CD*, ISO/IET JTC1/SC29/WGI N575, July 1997.

[Kei04] Keinert, F., *Wavelets and Multiwavelets*, Chapman & Hall, CRC, 2004.

[MatlabUG] Misitti, M. and Misitti, Y. and Oppenheim, G. and Poggi, J., *Wavelet Toolbox For Use with MATLAB, Version 2*, The MathWorks, 2002.

[Mar82] Marr, D., *Vision*, W. H. Freeman and Company, 1982.

[Mac01] Mackenzie, D., *Wavelets Seeing the Forest - And the Trees*, Beyond Discovery, Dec. 2001.

[Nad99] Nadenau, M. J., Reichel, J., *Opponent color, human vision and wavelets for image compression*, Proc. 7th Color Imageing Conf., Scottsdale, AZ, 16-19 Nov. 1999, 237–242.

[Pre92] Press, W. H. and Teukolsky, S. A. and Vetterling, W. T. and Flannery, B. P., *Numerical Recipes in C The Art of Scientific Computing, Second Edition*, Cambridge University Press, 1992.

[Rao96] Rao, K. R., Hwang, J. J., *Techniques and Standards for Image, Video and Audio Coding*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[Rao98] Rao, R. M. and Bopardikar, A. S., *Wavelet Transforms Introduction to Theory and Applications*, Addison Wesley, 1998.

[Res98] Resnikoff, H. L. and Wells, R. O. Jr., *Wavelet Analysis*, Springer, 1998.

[Son04] Song, M., *Color Image Compression with Wavelets*, Preprint, 2004.

[Sha93] Shapiro, J., *Embedded Image Coding Using Zerotrees of Wavelet Coefficients* IEEE Trans. Signal Processing **41** (Dec. 1993), 3445–3462.

[Sko01] Skodras, A. and Christopoulos, C. and Ebrahimi, T., *JPEG 2000 Still Image Compression Standard* IEEE Signal processing Magazine **18** (Sept. 2001), 36–58.

[Use01] Usevitch, B. E., *A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000* IEEE Signal processing Magazine **18** (Sept. 2001), 22–35.

[Vet01] Vetterli, M., *Wavelets, Approximation, and Compression* IEEE Signal processing Magazine **18** (Sept. 2001), 59–73.

[Wal99] Walker, J. S., *A Primer on Wavelets and their Scientific Applications*, Chapman & Hall, CRC, 1999.

[Waln02] Walnut, D. F., *An Introduction to Wavelet Analysis*, Birkhäuser, 2002.

[WaEx05] Wolfram Research, *Wavelet Explorer Documentation*, Wolfram Research, Inc., 2005.

[Wic94] Wickerhauser, M. V., *Adapted Wavelet Analysis from Theory to Software*, IEEE Press, A K Peters, 1994.

[Woh03] Wohlberg, B. E. and Brislawn, C. M., *Reversible integer-to-integer transforms and symmetric extenstion to even-length filter banks* Visual Commun. & Image Process., ser. Proc. SPIE **5150** (July, 2003), 1709–1718.

Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL62026, USA

E-mail address: `msong@siue.edu`

URL: `http://www.siue.edu/~msong`