The Effect of Wavelet Families on Watermarking

Evelyn Brannock, Michael Weeks, Robert Harrison
Department of Computer Science, Georgia State University, Atlanta, GA, USA
Email: {evelyn, mweeks, rharrison}@cs.gsu.edu

Abstract— **With the advance of technologies such as the Internet, Wi-Fi Internet availability and mobile access, it is becoming harder than ever to safeguard intellectual property in a digital form. Digital watermarking is a steganographic technique that is used to protect creative content. Copyrighted work can be accessed from many different computing platforms; the same image can exist on a handheld personal digital assistant, as well as a laptop and desktop server computer. For those who want to pirate, it is simple to copy, modify and redistribute digital media. Because this impacts business profits adversely, this is a highly researched field in recent years. This paper examines a technique for digital watermarking which utilizes properties of the Discrete Wavelet Transform (DWT).**

The digital watermarking algorithm is explained. This algorithm uses a database of 40 images that are of different types. These images, including greyscale, black and white, and color, were chosen for their diverse characteristics. Eight families of wavelets, both orthogonal and biorthogonal, are compared for their effectiveness. Three distinct watermarks are tested. Since compressing an image is a common occurrence, the images are compacted to determine the significance of such an action. Different types of noise are also added. The PSNR for each image and each wavelet family is used to measure the efficacy of the algorithm. This objective measure is also used to determine the influence of the mother wavelet. The paper asks the question:"Is the wavelet family chosen to implement the algorithm of consequence?"

In summary, the results support the concept that the simpler wavelet transforms, e.g. the Haar wavelet, consistently outperform the more complex ones when using a non-colored watermark.

Index Terms— **watermark, digital watermarking, image watermarking, wavelet, discrete wavelet transform,**

I. Introduction

In the past, the U.S. has attempted to utilize copyright and trademarking law to protect creative digital content. However, legal means have not proven to be sufficient, and it can be an almost impossible task to hold the ownership of these works in the author's control and curb the illegal theft of these labors. Watermarks embed a symbol of the owner of the works into the item itself, still allowing innocent consumers to continue to enjoy creative content. However, the hope is that the capability to identify those who nefariously copy, for example, music files from a CD to sell on the black market, can be identified, and perhaps more successfully prosecuted. Businesses lose unknown profits from those who are willing to inexpensively reproduce artistic digital data, such as movies on DVDs and music on CDs in great volumes and instantaneously distribute worldwide to sell, without the right to do so. Therefore, creators and owners of the work are concerned that unauthorized copying and redistribution of their copyrighted works causes their economic returns to decline. As a result it has become of more significant consequence to study and find the most effective approaches to solve this problem. Watermarks serve to identify the source of the content and thus aid in investigating abusive duplication.

Obviously, it is vital that the identifying marks which unassailably establish the true owner of the data are legible and identifiable: otherwise what is the use of embedding these proprietorships? Therefore, the impact of the size and nature of the data on the robustness of the embedded watermark will be investigated, in an extension of [1]. For the purposes of this study, an uncomplicated key will be used. However, as in other cryptographic systems, for commercial applications it should be large enough to make hacking attempts as unachievable as possible. Also, the effect of extensive search attacks on the watermark will be given a detailed examination.

In this paper, the next section will cover the background and digital watermarking principles, section III will cover wavelets, and then section IV will discuss the method used. Section V presents results, and section VI concludes the paper.

II. Digital Watermarking

A. Definition

A digital watermark is "a digital code unremovably, robustly, and imperceptibly embedded in the host data and typically contains information about origin, status, and/or destination of the data", according to Hartung and Kutter [2]. It is a form of steganography, because it hides the embedded data, often without the knowledge of the viewer or user. Since the purpose of steganography is the secret communication between two persons, the watermark can be considered to have been successfully attacked if its existence is determined. When contrasting with steganography, watermarks add the property of robustness, which is the ability to withstand most common attacks [1]. The two common categories of attacks are

This paper is based on "Watermarking with Wavelets: Simplicity Leads to Robustness," by E. Brannock, M. Weeks, and R. Harrison, which appeared in the IEEE Proceedings SoutheastCon 2008, Huntsville, AL, USA, April 2008. © 2007 IEEE.

removing the watermark and rendering the watermark undetectable [3]. Attack categorization may include (but are not limited to) compressing the image, adding noise, applying transforms such as translation, rotation and scaling, using linear filtering such as low-pass filtering, and rerecording or recapturing the signal, which makes extracting the watermark is nearly impossible [3], [4], [5].

The watermark has a long and distinguished history. The oldest watermarked paper has been dated back to the 13th century when founded in Italy and was used to distinguish the format, quality and price of paper produced by artisans [6]. Competition between over 40 paper-makers was stiff, and paper-making artisans needed to protect their provenance [6]. Only the media is different for today's watermarks.

B. Types and Applications

There are two main types of watermarks. A blind (or public) watermark is invisible, and is extracted "blindly" without knowledge of the original host image or the watermark itself. The second is non-blind (also asymmetric marking or private), in which the watermark is embedded in the original host, and is intentionally visible to the human observer. The original data is required for watermark extraction [1] [2]. The blind watermark has many more applications than the visible watermark. The subject of this paper is a blind watermark. The only requirements for extracting the watermark are the watermarked image and the key used to embed the watermark in the image. The watermarked image may have been subject to modifications, such as those detailed in section II-A.

There are various watermarking applications for images, as listed below [2], [5], [7].

- **Copyright protection** embeds the copyright owner information in the image. This information is used in order to prevent others from alleging ownership of the image. This is the most common use of watermarking applications currently.
- In an **image authentication** application the intent is to detect alterations to the data. The image's properties, such as it's pixel averages, maximums and minimums, are embedded and compared with the current images for modifications.
- **Prevention of unauthorized copying** is accomplished by embedding information about how often an image can be legally copied. An ironic example in which the use of a watermark might have prevented the wholesale pilfering of an image is in the ubiquitous "Lena" image, which has been used without the original owner's permission.
- **The fingerprint** embeds information about the legal receiver in the image. This involves embedding a different watermark into each distributed image and allows the owner to locate and monitor pirated images that are illegally obtained.

C. Watermarking Techniques

All watermarking methods share a watermark embedding system and a watermark extraction system [1] [2]. Figure 1 shows how the watermark is embedded into the image (on the left), and how the watermark can be later extracted from the image (on the right).

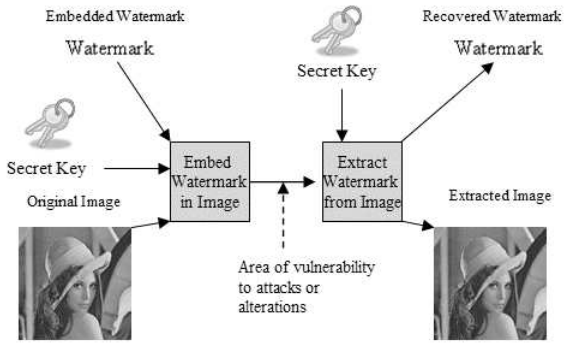

Figure 1. The watermarking method used

There are two main watermarking techniques available: spatial domain and frequency domain. The technique used in this paper will embed the watermark using the Discrete Wavelet Transform (DWT), utilizing a time-scale method.

D. Requirements

An implicit requirement for a blind watermark is that it is invisible to the naked eye and should look indistinguishable from the original. There are also other requirements for successful watermark techniques. Literature lists the following common requirements: robustness, imperceptible to statistical methods, capacity, recovery with or without the original data, extraction or verification of a given watermark, speed, and security issues and use of keys.

- **Robustness:** A watermark should withstand any alteration of the watermarked image, whether it is unintentional or malicious. This is a key requirement. We have yet to find a watermarking method that is 100% robust. When a watermark is extremely robust it is inclined to be visible.
- **Imperceptible to statistical methods:** The presence of a watermark should not be identified easily by viewing the properties about the image.
- **Capacity:** A watermark needs to contain a sensible amount of identifying information. If it is not differentiable, it is often useless.
- **Recovery with or without the original data:** For most watermarking applications it makes sense not to require the original host image to extract the watermark. However, if this image is obtainable, and it will increase the robustness and imperceptibility of the watermark, it may make sense to use it.
- **Extraction or verification of a given watermark:** No matter how robust or imperceptible a watermark

may be, if it cannot be successfully extracted it is useless.

- **Speed:** It may be important that a key can be embedded and/or extracted very rapidly (such while copying an image). Often the DWT methods do not excel in this area.
- **Security issues and use of keys:** As in all aspects of cryptology, the keys length weighed against its security, and how difficult it is to crack, are often at odds. The usefulness of a simplistic key is minimal.

[1] [2] [4] [7].

E. Scoring and Evaluation

What methods can be used to score and evaluate each of these requirements? The ideal would be to use real human vision: to gather a large group of people to view the original host image, the watermark, the host image containing the embedded watermark, and the extracted watermark under quality conditions (good lighting, quiet environment, no distractions, etc.). This is because evaluation of the watermark involves the subjective judgment of the distortion introduced through the process. Then an informed determination can be made in the trade-off between watermark robustness, whether the watermark is apparent in the image and the size of the watermark [8].

To measure results, we utilize the Peak Signal-to-Noise Ratio. The quality of an $N \times M$ host image ($f(x,y)$) is compared to the $N \times M$ image containing the watermark ($g(x,y)$) using the formula below.

$$\mathrm{PSNR}(f,g) = 20 \times log_{10} \frac{\text{max pixel value}}{\sqrt{\frac{\sum_{x,y}(f(x,y)-g(x,y))^2}{size}}}$$

III. WAVELETS

The word wavelet literally means “little wave”. It is attributed to Morlet and Grossmann in the early 1980s. They used the word ondelette; the French word was Anglicized by translating “onde” into “wave”, giving “wavelet”.

Various fields including mathematics, quantum physics, electrical engineering, and seismic geology developed the concept of wavelets independently. The development of many wavelet applications for digital signal analysis, such as image compression and edge detection [9] [10], ocean engineering, turbulence in hurricane winds, human vision, radar and earthquake prediction, is a result of these multiple disciplines working together.

The transform of a signal is just another way of representing the signal. It does not change the information content present in the signal. For example, using the Fourier transform, a signal can be expressed as the sum of a series of sines and cosines. In many ways, the wavelet transform is similar to the Fourier transform. Like the Fourier transform, wavelet transforms satisfy specified mathematical criteria to represent the data input. However, while the Fourier transform uses sinusoids to analyze signals, the wavelet transform uses wavelets of finite energy. Therefore, wavelets have their energy concentrated in time and are well suited for the analysis of transient, time-varying signals. Since most of the real-life signals encountered are time varying in nature, the Wavelet Transform fits many applications very well [11].

The introduction of the Discrete Wavelet Transform (DWT) brought ground-breaking influence in the area of signal analysis. Mallat, a pioneer in the field, established the idea that the wavelet transform, performed in a multi-scale manner, is effective for analyzing the meaning of the content in images.

For every wavelet, there is a function ψ called the mother wavelet described by

$$\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k), \qquad j,k = 0, \pm 1, \dots \ .$$

For any mother $\psi(x)$ there is a function $\phi(x)$ called the scaling function. Its dilations and translations are denoted by $\phi_{j,k}$. For example, the Haar mother wavelet ψ is pictured in the figure 2.

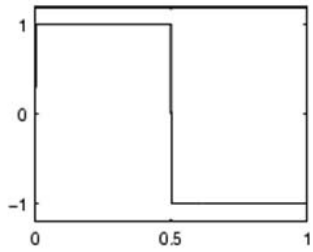

Figure 2. The Haar Wavelet

We use the DWT to implement an elegant watermarking scheme. The 2-D DWT decomposes the image into sub-images, three details and one approximation, as shown in figure 3. The upper left hand quadrant is the approximation image, which looks just like the original image of a filopodia (a snail neurite); only on 1/4 the scale. The three details are seen the other three quadrants.

The 2-Dimensional wavelet transform can be done in a separable fashion, meaning that we can use a 1-Dimensional DWT and apply it horizontally, then vertically. To accomplish the 1-D DWT, we split a 1-D input signal into two streams, and filter each. We have a low-pass filter (h) and a high-pass filter (g), corresponding to the scaling and wavelet functions, respectively. Filtering is a basic operation in digital signal processing where the input signal (i.e. a row or column from an image) is convolved with a set of coefficients. Convolution is described by the equation:

$$output[n] = \sum_{m=0}^{M-1} input[n-m] \times coefficient[m].$$

As the reader will notice, this can be computed with multiplication and additions (summation). The DWT separates an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components. For example, HL

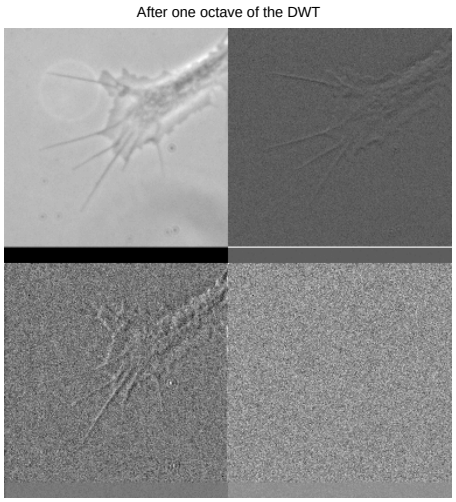

Figure 3. One level of decomposition using the DWT for Filopodia

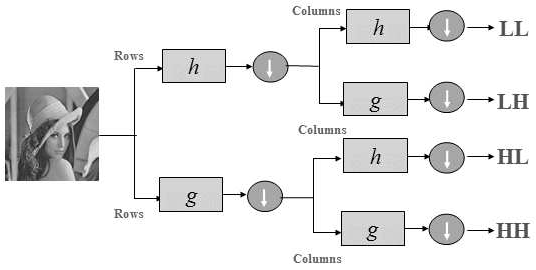

Figure 4. 2-D DWT decomposition or analysis tree

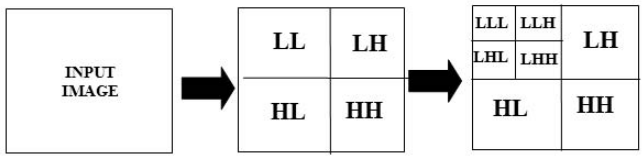

Figure 5. Analysis or decomposition hierarchy for 2-D DWT

means that we used a high-pass filter along the rows, and a low-pass filter along the columns. Figure 4 illustrates this concept. In figure 4 the high pass filter is denoted by *g* while the low pass filter is denoted by *h*. These details are the upper right hand quadrant, and both lower quadrants as shown in the figure 3. The low-pass and high-pass filters of the wavelet transform naturally break a signal into similar (low-pass) and discontinuous/rapidly-changing (high-pass) sub-signals.

The slow changing aspects of a signal are preserved in the channel with the low-pass filter and the quickly changing parts are kept in the high-pass filter's channel. Therefore we can embed high energy watermarks in the regions that human vision is less sensitive to, such as the high resolution detail bands (LH, HL, and HH). Embedding watermarks in these regions allows us to increase the robustness of our watermark, at little to no additional impact on image quality [12].

For a 2-D transform, we can filter along the rows, producing two sub-images each about half the size of the original. The heights are the same as the original, but the sub-images have half the width. We then filter these sub-images with low and high-pass filters along the columns. This produces two more sub-images each, for a total of four sub-images. This process is called decomposition or analysis. We label the resulting sub-images from an octave of the DWT as LL (the approximation), LH (horizontal details), HL (vertical details), and HH (diagonal details), according to the filters used to generate the sub-image.

Why not use this approximation as an image and recursively apply the DWT a second or third time? Multi-resolution is the process of taking one octave's LL output and putting this sub-image through another set of analysis filters. We can iterate this with LL again and again. The details at each succeeding octave are one-fourth the size of the previous octave. See figure 5.

The fact that the DWT is a multi-scale analysis can be used to the watermarking algorithm's benefit. The first approximation will be used as a "seed" image and recursively apply the DWT a second and third time (or however many times it is necessary to perform to find all of the areas of interest) [11].

As shown, the DWT can be used to analyze, or decompose, signals and images. On the flip side, since none of these pieces are lost, these components can be assembled back into the original signal without loss of information. This process is called reconstruction, or synthesis. The finite impulse response filters used are related to each other such that their coefficients satisfy perfect reconstruction criteria. That is, these filters effectively cancel aliasing, and no scaling is needed. This is important to satisfy the criteria of embedding a watermark, then having the capability to (almost) perfectly reconstruct the original image, except in the locations the watermark is inserted. Therefore, the signal can be reconstructed by undoing the transform. The approximation looks a lot like the original. When we add in the high frequency content, we get back to where we started (filling in the details). This mathematical operation is called the inverse discrete wavelet transform (IDWT). This manipulation is shown in figure 6, for one octave only.

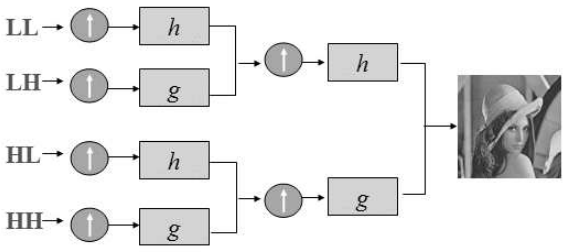

Figure 6. 2-D DWT synthesis or reconstruction tree

When using the discrete wavelet transform, there are many wavelets to choose from. We change the wavelet simply by changing the filter coefficients. The focus of this paper is to discover if there is a best way to choose a wavelet family when using the DWT for watermarking.

See [13] for more background on wavelets, and [14] for wavelet history.

IV. Implementation

A. Watermarks Used

Three distinct watermarks were chosen to embed in each of the images in the image database. The aim in employing different watermarks is to discover if there may be any dependence upon the choice of the watermark, or how the watermark is constructed. This is very different than the idea of incorporating two watermarks in an image to improve the protection and robustness for the image presented in [15], as only one watermark is embedded in any iteration in any image in the image database for this methodology.

The first watermark was a black and white watermark. It was created by cutting and pasting the word "Watermark" from a Microsoft Word® document into Adobe Photoshop®. This exhibits the use of text, such as the owner's name, as a watermark image. It will be referred to as the "text watermark". This watermark is 80 x 20 pixels. The larger a watermark is, the more likely it will become visible.

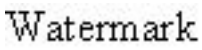

Figure 7. Text watermark bitmap embedded

The second watermark demonstrates the use of color. The letters "ABC" are presented in the primary colors red, yellow and blue respectively. This watermark was created by using Microsoft Word®, and changing the font color. Then the watermark was pasted into Adobe Photoshop® as well to create the colorful bitmap. It will be referred to as the "color watermark". This bitmap is 40 x 20 pixels.

Figure 8. Colored watermark bitmap that contains a red "A", yellow "B" and blue "C"

The third watermark illustrates the use of a graphic as the marking. It is a greyscale happy face. It was created by scanning in a manually drawn picture, then reducing it in size. It will be referred to as the "happy face watermark". The happy face is 50 x 36 pixels.

Since there are many choices of wavelets, we naturally ask the question, "Is there a particular wavelet that is better suited for watermarking applications?" Since a unique answer to this question was not discovered in previous research, eight differing popular wavelets were used to implement for comparison and contrast; the Haar wavelet, the orthogonal, 4-coefficient, Daubechies wavelet (e.g. db2), the 32nd order Daubechies wavelet (e.g. db32), three biorthogonal wavelets, including a reverse biorthogonal wavelet (bior2.2, bior5.5, rbio6.8), the symlet 8-coefficient wavelet and the 4th order Coiflet wavelet. MATLAB has built in support for numerous wavelets, and we adopt MATLAB's wavelet names here [16].

Figure 9. Graphical happy face greyscale watermark bitmap embedded

B. Images Used

Varying sizes of images were chosen. The image database also contains greyscale, black and white and color images. The watermarking algorithm for the color images in this research is not unique to the color images: i.e. the same operation was used for the black and white and greyscale images and for embedding the color watermark. Other specific proposals that differentiate the algorithm based on color include [17]. As well, the images differ in complexity and type. Some of the images are natural scenes, others are manufactured. As well, there are various formats including .jpg, .gif and .tif formats.

The image database consists of:

- "Barbara" - 512 x 512 pixels
- "Bird" - 256 x 256 pixels (available at [18])
- "Boat" - 512 x 512 pixels [18]
- "Box" - 256 x 256 pixels
- "Cameraman" - 256 x 256 pixels (available at [18])
- "Cell" - 190 x 158 pixels
- "Circles" - 256 x 256 pixels (available at [18])
- "Circuit" - 272 x 280 pixels
- "Colorful Butterfly" - 768 x 512 pixels (available at [18])
- "Colorful Geometric Frog" - 1120 x 1112 pixels (available at [18])
- "Colorful Sunny Flowers" - 814 x 880 pixels (available at [18])
- "Colorful Tulips" - 768 x 512 pixels (available at [18])
- "Colorful Windsails" - 768 x 512 pixels (available at [18])
- "Crosses" - 256 x 256 pixels (available at [18])
- "Dog on Porch" - 256 x 256 pixels [11]
- "Eight Coins" - 242 x 308 pixels [16].
- "Filopodia" - 640 x 480 pixels (special thanks to Dr. Vincent Rehder at Georgia State University)
- "France" - 672 x 496 pixels (a converted PowerPoint slide [18])
- "Frog" - 620 x 498 pixels [18]
- "Gold Hill, An Alpine Village" - 512 x 512 pixels [18]

- "Horizontal Lines" - 256 x 256 pixels [18]
- "Jumping Shark" - 534 x 301 pixels (special thanks to ©kemmcnair.com)
- "Lena" - 512 x 512 pixels
- "Lena in Color" - 512 x 512 pixels [18]
- "Library" - 464 x 352 [18]
- "Mandrill" - 512 x 512 pixels [18]
- "Montage" - 256 x 256 pixels [18]
- "Moon" - 358 x 536 pixels
- "Mountain" - 640 x 580 pixels [18]
- "MRI" - 128 x 128 pixels
- "Peppers" - 512 x 512 pixels [18]
- "Peppers in Color" - 512 x 512 pixels [18]
- "Pout" - 240 x 290 pixels
- "Pretty Calico Kitty" - 640 x 480 pixels (the author's cat)
- "Satellite View of Washington" - 512 x 512 pixels [18]
- "Serrano Artwork" - 630 x 794 pixels [18]
- "Slope" - 256 x 256 pixels [18]
- "Squares" - 256 x 256 pixels [18]
- "Text" - 256 x 256 pixels [18]
- "Zelda" - 512 x 512 pixels [18]

Sample images include figures 10, 11, 12, 13, 14, 15 and 16.

Figure 10. Box

Figure 11. Eight Coins

Figure 12. Cameraman

Figure 13. Lena

C. Algorithm

The 2-D DWT is applied to the image, giving four quadrants 1/4 the size of the original image, and producing the two matrices of coefficients that will be manipulated, the horizontal details (HL) and vertical details (LH). A pseudo random noise pattern is generated using the secret key as a seed, and each of the bits of the watermark are embedded in the horizontal (HL) and vertical (LH) coefficient sub-bands using this pattern. The equation from [19] used to embed one of the three watermarks is:

$$W_i' = W_i + \alpha W_i x_i \quad \text{for all pixels in LH, HL}$$

Figure 14. Dog on porch

Figure 15. Colorful geometric frog

Figure 16. Pretty calico kitty

$$W_i' = W_i \quad \text{for all pixels in HH, LL.}$$

W_i' is the watermarked image, W_i is the original image, and α is a scaling factor. Increasing α increases the robustness of the watermark, but decreases the quality of the watermarked image. We use the same α (the constant 2 on a scale of 1 to 5) as used in [19]. Finally, we write the image, and calculate the PSNR. Inoue in [20] also puts forth the algorithm of classifying the coefficients in the decomposed image as significant or insignificant. Then watermark data is embedded in the location of insignificant coefficients. This is in contrast to the process presented in this paper in which the watermark is inserted in the portions of the image that are, in theory, least perceptible to the human eye.

All of the images in the image database are combined with the three unique watermarks and continue to the extraction step, detailed in the next paragraph. This is without the introduction of the simulation of interference. As well, each of the images that is embedded with the text watermark, the colored text watermark, or the "happy face" graphical watermark, will be altered to model image distortion (i.e., innocent or purposeful tampering). Three types of noise were individually applied in different iterations of the process:

- Gaussian white noise with a zero mean noise and 0.01 variance.
- Salt and pepper noise with a noise density of 0.04.
- Speckle noise with a mean of 0 and variance 0.04.

Another common action that occurs with images is compression. After one of the watermarks is embedded, the discrete cosine transform matrix of each 8-by-8 block of the image is computed. The matrix of pixels for the image is divided into 8-by-8 sections, starting in the upper left corner, with no overlap. Zero padding is applied if the 8-by-8 blocks does not fit exactly over the image. A mask is applied so that only 16 % (10 out of 64 for each 8-by-8 block) of the pixels are retained from the original image. The inverse of the discrete cosine transform is performed to recover the now condensed image. This is similar to the method used in JPEG compression.

See section V for the original image compared to the visuals that illustrate the some of the operations of applying noise or compression to the "Text" image after the text watermark has been embedded.

To extract the watermark, we apply the 2-D inverse DWT to the possibly corrupted watermarked image W_i^*. The same secret key, which is obviously known to the owner of the content but not to anyone who wishes to access the content illegally, is used to seed the random function and to generate the pseudo random noise pattern. The more complex the key is, the less likelihood there is of tampering. The correlation, z, between the pseudo random noise and the horizontal and vertical details is found, and if that correlation exceeds a threshold (the mean of the correlation), a pixel in the watermark is located. y_i is a candidate pixel of the watermark and M

is the length of the watermark. The equation below comes from [19].

$$z = \frac{1}{M} \sum_{1,M} W_i^* \times y_i$$

Finally the extracted watermark is written, the multiple PSNRs are calculated and a file is written with all of the results. There are two PSNR calculations performed. The first is between PSNR between the noisy or compressed image with the watermark embedded and the original image. The second PSNR is between the recovered watermark and the original watermark.

The algorithm is the following:

1. Iterate through each watermark.
   - Black and white text watermark.
   - Red "A", yellow "B" and blue "C" colored text watermark.
   - Happy face graphical watermark.

2. For each watermark iterate over each type of transformation.
   - No manipulation
   - Gaussian noise
   - Salt and pepper noise
   - Speckle noise
   - Compression

3. For each watermark and each transformation, iterate over each wavelet family.
   - Orthogonal 4 coefficient Daubechies
   - 32nd order Daubechies
   - Three biorthogonal wavelets including a reverse biorthogonal type
   - Symlet 8 coefficient
   - 4th order Coiflet

4. For each image in the database
   - Embed the watermark.
   - Calculate the PSNR for the original image and the image with the watermark embedded.
   - Add noise or compression.
   - Calculate the PSNR for the original image and the manipulated image with the watermark embedded.
   - Extract the watermark.
   - Calculate the PSNR for the watermark and the extracted watermark.

5. Write the data
   - Write all created images with the watermark embedded including the images (noisy, compressed or no action) and Recovered watermarks
   - Write PSNR data.

V. RESULTS

Forty pristine original images, with eight wavelet families, including no tampering or three types of noise, or compression introduced results in 1600 watermarked images and 1600 recovered watermarks. The process was performed utilizing three distinct watermarks, so consequently 4800 watermarked images and 4800 recovered watermarks were produced. The outcome is almost 10,000 images, which are too many to reproduce in this forum. Various pictorial portions will be shown in this section as a representative sample. Please see the tables for more specific statistics.

First, one of the watermarks is embedded in the original image. No noise is introduced into the image. To simulate effects of such innocent problems such as transmission errors, or perhaps alterations of the image for other more predatory reasons, three types of noise are then independently applied to the image to simulate image corruption. The Gaussian white noise added had a zero mean noise with 0.01 variance. The salt and pepper noise had noise density of 0.04, affecting approximately 4% of the pixels. Lastly, the speckle added multiplicative noise that is uniformly distributed random noise with mean 0 and variance 0.04. Another type of alteration which may be applied to the image is compression. Then the watermark is extracted from the noisy image. The results are shown in the tables.

When embedding any of the three watermarks, the Text image was consistently one of the worst performing images. This can be seen in some of the multiple variations that are shown in the following figures. For comparison sake, the Text image is shown with speckle noise alone, and no watermark embedded. The Text image repeatedly produced results that gave the lowest PSNR. We chose to show these as an example so that not just the images that excel can be viewed by the reader.

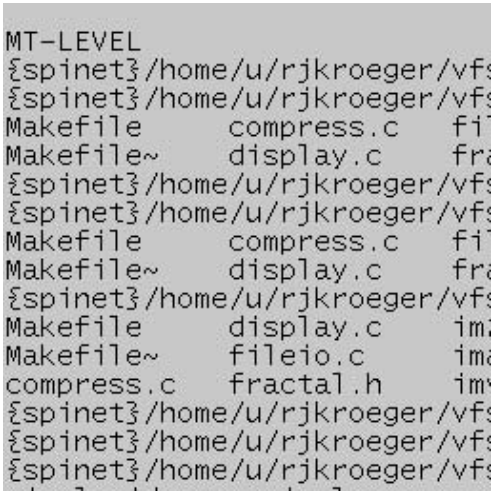

Figure 17. The original image - Text

Visually it is evidenced, and supported by the PSNR measurements as well, that the Haar wavelet (even when simulating an attack with noise) outperforms the other more complex wavelets. The Biorthogonal 5.5 wavelet is shown in the figure 21 as indicative of this pattern, but any of the other seven wavelets could have been pictured and the same results would be supported.

Figure 18. The original Text image without a watermark embedded with speckle noise - PSNR 16.062

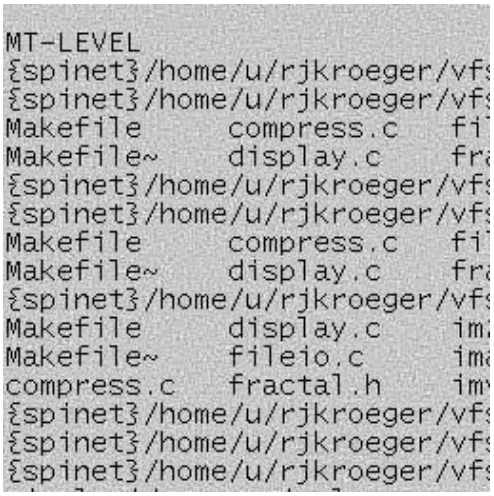

Figure 19. The Text image embedded with the text watermark using the Haar wavelet with no noise added - PSNR 25.40

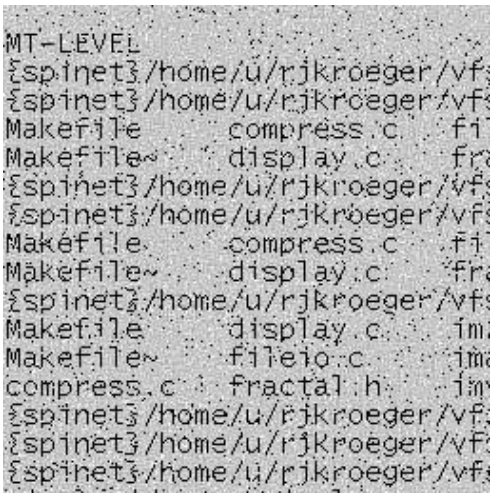

Figure 20. The Text image embedded with the text watermark using the Haar wavelet with salt and pepper noise added - PSNR 18.4324

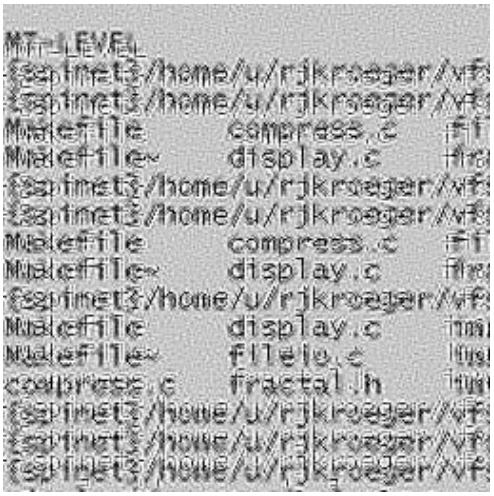

Figure 21. The Text image embedded with the text watermark using the biorthogonal 5.5 wavelet with no noise added - PSNR 12.49

Figure 22. The Text image embedded with the text watermark using the Coiflets 4 wavelet with compression - PSNR 14.74

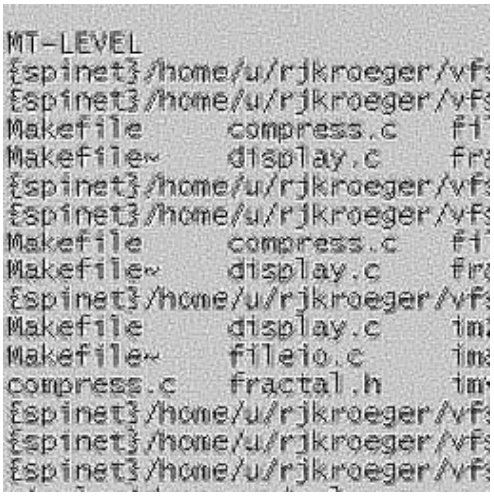

Figure 23. The Text image embedded with the color text watermark using the Daubechies 4 coefficient wavelet with no noise added - PSNR 11.53

Figure 24. The Text image embedded with the color text watermark using the Daubechies 4 coefficient wavelet with speckle noise added - PSNR 16.65

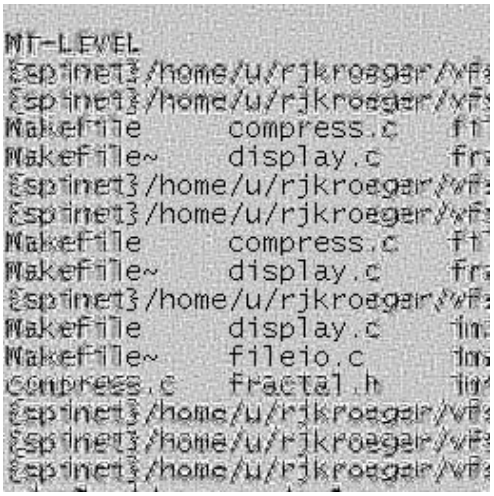

Figure 25. The Text image embedded with the happy face watermark using the biorthogonal 2.2 wavelet with no noise added - PSNR 14.30

The following images (figures 26, 27, 28, 29, and 30) are the results of the extraction of the various watermarks when the project is implemented. For brevity, the watermarked images have been resized from their original dimensions and not all of the recovered watermarks are pictured. The text watermarks in figure 26 are all from the Lena image, encoded from left to right with the Haar wavelet, Daubechies 4 coefficient wavelet, and the biorthogonal 2.2 wavelet. In figure 27, we see the recovered text watermarks from the Frog image [18], with Gaussian noise, and encoded from left to right with the Haar wavelet, the biorthogonal 5.5, and the symlet 8 coefficient wavelet. In figure 28 the text watermark image is shown after being extracted from the Filopodia image. The Filopodia image was compressed to less than 16 percent of its original size after the watermark was embedded. These three text watermarks used the Haar, reverse biorthogonal, and symlets wavelet respectively. Next, figure 29 details the recovery of the color watermark from the Alpine Village image using the Haar, Daubechies 4 coefficient and Coiflet wavelets. Lastly figure 30 illustrates the happy face watermark after it has been recovered from the Mountain image that has been doused with salt and pepper noise. The wavelets employed for this procedure were the Haar, Coiflet and Daubechies 32. Clearly, the watermarks encoded with the Haar wavelet were the best ones recovered. These recovered watermarks are representative; each image in the database had corresponding visual results.

Figure 26. Text watermark images recovered from Lena: Haar, Daubechies 4 coefficient, and biorthogonal 2.2

Figure 27. Text watermark images recovered from Frog (with Gaussian noise): Haar, biorthogonal 5.5, and symlets 8

Figure 28. Text watermark images recovered from the compressed Filopodia: Haar, reverse biorthogonal, and symlets 8

Figure 29. Colored watermark images recovered from Alpine Village: Haar, Daubechies 4 coefficient, and Coiflet

Figure 30. Happy face watermark images recovered from the Mountain (with salt and pepper noise): Haar, Coiflet 4, and Daubechies 32

The tables shown represent some of the statistics obtained. The tables concentrate on the numbers reported for the watermark extraction; this notes which wavelet families allow the extracted object to be identified.

Table I examines the watermark embedding process, using the text watermark, with no noise added, showing the average, minimum and maximum PSNRs. All of the data for the other possible combinations for embedding each of the watermarks (such as embedding the happy face watermark using salt and pepper) are available. For a small portion of the images in the image database, table II shows the PSNR for each wavelet for the text watermark

TABLE I.
TEXT WATERMARK EMBEDDING WITH NO NOISE

Wavelet	Average PSNR	Minimum PSNR	Maximum PSNR
Haar	25.7233	25.1458	28.1656
Daubechies	22.1884	11.5328	27.6467
Daubechies 32	23.4181	13.3125	28.0125
Bior 2.2	23.1592	14.3529	27.8656
Bior 5.5	22.6510	12.4863	27.4846
Symlets 8	22.8566	12.7960	27.6503
Coiflets 4	22.9280	12.8687	27.706
Rev. Bior 6.8	22.9505	12.9984	27.6137
All	23.2343	11.5328	28.1656

TABLE II.
SAMPLE DATA FOR TEXT WATERMARK EXTRACTION

Wavelet	Image	PSNR	Image	PSNR
Haar	Barbara	6.11	Frog	6.11
Daubechies		5.82		5.81
Daubechies 32		5.94		5.94
Bior 2.2		6.02		6.04
Bior 5.5		5.92		5.92
Symlets 8		5.93		5.93
Coiflets 4		5.93		5.94
Rev. Bior 6.8		5.94		5.95
Haar	Box	6.10	Lena	6.11
Daubechies		5.87		5.80
Daubechies 32		5.94		5.94
Bior 2.2		5.98		6.03
Bior 5.5		5.93		5.92
Symlets 8		5.93		5.93
Coiflets 4		5.93		5.94
Rev. Bior 6.8		5.94		5.95
Haar	Camera-	6.06	Mandrill	6.11
Daubechies	man	5.86		5.82
Daubechies 32		5.94		5.94
Bior 2.2		5.98		6.02
Bior 5.5		5.93		5.92
Symlets 8		5.94		5.93
Coiflets 4		5.94		5.94
Rev. Bior 6.8		5.94		5.94
Haar	Cell	6.01	Moon	6.11
Daubechies		5.88		5.82
Daubechies 32		5.94		5.93
Bior 2.2		5.97		6.04
Bior 5.5		5.93		5.91
Symlets 8		5.94		5.93
Coiflets 4		5.94		5.93
Rev. Bior 6.8		5.94		5.94

recovery process. Table III shows a summary for all of the images using the each of the watermarks and no noise. Table IV shows, for the entire image database, the average PSNR obtained for each type of noise, including Gaussian, salt and pepper, and speckle using the text watermark. Tables V and VI show the same thing, but for the happy face and color watermarks respectively. The last table shows the results of compression using the text watermark.

When embedding the watermark, the higher the PSNR value is, the less perceptible the embedded watermark will be to the human eye. As well, the higher the PSNR is for the recovered watermark, the easier it is to identify.

VI. CONCLUSIONS

The Discrete Wavelet Transform enables the embedding of a watermark at higher level frequencies, which

TABLE III.
WATERMARK EXTRACTION WITH NO NOISE

Wavelet	Text Watermark Average PSNR	Color Watermark Average PSNR	Happy Face Watermark Average PSNR
Haar	6.09	5.41	6.67
Daubechies	5.83	5.39	6.42
Daubechies 32	5.94	5.40	6.52
Bior 2.2	6.01	5.40	6.59
Bior 5.5	5.92	5.40	6.51
Symlets 8	5.93	5.40	6.52
Coiflets 4	5.93	5.40	6.52
Rev. Bior 6.8	5.94	5.40	6.53
All	5.95	5.40	6.535

TABLE IV.
TEXT WATERMARK EXTRACTION WITH NOISE

Wavelet	Speckle Average PSNR	Gaussian Average PSNR	Salt and Pepper Average PSNR
Haar	6.07	6.06	6.06
Daubechies	5.87	5.88	5.88
Daubechies 32	5.94	5.94	5.94
Bior 2.2	5.99	5.98	5.98
Bior 5.5	5.93	5.93	5.93
Symlets 8	5.93	5.93	5.93
Coiflets 4	5.94	5.94	5.94
Rev. Bior 6.8	5.94	5.94	5.94

TABLE V.
GRAPHICAL HAPPY FACE WATERMARK EXTRACTION WITH NOISE

Wavelet	Speckle Average PSNR	Gaussian Average PSNR	Salt and Pepper Average PSNR
Haar	6.66	6.65	6.65
Daubechies	6.45	6.46	6.47
Daubechies 32	6.53	6.52	6.52
Bior 2.2	6.58	6.57	6.57
Bior 5.5	6.52	6.52	6.52
Symlets 8	6.52	6.52	6.52
Coiflets 4	6.52	6.52	6.52
Rev. Bior 6.8	6.52	6.53	6.52

TABLE VI.
COLOR TEXT WATERMARK EXTRACTION WITH NOISE

Wavelet	Speckle Average PSNR	Gaussian Average PSNR	Salt and Pepper Average PSNR
Haar	5.41	5.40	5.41
Daubechies	5.39	5.39	5.39
Daubechies 32	5.39	5.39	5.40
Bior 2.2	5.40	5.40	5.40
Bior 5.5	5.40	5.39	5.40
Symlets 8	5.39	5.39	5.40
Coiflets 4	5.39	5.39	5.40
Rev. Bior 6.8	5.39	5.39	5.40

TABLE VII.
TEXT WATERMARK EXTRACTION WITH COMPRESSION

Wavelet	Average PSNR
Haar	5.98
Daubechies	5.94
Daubechies 32	5.94
Bior 2.2	5.96
Bior 5.5	5.94
Symlets 8	5.94
Coiflets 4	5.94
Rev. Bior 6.8	5.94
All	5.9475

are not as visible to the human eye, via the access to the wavelet coefficients in the HL and LH detail sub-bands. For both the impact on the original image and for the recovery of the embedded watermark, the Haar wavelet, both visually and objectively measured by PSNR, outperforms the other families tested when using greyscale or black and white images and watermarks. However, when color is introduced, such as with the color watermark and the color images, the preference demonstrated is smaller (and in a few cases nonexistent). This appears to be a byproduct not of the Haar wavelet's capabilities itself, but of using the same algorithm for the color, greyscale and black and white images. These statistics suggest further work for the impact of color. This is an intriguing result as the Haar wavelet, which is discontinuous, and resembles a step function, conveys the most simplicity of all of the wavelet families.

This remains true when three types of noise are added, including Gaussian, speckled and salt and pepper, as well as when the watermarked image remains uncorrupted. Compression at such a high rate (retaining only 16% of the image) seems to negate the Haar wavelet's benefit. Furthermore, the introduction of color also has a large influence in equalizing the wavelet's capabilities, although the Haar wavelet has a small superiority in these cases too. In almost every other situation the Haar wavelet repeatedly outperforms the others. Therefore, the size, type and complexity of the image, and the introduction of noise does not seem to change the advantage that the simple, but effective, Haar Wavelet displays when working with greyscale watermarks.

As well the visual results are more stunning than is reflected by the PSNRs. When viewing the recovered watermarks for each of the images, even for the color watermark, the image recovered when using the Haar wavelet shines. Therefore, the need for a more exact measurement is suggested.

REFERENCES

[1] F. A. P. Petitcolas, R. Anderson, and M. G. Kuhn, "Information hiding - A survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[3] D. Kirovski and F. A. P. Petitcolas, "Blind Pattern Matching Attack on Watermarking Systems," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1045–1053, 2003.

[4] F. A. P. Petitcolas, "Watermarking Schemes Evaluation," *IEEE Signal Processing Magazine*, vol. 17, pp. 58–64, September 2000.

[5] F. Hartung and M. Kutter, *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2000, ch. Chapter 5 - Introduction to watermarking techniques.

[6] J. Weiner and K. Mirkes, *Watermarking*. Inst. Paper Chemistry, Appleton, WI, 1972.

[7] J. Seitz, *Digital Watermarking for Digital Media*. Information Science Publishing, 2005.

[8] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding : Steganography and Watermarking - Attacks and Countermeasures (Advances in Information Security, Volume 1) (Advances in Information Security)*. Kluwer Academic Publishers, Norwell, MA, 2006.

[9] E. Brannock, M. Weeks, and V. Rehder, "Detecting filopodia with wavelets," in *Proceedings of the 2004 International Symposium on Circuits and Systems*, May 2006.

[10] E. Brannock and M. Weeks, "Edge detection using wavelets," in *Proceedings of the 44th ACM SE 2006*, March 2006, pp. 649–654.

[11] M. Weeks, *Digital Signal Processing Using MATLAB and Wavelets*. Infinity Science Press, 2007.

[12] G. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking Digital Image and Video Data," *IEEE Signal Processing Magazine*, vol. 17, pp. 20–43, September 2000.

[13] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[14] Y. M. Stephane Jaffard and R. D. Ryan, *Wavelets Tools for Science and Technology*. Society for Industrial and Applied Mathematics (SIAM), 2001.

[15] G. Zhu, "Research on an advanced novel watermarking techology with higher robustness," in *2008 International Symposium on Electronic Commerce and Security, 2008*, August 2008, pp. 946–950.

[16] *Image Processing Toolbox User's Guide*. Natick, MA: The MathWorks, Inc, 2004.

[17] G. Sun and Y. Yu, "DWT Based Watermarking Algorithm of Color Images," in *Industrial Electronics and Applications, 2007, ICIEA 2007*, May 2007, pp. 1823–1826.

[18] E. R. Vrscay, F. Mendivil, H. Kunze, D. L. Torre, S. Alexander, and B. Forte, "Waterloo Repertoire GreySet (1 and 2) and Waterloo Repertoire ColorSet (1 and 2)," http://links.uwaterloo.ca/.

[19] A. M. H. Inoue and T. Katsura, "An image watermarking method based on the wavelet transform," *ICIP 99. Proceedings. 1999 International Conference on Image Processing*, vol. 1, pp. 296–300, October 1999.

[20] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A digital watermark based on the wavelet transform and its robustness on image compression," in *Proceedings of the IEEE International Conference on Image Processing, ICIP '98*, 1998, pp. 391–395.

Evelyn Brannock is currently a Ph.D. candidate at Georgia State University, in Atlanta, GA, USA. She received her M.S. degree at McNeese State University in Mathematics and her B.S. degree in Mathematics at Georgia Southern University. Her research interests include image processing and the Discrete Wavelet Transform.

Michael Weeks received a Bachelor of Engineering Science in 1993 and Master of Engineering in 1994 from the University of Louisville, both in Engineering Math and Computer Science. He next enrolled at the Center for Advanced Computer Studies at the University of Louisiana at Lafayette (formerly known as the University of Southwestern Louisiana), where he received a Master of Science degree in Computer Engineering in 1996, followed by a Ph.D. in Computer Engineering in 1998.

He was hired by the Computer Science Department at Georgia State University as part of the State of Georgia's Yamacraw program. He is currently an Associate Professor.

Robert W. Harrison received a B.S. degree in biophysics from Pennsylvania State University, University Park, in 1979 and the Ph.D. degree in molecular biochemistry biophysics from Yale University, New Haven, in 1985.

He is currently a full professor in the Department of Computer Science at Georgia State University with a secondary appointment in the Biology department. His active and diverse research interests span many areas of bioinformatics and computational biology, and he has published over a hundred papers in these areas. His current research interests include computational approaches to the prediction and design of molecular structure, machine learning, and the development of grammar based models for systems and structural biology.

Dr. Harrison is a Georgia Cancer Coalition Distinguished Scholar, and his search has been supported by the National Institutes of Health and the Georgia Cancer Coalition