The Author

February 12, 2004

1 Wavelet Definition

Throughout the field wavelet theory there are a few dominant views of what a wavelet is. This document acknowledges two of those definitions and shows the practical strengths and weaknesses of each. Each acknowledge the wavelet as a short segment of a wave. Each recognize the existence of the wavelet function, wavelet pair, and wavelet transform. However, the digital image process and scientific computing views differ in how each of these terms are either defined or derived.

Scientific computing introductions tend to start with a fundamental definition of a wavelet function and derive wavelet pairs and wavelet transform from wavelet functions. The Digital Image Processing and Audio Signal Processing introductions tends to derive the wavelet function, wavelet pairs, and wavelet transform in terms of multi-resolution analysis techniques.

This introduction shows two different approaches to defining wavelet basis functions, wavelet pairs and the wavelet transform and how they are applied. Like most scientific computing introuductions, the fundamental definition of the wavelet function is shown. Next, the concept of multi-resolution is shown in the one dimensional and two dimensional world to provide context. Lastly, the wavelet pair and wavelet transform are introduced to bridge the two ideas. Later sections show a specific form of the wavelet transform, the convolution wavelet transform (CWT) which comparable to the Fast Wavelet Transform (FWT).

1.1 The Wavelet Basis Function

In order to understand the wavelet basis function, wavelet pair, and wavelet transform it is necessary to understand and define the $L^2(R)$ space. This term defines any function whose square is finite along with its integral from infinity to infinity. In equation form:

$$f(x) \in L^2(R) if \int_{-\infty}^{\infty} (f(x))^2 dx < \infty$$

It is this $L^2(R)$ criteria which defines a whole category of functions. First quality of a wavelet basis isit must satisfy the $L^2(R)$ condition. There are many basis which are commonly called a wavelet basis. Some basis are called orthonormal basis and others are called bi-orthonormal. Orthonormal basis have the property of generating a precise mapping form

$$\psi : L^2(R) \rightarrow L^2(R)$$

A second quality of a wavelet basis is that such a basis function must decay quickly as it approaches infinity. Also, these wavelets exhibit properties of being orthonormal. The goal is to represent entire $L^2(R)$ spaces by these wavelets with respect to their binary dilation properties and dyadic translation properties.

A third quality must be satisfied if the wavelet basis to be considered orthonormal. The wavelet basis must be orthonormal to an dilated and or translated version of itself. This quality can be satisfied in either the real or complex number space. In this paper only the real number space shall be addressed.

Three independent writers emphasize this definition (Beylkin, Chui, and Jawerth.) Each define the wavelet function as an orthonormal basis satisfying a dilation and translation scheme. The wavelet function in the context of applied mathematics has a formal definition as stated by Charles Chui.

"A function $\psi \in L^2(R)$ is called an orthogonal wavelet (or o.n. wavelet) if the family $\{\psi_{j,k}\}$ defined

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \forall j, k \in Z$$

is an orthonormal basis of $L^2(R)$ where $\langle \psi_{j,k}, \psi_{l,m} \rangle = \delta_{j,l}\delta_{k,m}$, $\forall j, k, l, m \in Z$ and every $f \in L^2(R)$ can be written as

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{j,k}\psi_{j,k}(x)$$

where the series convergences and is in $L^2(R)$ such that

$$\lim_{M_1, M_2, N_1, N_2} ||f - \sum_{j=-M_2}^{N_2} \sum_{k=-M_1}^{N_1} c_{j,k}\psi_{j,k}|| = 0$$

The simplest example of orthogonal wavelets is the Haar Transform."[18] Also, no author has be found yet to have an orthogonal wavelet in the $L^2(R)$ space. Others may exist in $L^2(C)$ space, but that complex number space is not to be explored by this paper.

1.2 Multi-Resolution and Wavelet Basis Pairs: A Splitting Concept

Multi-resolution analysis (MRA) and multi-resolution expansion (MRE) are the primary schemes are used in both image processing for either feature extraction and/ or image compression. In the one dimensional world, MRA and MRE are used isolate octave bands for their properties and contributions. MRA consists of schemes to generate averaging estimates of some function (signal or image) such that the average representation is smaller than the original by some integer amount. This averaging process is continued until a small enough size is reached for the analysis being conducted. Estimate correction factors are used translation from averages to the next larger average to recover an function larger than the function being worked on. The elements of average functions are averaging terms. Likewise, the elements of the estimate correction factor function are called differencing terms.

There are simple ways to visualize this concept. In the one dimensional form, the visual term is a triangle. In this triangle, layers of functions are stacked up from the base to the apex. The original function is always placed on the bottom. Subsequent average terms are placed between the base and the apex. For every level between the apex and the base, there is an averaging method to map that level to the adjacent level closer to the apex. Also, there exist a set of difference elements such that when the average is expanded it can be mapped to the adjacent level toward the base. The apex has no adjacent level to average to. The base has no difference elements to map to a larger level.

The two dimensional version is similar, and is simply expanded to a pyramid. In both cases, the average term is expanded to bring it closer to the base level. The difference terms are define to correct errors between the estimate and the original by simple linear operations.

To acquire each estimate by wavelets, the wavelet averaging basis designated ϕ is introduced. Like its wavelet differencing term, the averaging term is defined as a mother basis function which can be dilated and translated. This averaging term is used as a filter in either a convolution or spatial filtering scheme to map one matrix to the other.

The wavelet basis happens to be the basis function for the differencing function. Qualities of the averaging and wavelet basis is that they must be orthogonal, and the differencing term must satisfy the wavelet basis function criteria. This wavelet basis function is defined as ψ. The two basis functions together form a wavelet basis pair.

In the one dimensional form, the multi-resolution schemes have the effect of applying high and low pass filters to the input. Thus there is an analogus to time and frequency domains with the one dimensional wavelet transform. In the two dimensional wavelet transform, the analogus is the average intensity relative to a points neighbors and the intensity difference with a points neighbors.

1.3 Wavelet Transform, where the concepts become practical

The concept of the wavelet function and wavelet basis pair produce a practical multi-resolution technique called the Wavelet Transform. The Wavelet Transform uses the wavelet pairs to generate functions whose elements are coefficients of the averaging and wavelet functions. The functions together produce similar function to the original. Also for every Wavelet Transform exists an inverse that takes two functions, joins them, and results in an identical function to the original. There are several methods to perform the Wavelet Transform.

One form is the integral wavelet transform (IWT) described by Chui. Another two are discrete wavelet transform (DWT) and the fast wavelet transform (FWT). The convolution wavelet transform (CWT) is a general form of the FWT. In the case of the CWT, any proper wavelet pairs can used to generate corresponding average and difference terms. Application of the CWT is described in the section "Wavelet Transform via Convolution" of this thesis.

2 Wavelet Transform via Convolution

The wavelet transform is defined in terms of average and difference components. Each component can further be transformed to isolate properties of each component. Typically, each component has the form $S \rightarrow (A|D)$ where S is the original signal, A is the average component, D is the difference component and $(A|D)$ the signal A concatenated with D. Each of these components are produced by an orthonormal basis. Also, these components are produced such that the original can be constructed easily from them.

Many mathematicians such as Walker[3], use a form that eliminates half of the values. Thus a form can be defined which has the same number of elements as the original. The rules for choosing the member elements are dependent on the wavelet filter choice.

Another useful property of wavelets via convolution is the simplicity of the operation. The general case works for all. Such an algorithm requires one nested loop as seen below:

$\forall i \in [0, M)$
    $\forall j \in [0, N)$
        `n=i-j`
        `if` $(n \in [0, M))$
            $y_i += x_n \cdot h_j$

This filter simply equates to the mathematical function: $x * h = \sum_l h(l)x(k-l)$, which is the convolution operation. As we can see the operation is slightly less than $O(N^2)$. For practical use, the filter is made smaller than the actual signal being analyzed. In some cases, the filter may be much smalled than the signal. Filter size matters in extracting features from the original signal.

In the case of this convolution operator, the limit is actually M, not N. The value of M is the size of the original signal. Since the resulting vector is the same as the original, the vector is said to be fully qualified. Only half of those values are necessary to reconstruct the original (every other element).

To perform a wavelet transform via convolution, each signal is convolved twice.

$$A_i = A_{i-1} * V, \qquad D_i = A_{i-1} * W$$

where

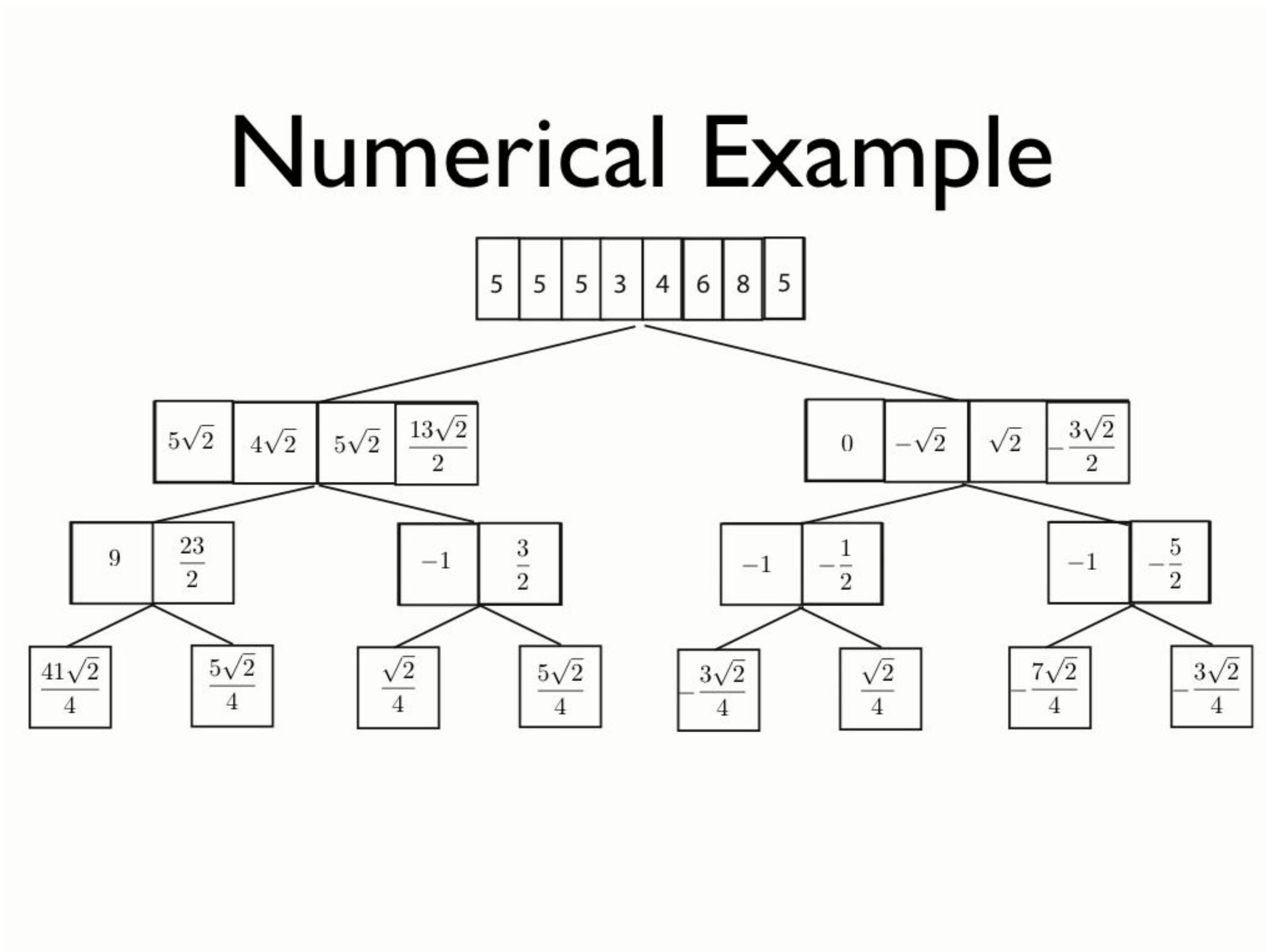

Figure 1: The Haar Transform performed a sample function show each step of the transform in wavelet packets

- V is the scaling wavelet vector,
- W is the differencing wavelet vector,
- A is the average vector (scaled vector),
- D is the difference component vector,
- $\forall i \in [1, L)$ and $A_0 = f$ which is the original signal, and
- L is the limit on the number resolutions that signal can have based on the wavelet type.

3 Class 2D Wavelet: Complete Form

The convolution version can be used to derive a Wavelet Matrix. For a general case, it is simpler to use the convolution method. The matrix form becomes practical in repetitive special case applications.

The 2D transform has four components: the average, vertical, horizontal, and diagonal. Two general computational means exist to generate a one-resolution transform. These can derive means for performing many resolutions.

A complete transform method returns a result matrix which is the same size as the source matrix. The result contains the four components. Each component resides on 4 corners of the matrix. Given a matrix B, the transform is to yield the following form:

$$B \Rightarrow \begin{pmatrix} H & D \\ A & V \end{pmatrix}$$

where A is the average component, H is the horizontal component, V is the vertical component, and D is the difference component. There is another form which is also used as an example:

$$B \Rightarrow \begin{pmatrix} A & V \\ H & D \end{pmatrix}$$

The first version is simple in concept, but provides a few more possibilities for error and confusion. Regardless of the case, the four components have the following definitions:

1. Average component: produced by filtering the row vectors and the column vectors with the averaging filter.
2. Vertical Component: produced by applying the average filter to the column vectors and the difference filter to the row vectors.
3. Horizontal component: produced by applying the average filter to the row vectors and the difference to the column vectors.
4. Diagonal component: produced by applying the difference filter to both the row and column vectors.

3.1 Proof of Concept

Two methods of convolving a matrix are easily conceived. First is to use 1D wavelet. The other is to apply the convolution scheme straight to the matrix. Included in the wavelet experiment are both. Realistically, both can and do achieve the same result. However, the direct method achieves speed advantages by the lack of overhead. The direct method only stores a temporary vector resident in memory. Also, there are two fewer transfers per row and column.

3.1.1 1D to 2D Method

Both rows 1D and 2D and columns 1D and 2D transform are performed similarly. The obvious difference is the indexing of rows and columns.

Given: 1D wavelet transform source matrix

Algorithm: (Row Transform)

$\forall i \in rows$

- $\forall j \in columns$
- $S[j] \leftarrow source[i][j]$
- $S \Rightarrow^{W} R$

This principle of this algorithm is simple. Only three intuitive steps are necessary per row or column. Two of these steps are array transfers (row/column transfer to an array). These arrays are fed into the 1D transforms.

However, the 1D wavelet transform itself includes a series of memory allocation and deallocation operations. Each memory call is at the minimum a system call.

3.1.2 Vector - Matrix Method

The principle of this algorithm is more complicated. All functionality, such as convolution, is built into the method. There are fewer calls and passing of structures to external functions to compute the transform.

This method has a few givens. The source matrix, the Haar average filter, and the Haar difference filter are given arguments. The result argument is the return argument. The transform signals sub-function row transforms and column transforms to perform the work.

The algorithm is as follows for the row transform (and is similar for the column transform):

$\forall i \in rows$

1. Initialize temporary array/vector to all zeros (x).
2. $\forall k \in columns$
   (a) $\forall l \in ha.Size$
3. $\forall k \in columns/2$
   $result_{i,k} = x_{2k+1}$ (In other words, odd split)
4. Initialize x to all zeros.
5. $\forall k \in columns$
   (a) $\forall l \in hd.Size$
6. $\forall k \in columns/2$
   $result_{i,k+columns/2} = x_{2k+1}$ (In other words, odd split)

3.2 Multiresolution

The multiresolution wavelet transform and the inverse multiresolution transform resemble the vector-matrix version. All functionality is built into this method. However, there are structural changes.

The wavelet transform (multiresolution) uses private members of the class (hA, hD, xD/yD, xA/yA). Both Haar filters are maintained this way. Also both row and column transforms have average and difference myVector classes for temporary storage. All of these members are allocated and destroyed by the wavelet transform method itself. The simplified algorithm of the row transform is the following:

1. Initialize xA and xD to zero
2. $\forall k \in columns, \qquad \forall l \in filter$
   - $n = k - l$
   - if ($n \in columns$)
     $xA_k = W_{i,n} * hA_l$
     $xD_k = W_{i,n} * hD_l$
3. Transfer back to W
   $W_i = xA|xD$

And the column transform is represented by:

1. initialize yA and yD to zero
2. $\forall k \in rows, \qquad \forall l \in filter$
   - $n = k - l$
   - if ($n \in columns$)
     $yA_k = W_{i,n} * hA_l$
     $yD_k = W_{i,n} * hD_l$
3. Transfer back to W
   $W_j = yA|yD$

Note: W_i names the row vectors and W_j names the column vectors, and $W_{i,j}$ is the element from the i^{th} row and j^{th} column.

3.3 Computational Cost

The cost of this algorithm is computed first for each row and each column. This value is used to compute the cost of the matrix. The cost of computing the matrix is used to compute the cost of the multiresolution steps. Per row the cost is $3K$, where K is the number of columns. Per column the cost is $3L$, where L is the number of rows. For the whole matrix, one resolution costs $6KL$ operations to compute the wavelet transform. Per resolution, the rows and columns shrink by 2^i for each resolution, i, performed. The limit of this cost equals $12KL$ operations. Thus, the cost is linear.

4 Results - 1D Wavelet Transform

Testing of the 1D wavelet was performed on a sinusoidal wave form of 128 elements. The given function has the equation shown in Figure 2.

$$y(n) = 10\sin\left(\frac{n}{128}\right) - 5\sin\left(\frac{n}{64}\right) + 2\sin\left(\frac{3n}{128}\right) - \sin\left(\frac{n}{32}\right)$$

The first version of the 1D transform used the even elements of both convolutions to generate the wavelet transform. These even elements came from the over-complete form and naturally allow the potential to have complete information. However, in doing so, a fundamental flaw appears.

In order to evaluate the effectiveness of the wavelet transform three tests have been devised. First, energy equivalence is used to determine how much energy is retained in the transform from the original. The general shape is used on a the first resolution to test if the average signal has the same general shape as the original. Lastly, the inverse transform is used to recover the original signal. A comparison is made between the original and the recovered signal.

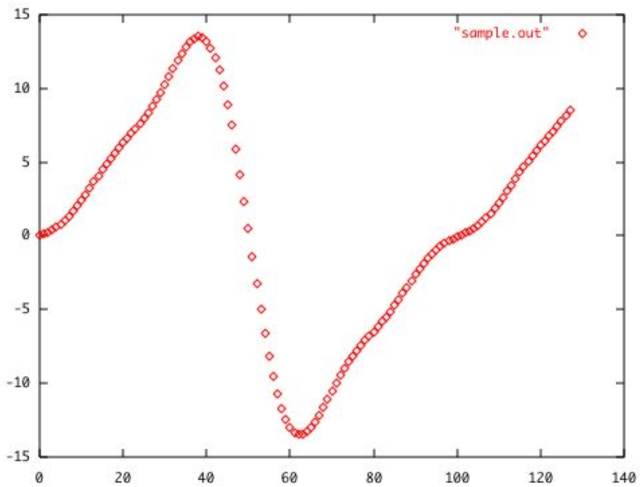

Figure 2: Sample function. The x-axis is the array index (index n). The y value is simple – the value $y(n)$.

After one resolution, the transformed signal has the same energy as the original. This is good since it allows the original to be recovered from the transform. Also, the average component of the transform has the same shape as the original. This is good. However, the recovered signal is missing the last element. Refer to Figure 3. The secret is in which elements are used from the over-complete to make the complete. The over-complete in this project comes from the average component and difference which are simply the result of convolution.

The convolution means is at the heart of the issue. The convolution operator in this case starts with the first element of the filter against the first element of the signal. In the simple Haar Wavelet case, there is a transformation pairing

$$(S_i, S_{i-1}) \to A_i, \text{ and } (S_i, S_{i-1}) \to D_i.$$

In this pairing with zero indexed signals, the odd indexed elements from the over-complete must be used to have all elements of the original accounted for.

Also this produces a functional difference between wavelet inverse transform for odd and even versions. The difference is slight; however, the last element is lost in the even indexed form.

$$\begin{array}{lll} \text{Odd:} & R_{2i} = (A_i - D_i)\sqrt{1/2}, & R_{2i+1} = (A_i + D_i)\sqrt{1/2} \\ \text{Even:} & R_{2i} = (A_i + D_i)\sqrt{1/2}, & R_{2i-1} = (A_i - D_i)\sqrt{1/2} \end{array}$$

An odd indexed wavelet transform yields the same energy. However, all of the values are accounted for. Refer to Figure 4.

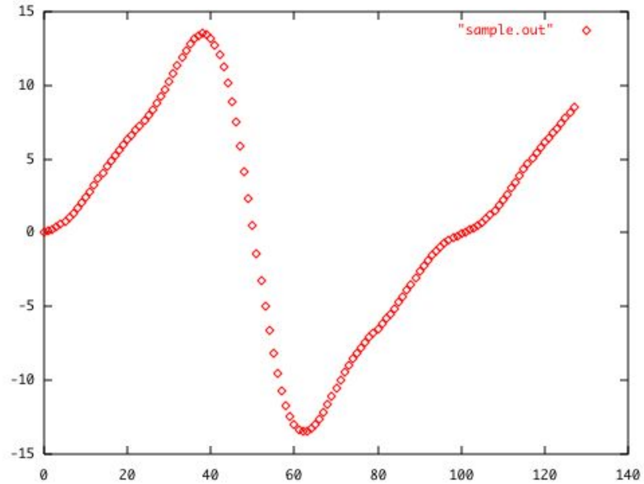

Figure 3: Recovered function. The x-axis is the array index (index n). The y value is simply the value $y(n)$. The function was recovered from an even indexed wavelet transform.

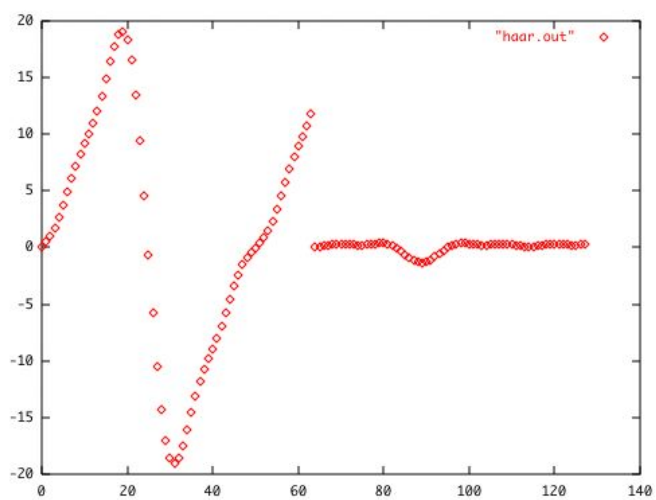

Figure 4: Recovered function. The x-axis is the array index (index n). The y value is simply the value y[n]. The function was recovered from an odd indexed wavelet transform.

5 Results: 2D Wavelet Transform

A simple room picture shows the difference correct indexing produces in the wavelet transform and its inverse. The 1D to 2D method shows the incorrectly indexed case. A correctly indexed version is shown in the vector-matrix method.

The 1D to 2D method has a serious issue with memory leak errors (Macintosh OSX, using gcc 3.1). Memory is allocated and deallocated quickly, and on some platforms shows up as an error. On other platforms, the result is degraded performance (IRIX, SGI Octane2 using gcc 2.9). An example image of 720 x 486 requires nearly 10 minutes to compute the wavelet transform by this method on an SGI Octane2. However, this method does eventually return a correct result.

The matrix-vector method also yields the correct result. However, there is less memory overhead in this method as compared to the 1D to 2D method. As a result, both the row wavelet transform and column wavelet transforms are performed more quickly, with fewer memory transfers and allocations. Obviously, this also allows for the operation to be conducted almost entirely in cache memory on both the SGI Octane2 and Macintosh G4 based machines. A Macintosh G3 based machine still requires main memory at a minimum to execute the same operation which yields a slower performance.

A correct result must also be matched to a correct inverse method. The indexing order matters. The inverse transform method is a forward inverse transform method. In the case of 1D to 2D transform, the ordering was reverse indexed (Figure 7). As a result of an error in indexing, ringing is seen on edges in this method(Figure 6) for a case in point. Caution is incredibly important when matching both forward and reverse indexing, since matching the mathematics to the actual ordering can be obscure and tricky.

A correct result is shown in Figure 9. In this case, the indexing was matched up and ringing is not present. It is clear that the recovered image and the original (Figure 5) are nearly indistinguishable.

5.1 Multiresolution

The expected result is a picture within a picture. Each average component has a further transform on it. The three resolution transform has the form:

$$W_3 = \begin{pmatrix} \begin{pmatrix} \begin{pmatrix} A_3 & V_3 \\ H_3 & D_3 \end{pmatrix} & V_2 \\ H_2 & D_2 \end{pmatrix} & V_1 \\ H_1 & D_1 \end{pmatrix}$$

Refer to Figure 10 for the image transform results.

To obtain the inverse, an exact reverse procedure is necessary, otherwise the distortion is hideous. The first attempt of the wavelet inverse transform was out of order, refer to Figure 11 . A correct picture was obtained during the second attempt. Correct order yielded correct results, refer to Figure 6.

Figure 5: Original Image. This image is the original image.

5.2 Threshold Filtering

After a triple resolution, a 0.02 threshold will eliminate 81.1706 percent of elements in the original sample picture. Also at this point, the effects of removing these elements becomes visually evident (Figure 12). At a 0.01 threshold, 66.0205 percent of the elements are removed. Visually, the recovered sample and the original appear to be the same (Figure 15). At a threshold of 0.1, 92.9987 percent of the elements are reduced to zero. However, the distortions are clearly visible at this level of thresholding (Figure 13). Even at a threshold of 0.001 which is below the numerical precision of the original, 16.0814 elements are reduced to zero. At a threshold of 0.002, 28.9683 percent is removed.

Consequently after a triple resolution, nearly 29% of the data was irrelevant for the image's brightness resolution (which also applies to color). Subjective examination reveals that removing 60% to 85% of the data was not noticeable to human perception. Which leaves only 15% to 40% of the data actually contributing or being necessary to reconstruct the image.

References

[1] Charles Chui *An Introduction to Wavelets* published by Academic Press San Diego, CA 92101-4495 copyright 1992

[2] Howard L. Resnikoff. and Raymond O. Wells, Jr. *Wavelet Analysis: The Scalable Structure of Information* copyright Springer-Verlag New York, Inc. New York, NY 10010, USA, 1998

[3] James Walker *A Primer on Wavelets and Their Scientific Applications* copyright Chapman & Hall/CRC : Boca Raton, FL, USA, 1999

[4] Gavin Tabor *Wavelets - The Idiots Guide* http://monet.me.ic.ac.uk/people/gavin/java/waveletDemos.html

Figure 6: Recovered Image. This image is the recovered image. Depending on whether the image was saved as a picture first can affect the white spots in the picture. Ringing is also an issue.

[5] Amara Graps *Introduction to Wavelets* http://www.amara.com/current/wavelet.html

[6] Singiresu S. Rao *Applied Numerical Methods for Engineers and Scientists* published Prentice Hall Upper Saddle River, NJ 07458 copyright 2002

[7] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery

Numerical Methods in C Published by the Press Syndicate of the University of Cambridge The Pitt Building, Trumpington Street, Cambridge CB2 1RP 40 West 20th Street, New York, NY 10011-4211, USA Copyright Cambridge University Press 1988, 1992

[8] G. Beylkin *On wavelet-based algorithms for solving differential equations*

[9] G. Beylkin *Wavelets and Fast Numerical Algorithms*

[10] G. Beylkin, R. Coifman, and V. Rokhlin *Fast Wavelet Transforms and Numerical Algorithms I, Article in Communications on pure and applied mathematics* copyright 1991 Wiley, New York

[11] G. Beylkin, D. Gines and L. Vozovoi *Adaptive Solution of Partial Differential Equations in Multiwavelet Bases*

[12] Stanly J. Farlow *Partial Differential Equations for Scientists and Engineers* copywrite 1993 Dover Publications, Inc Mineola, N.Y. 11501

[13] Murray R. Spiegel, *Theory and Problems of Advanced Mathematics for Engineers and Scientist* copy-write 1996 McGraw-Hill

[14] Stephane Jaffard, Yves Meyer, and Robert D. Ryan *Wavelets: Tools for Science and Technology* copyright 2001 by the Society for Industrial and Applied Mathematics Philadelphia, PA 19104-2688

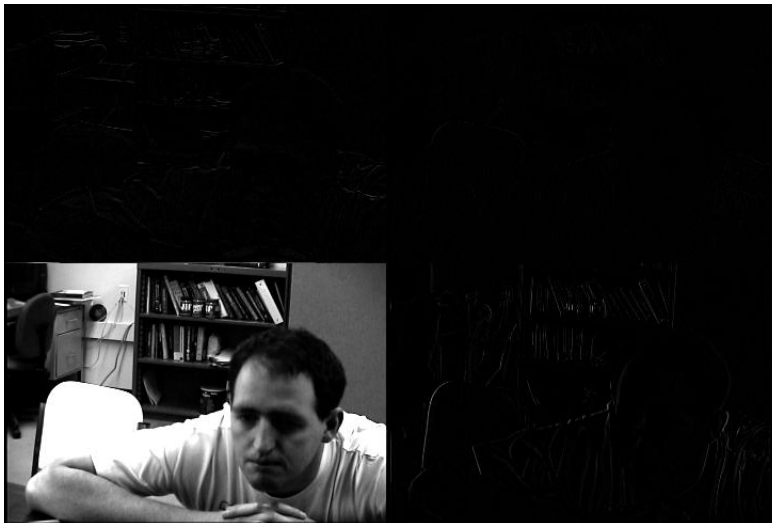

Figure 7: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components.

[15] Mladen Victor Wickerhauser *Adapted Wavelet Analysis from Theory to Software* copyright 2001 by the Society for Industrial and Applied Mathematics Philadelphia, PA 19104-2688

[16] n-Bing Lin and Zhengchu Xiao *Multiwavelet Solutions for the Dirichlet Problem* Department of Mathematics, University of Toledo; Toledo, OH 43606, USA

[17] ian-Xiao He *Wavelet Analysis and Multiresolution Methods (Lecture Notes in Pure and Applied Mathematics* published Marcel Dekker, Inc. New York, NY 10016 copyright 2000

[18] Charles Chui *An Introduction to Wavelets* published by Academic Press San Diego, CA 92101-4495 copyright 1992

Figure 8: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components, using the vector-matrix version.

Figure 9: Recovered Image (Vector-Matrix Method). This image is the recovered image. This version avoids the ringing by using the vector-matrix version which is more aligned for the inverse wavelet transform.

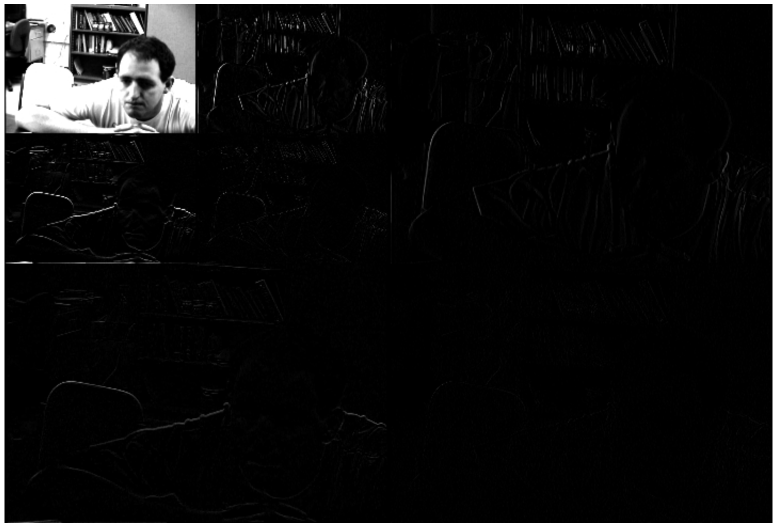

Figure 10: Wavelet Transform Image. This image is divided in to average, horizontal, vertical and diagonal components using multiresolution wavelet transform. Note the the average component was transformed one step further.

Figure 11: Recovered Image - Wrong Order (Multiresolution). This image shows a 2D wavelet transform after it was recovered out of order. Obviously, the distortion is hideous.

Figure 12: Recovered Image - 2% threshold (Multiresolution). This image had nearly 83% of its elements removed in the triple resolution wavelet transform.

Figure 13: Recovered Image - 10% threshold (Multiresolution). This image had nearly 93% of its elements removed in the triple resolution wavelet transform.

Figure 14: Recovered Image - 5% threshold (Multiresolution). This image had nearly 85% of its elements removed in the triple resolution wavelet transform.

Figure 15: Recovered Image - 1% threshold (Multiresolution). This image had nearly 60% of its elements removed in the triple resolution wavelet transform.