DWT IMAGE ENCODING AND MESSAGE CORRECTIONS ON PRINTED IMAGES

By

Francis Roxas

Senior Thesis in Electrical Engineering

University of Illinois at Urbana-Champaign

Advisor: Pierre Moulin

May 2019

Abstract

As more information gets stored digitally, intellectual property (IP) holders needed a way to protect their content from being reused without their permission. One way that IP holders protect their content is through the use of watermarks which are pieces of information that are embedded in the IP holder’s content. This thesis explores the use and effectiveness of watermarking techniques on printed images. Specifically, this thesis explores techniques to embed printed images with a watermark and to retrieve those same watermarks from photographs of those images. The watermarks found in these images suffer from various different degradation effects which may compromise the message that is being transferred in the paper. To alleviate these degradations, the messages will be encoded into the images using Error Correction Codes that will help the user retrieve some of the information that would be lost as a result of these effects. Experiments are performed using a two-dimensional Discrete Wavelet Transformation and various Error Correction Coding Techniques including repetition Error Correction codes, hamming encoding, and Reed Solomon encoding schemes. These experiments are performed on various logos.

Subject Keywords: Discrete Wavelet Transforms, Reed Solomon Codes

Acknowledgments

I would like to thank Professor Pierre Moulin for his guidance over the past two semesters.

Contents

1. Chapter 1: Introduction

1.1 Motivation and Problem Statement

The advent of digital technology has led to easy access to a tremendous amount of knowledge attainable and distributable on the internet. This, in turn, leads to many opportunities for people to make illegal copies of content they obtain online and share those copies across the internet. So, as the amount of information accessed and distributed online grows so does the need of IP owner to protect their digital data does which has led to the growing issue of copyright protection. Nowhere is this more necessary than in the case of company logos which if misappropriated can lead to severe damage to a company's reputation and, thus, their own revenue. In the case of digital logos, these logos can be downloaded as digital images which can be redistributed physically via printers and then reuploaded digitally as pictures taken from phones or cameras. Once photographed, the newly obtained image, which is a slightly modified version of the original image, can be used in a way that violates the copyright claims of the original IP holders. As a result, IP holders search for a method to protect what they own after it is printed and reuploaded to the internet.

The common technique that many content creators use to protect their digital visual media is through the use of watermarks. Digital image watermarking gives these IP holders a means of securely providing their logos online. This technique involves encoding a message that may or not be visible to the unaided eye into the original data with the intent of recovering that message at a later date. Typically, these messages are either secret messages that modify the original image in some capacity meant only to be read by the receiver for purposes such a validation of ownership. A common approach that many content creators use to protect their digital visual media is the use of watermarks which modify the images through the use of visual symbols such as letters and numbers. These techniques often ruin the visual integrity of an image which limit their use. Furthermore, degradation from printing creates multiple challenges for most digital watermarks which are outlined below.

The first challenge in handling the processing of images that have been digitally retrieved by a photograph in a camera is due to the image capturing device itself. When a photographic image is taken, the newly created digital image may seem visually similar to the original digital image. However, the newly created image differ from the original image will have multiple differences from the original image due to the that have been created by the settings of the image capturing device. These settings include different aperture sizes and different shutter speeds which can severely change the composition of the base image by changing color values in the original image. Another challenge in handling the processing of photographic images is due to environmental factors. When the image is captured, there may be slight orientation differences in visually similar images, there could be lighting differences induced by the environment, and there may be slight cropping of the original image. The differences are even present in images that have been taken by the same camera and by the same person.

In this thesis, we consider the problem of digitally transmitting a watermark using a photograph of a printed image. We will provide an approach to this problem with a technique based on working in the 2D Discrete Wavelet Transform domain which is an extension of the 1D Discrete Wavelet Transform.

2. Literature Review

2.1 Overview of the Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) uses a basis function known as a wavelet which starts at zero, oscillates for a little bit, and returns to zero to approximate the original function locally [3][5][6]. These wavelets are created from shifting and scaling a fixed function known as a mother wavelet [6]. This transformation can be extended to two dimensions to decompose two dimensional signals in three spatial directions which are the horizontal, vertical and diagonal dimensions [6]. Each of these directions is represented by a sub-matrix which contains coefficients that relate to the higher frequency details of the original image in that direction [3]. In addition, the DWT also returns an approximation matrix which contains wavelet coefficients that correspond to the lower frequencies of the base image [3]. This approximation image can be decomposed further in a pyramidal manner to get higher level DWT decompositions [6]. As the image is decomposed further, the magnitude of the coefficients increases which signifies that that coefficient corresponds to lower frequencies [6]. An example of the first level of the Discrete Wavelet Decomposition using the Haar Wavelet is shown below.

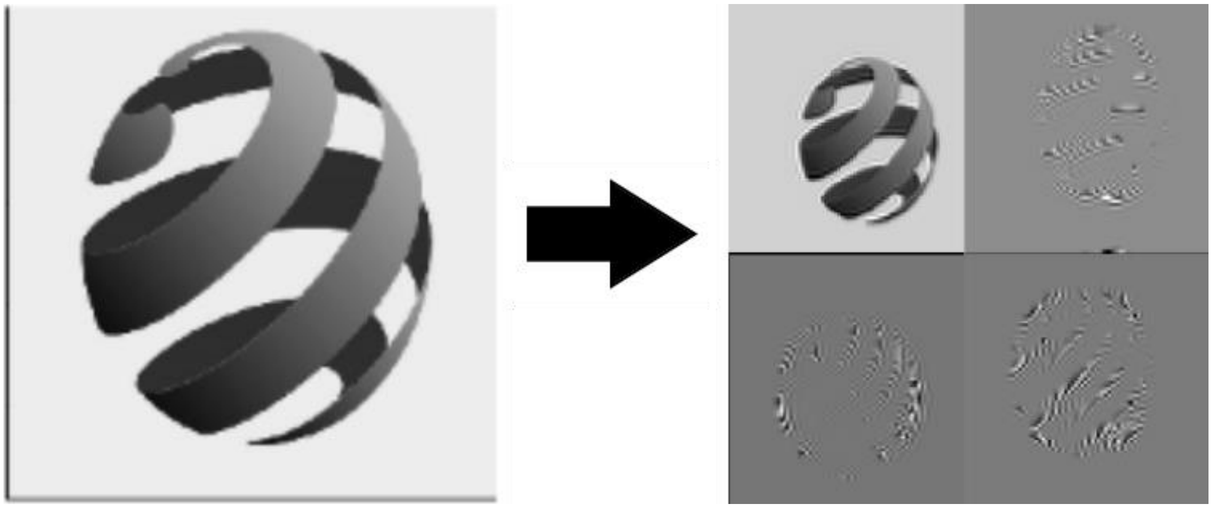

Figure 2.1 Example of first level discrete wavelet decomposition. The example image used for decomposition was found at [7].

Since each of the coefficients in the wavelet decomposition are shifted and scaled versions of the mother wavelet, the Discrete Wavelet Transform can be implemented in a straightforward manner by convolving a high pass filter along the vertical, horizontal or diagonal directions to get the coefficients along that correspond to each respective direction [6]. The approximation matrix can be found by applying a low pass filter to the image to capture the remaining frequencies [6].

2.2 Print and Scan Watermarking Models

Just as cameras can be used to redistribute images, the print scan process can be used to reproduce and redistribute images and from there the images can be used in ways that violate the original copyright of the images [4][9]. A typical way to protect the original IP holders copyright of the image is to embed the image with a watermark which is resilient to degradations due to the printing and scanning process [4]. These degradations lead to changes in the intensity of pixel colors found in the resultant image which are an effect of the luminance, contrast, gamma correction, blurring and cropping changes in the printing and rescanning process [10]. Over the past fifteen years, some research has gone into providing different models for copyright protection against print can attacks [4]. Solachidis and Pitas [10] introduce a method that embeds a watermark in the DFT domain in a ring structure. This method allows their model to encode a large amount of watermarked information into the image and in order to retrieve the watermark they implement a correlation measure [10]. Filiba [3] proposed a wavelet based watermarking technique to encode payloads into images. In this approach, Filiba decomposes the image to be watermarked into its third wavelet decomposition level [3]. Afterwards, the decomposed image can be divided into N different chunks of data which are made by splitting the concatenation of the Horizontal, Vertical, and Diagonal decomposition matrices into N non-overlapping groups [3]. Each chunk is then modified with a Code-Division Multiple Access (CDMA) technique [3]. This technique involves encoding the image in each chunk with either a 1-sequence or a 0-sequence known as an S1 sequence and a S0 sequence respectively [3]. To retrieve the original message, the correlation of the image is tested with the S1 and the S0 sequences [3]. If the correlation of the image with S1 is greater than the correlation of the image with S0, the message bit is decoded as a 1 and a 0 otherwise [3].

2.3 Error Correction Codes

In the discrete wavelet transform based image watermarking information in the original image which are contained in the spatial domain through pixel values are captured in the wavelet coefficients [12]. In the base image these pixel values are expressed as quantized signal values [12]. To create a watermark, the wavelet coefficients are modified in some way to embed information [12]. A problem arises after the coefficients after performing the inverse transform on the DWT coefficients and the pixel values are re-quantized because rarely can all of the modified coefficient values be expressed as quantized pixel values [12]. Thus, some of the embedded information becomes corrupted from this process. In addition, external factors such as lighting, blurring, and slight rotations can disturb the pixel values at each of the original locations which can further corrupt the message contained in the modified coefficients. To deal with these errors, the watermark in the image needs to incorporate redundant information in the DWT domain [12]. In this thesis, redundant information found in parity bits will be added to the original message through the use of Error Correction Code (ECC). The ECC used in this thesis are Repetition Codes, Hamming Codes, and Reed Solomon Codes.

2.3.1 Repetition Codes

A naïve solution to encoding redundant information into the source image is through Repetition Codes. Repetition Codes work by repeating the information found within the input sequences a set number of times [13]. For instance, if the rate of the output sequence were 1/3 we would have 3 bits for every 1 bit in the input sequence. This means that there will be 2 parity bits for each bit in the input sequence. Following is an example of a 10-bit input sequence extended to 30 bits using Repetition Codes.

Input Sequence: [0. 1. 1. 0. 0. 1. 0. 1. 1. 0.]

Output Sequence: [0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0.]

To decode a message encoded with a rate of 1/N, every sequential N-bits in the retrieved sequence are averaged together. If the average is closer to 1, then the bit associated with those N bits in the decoded sequence is read as a 1. Else, the bit is read as a 0. An example of the retrieved sequence and the decoded sequence are shown below.

Retrieved Sequence: [0. 0. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0.]

Decoded Sequence: [0. 1. 1. 0. 0. 0. 0. 0. 1. 0.]

2.3.2 Hamming Codes

Hamming Codes are a type of linear block error correction codes that were developed by Richard Hamming [16]. Hamming codes relate to Hamming distance, which is the amount of differences between the "check" bits and the information bits. In this thesis, we will be using the (7,4) binary Hamming code. In this code, any message of length equal to 7 bits satisfies the following three parity-check equations:

$$c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0 \ , \qquad (1)$$

$$c_2 \oplus c_3 \oplus c_6 \oplus c_7 = 0 \ , \qquad (2)$$

$$c_4 \oplus c_5 \oplus c_6 \oplus c_7 = 0 \ , \qquad (3)$$

where c_k is the kth bit of the 7-bit sequence [17]. The above equations are used to create the parity bits c_1, c_2, and c_4 for the input 4-bit sequence. Following is an example of a 4-bit sequence being extended to 7-bits.

Input Sequence: [0. 0. 1. 1.]

Output Sequence: [1. 0. 0. 0. 0. 1. 1.]

Upon receiving a 7-bit sequence, the Hamming code can correct up to 1 error within the sequence [16]. This is done by computing a vector called the syndrome. The syndrome s = [s_0, s_1, s_3] where s_0, s_1, s_2 are computed using the following parity-check equations:

$$s_0 = r_1 \oplus r_3 \oplus r_5 \oplus r_7 \ , \quad (4)$$

$$s_1 = r_2 \oplus r_3 \oplus r_6 \oplus r_7 \ , \quad (5)$$

$$s_2 = r_4 \oplus r_5 \oplus r_6 \oplus r_7 \ , \quad (6)$$

where r_k is the kth bit of the 7-bit received sequence [17]. The syndrome can then be used to determine and correct the location of the error bit by knowing that the bits s_0, s_1, s_2 returns the binary location of the error bit with s_2 being the most significant bit and s_0 being the least significant bit. An example of a retrieved sequence with an incorrect 4th bit is shown below.

Retrieved Sequence: [1. 0. 0. 1. 0. 1. 1.]

$s_0 = 0$, $s_1 = 0$, $s_2 = 1$ so error at 4th bit

Corrected Sequence: [1. 0. 0. 0. 0. 1. 1.]

Decoded Sequence: [0. 0. 1. 1.]

2.3.3 Reed Solomon Codes

Reed Solomon Codes are a type of BCH codes were codes originally developed in the 1960's to be used in satellite communications [3]. Due to the fact that it has very powerful error correction capabilities, Reed Solomon Codes have been used in more widespread communication technologies such as DVDs and barcode systems [3]. Reed Solomon Codes differ from the previous code due to the fact that unlike Repetition Codes, Reed Solomon Codes work at the byte level rather than the bit level [3]. This allows the Reed Solomon Code to correct multiple bit errors within any given byte of information that has been received [3].

To encode an input sequence, the input sequence is turned into a polynomial with coefficients from the finite field referred to as the Galois Field [3]. In the Galois Field, arithmetic operations such as addition and multiplication operations performed on elements in the field return elements in the field [14]. Afterwards, a generator polynomial which takes the general form of

$$g(x) = \ (x - a^y) \cdot (x - a^{y-1}) \ldots \cdot (x - a^1) \cdot (x - a^0), \quad (7)$$

is used to create a codeword where a^y, a^{y-1}, …, a^1, a^0 are the zeros of the polynomial [3]. Next, the input sequence polynomial is padded with y zeros and is divided by the generator polynomial to get a remainder polynomial [3]. This remainder is added to the input sequence polynomial to get the parity bites for the polynomial [3].

To decode the retrieved sequence, the retrieved sequence is again turned into a polynomial with coefficients from the Galois field and the polynomial is evaluated at $x = a^y, a^{y-1}, \ldots, a^1, a^0$ [3]. If the result is 0 for all the zeros of the embedded polynomial, then there is nothing to correct in the retrieved polynomial [3]. If plugging in one of the embedded polynomial's zeros does not return a zero in the retrieved polynomial, that zero location reveals which coefficient is incorrect in the resultant polynomial [3].

3. Experimental Setup

3.1 Proposed Method

The proposed method for embedding and recovering watermarks into photographic images involves randomly generating an N-bit payload message. After this, the message would be extended such that for any one bit of the original message, 2 more error correction bits were created. This would make the error correction rate equal to (1,3). The error correction codes that will be used in this thesis will be the Repetition Code, Hamming Code and the Reed Solomon Code.

The next step after this is to separate each of the cH3, cV3 and cD3 matrices into N non-overlapping sections and in each section a single location is chosen at random. Once we have all of the locations, we modify a group of 2x2 coefficients at each location based on its corresponding bit value in the payload. If the correspond bit to that location were a 1 the corresponding wavelet coefficients would be modified through the addition of a constant K. If the corresponding bit were a 0, the group of wavelet coefficients would be left unmodified. Afterwards, the modified cH, cV and cD matrices are combined with the original cA matrix and are transformed back into their original spatial domain representation through the inverse DWT.

This process is outlined in pseudo-code as follows:

```
FUNCTION encode (img, message, wavelet):
    cA1, cH1, cV1, cD1 <- dwt2(img, wavelet)
    cA2, cH2, cV2, cD2 <- dwt2(cA1, wavelet)
    cA3, cH3, cV3, cD3 <- dwt2(cA2, wavelet)
    segment_length <- floor( length(cH3) / length(message) )
    for each bit in message:
        if(bit == 1)
            cH3’ <- cH3 + K
            cV3’ <- cV3 + K
            cD3’ <- cD3 + K
        S1_cH3, S1_cV3, S1_cD3 <- concatenated_vector + K
        S0_cH3, S0_cV3, S0_cD3 <- concatenated_vector
    cA3 , cH3’, cV3’, cD3’ <- dwt2(cA2’, wavelet)
    cA2’, cH2’, cV2’, cD2’ <- dwt2(cA1’, wavelet)
    cA1’, cH1’, cV1’, cD1’ <- dwt2(img’, wavelet)
    return img’, S1_cH3, S1_cV3, S1_cD3, S0_cH3, S0_cV3, S0_cD3
```

Once the image has been embedded, the image must be captured. In this thesis, all images were printed using an HP LaserJet MFP M130 and captured using a Samsung Galaxy S5 with all image enhancements turned off. Once the image has been captured it must be realigned to match the original image. This was accomplished using OpenCV's SIFT modules to generate feature vectors between the digitally modified image and the digital photograph for realignment. Since, alignment is beyond the scope of this thesis, the alignment process will not be discussed much more in depth here.

To decode the image, we decompose the newly captured photographic image into it's third level wavelet decomposition. Then using the sections created earlier and the S1 and S0 sequences created during the encoding process, we calculate the cross correlation of each of the sections of the modified image with the S1 and S0 signals respectively. If the correlation between the photographic image and S1 is greater than the cross correlation between the photographic image and S0 for any given segment, the bit corresponding to that segment is determined to be a 1. It is chosen to be a 0 otherwise. Once, the bits have been retrieved from the photographic image, the error correction coding technique used for embedding is applied to the retrieved message to correct any bits that may have been modified in transmission.

The pseudo code for this process is as follows:

```
FUNCTION decode (img', msg_len, wavelet, ECC, S1_cH3, S1_cV3, S1_cD3, S0_cH3,
S0_cV3, S0_cD3):
        cA1', cH1', cV1', cD1' <- dwt2(img', wavelet)
        cA2', cH2', cV2', cD2' <- dwt2(cA1, wavelet)
        cA3', cH3', cV3', cD3' <- dwt2(cA2, wavelet)
        segment_length <- floor( length(cH3) / msg_len )
        for i in range message_length:
                if(correlation ((S1_cH3, S1_cV3, S1_cD3) ,  (cH3', cV3', cD3') >
                   correlation ((S1_cH3, S1_cV3, S1_cD3) ,  (cH3', cV3', cD3')))
                        message[i] = 1
                else
                        message[i] = 0

        decoded_message = correct_message(msg, ECC)
        return decoded_message
```

3.2 Choosing an Offset

In the previous section, we determined the method in which the base image could be modified with a message and in which the message could be retrieved from the photographic image. With that done, we wish to search for a choice of wavelet that can be most effectively used in the proposed algorithm. This wavelet desirably must retain its information after transmission and will not distort the original image in a way that severely degrades the quality of the base image.

Since information in the algorithm is retrieved from determining the cross correlation of the original image with the modified image for each of the segments, we know that how well the information can be retrieved from the base image must be related to the effect of the offset constant K used to modify the wavelet coefficients and to the size of the affected regions during modification. To determine the optimal constant K used for the message retrieval operation, an image was modified with a sequence of all 1's with varying K's using the above algorithm where the size of the affected region is 3x3. The wavelet used here will be DB10. The Structural Similarity Integrity Measurement (SSIM) was then taken between the retrieved image and the original image to determine the level of degradation to the image and was compared to the bit retrieval rate at that specific K value.

The SSIM between two windows x and y is defined as [11]:

$$S(x,y) = l(x,y) \cdot c(x,y) \cdot s(x,y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right), \quad (8)$$

where *l*, *c*, and *s* represent the similarity of the luminance of the windows, similarity of the contrasts of the windows and similarity of the structures of the windows respectively. μ_x and μ_y represent the mean intensities of each of the windows, σ_x and σ_y represent the standard deviations of the intensities of the windows using zero mean inputs and C_1, C_2, and C_3 are stabilizing constants. It has a maximum value of 1 and a minimum value of 0.

3.3 Choosing the Size of Watermark

Once the optimal offset has been determined, we must the size of the effected region of the watermarks. This will be done in a similar manner to how the offset was chosen. We will modify the original image with a sequence of all 1's with varying sizes of the affected regions. The offset obtained in the previous sections results will be utilized here and will remain constant. The level of degradation between the original image and the photographic image will once again be captured by the SSIM and will be compared to the size of the affected region to determine the optimal results.

In this thesis, the watermark will be created by modifying a 3x3 group of coefficients in each of the subsections in the third level wavelet decomposition discussed above. This is done to make sure that the original message can still be recovered even if there are slight shifts or contortions of the image caused by either the camera or the homography function in OpenCV.

3.4 Choice of Wavelet

Probably the most important factor in determining how much the base image will be modified in this algorithm is the choice of wavelet. In the DWT, there are two types of wavelet families: orthogonal wavelets and biorthogonal wavelets [3]. By modifying the coefficients of wavelets from an orthogonal family, we can modify the base image with distortions in the horizontal, vertical or diagonal directions [3]. Modifying the coefficients of a biorthogonal wavelet family leads to distortions in the base image that have no inherent direction [3]. In this thesis, these two types of wavelets will be compared against one another by measuring how well a message can be retrieved after being encode with either an orthogonal family of wavelets or a biorthogonal family of wavelets. The wavelets in question are as follows: Haar, Daubechies, and Biorthogonal wavelet families. Below are two images. One of them is an image modified by the Daubechies 10 sample wavelet and the other is an image modified by the Biorthogonal 3 sample wavelet.

Figure 3.2 Example of the modifying the coefficients of Daubechies 10 sample (left) and Birothogonal 3 sample wavelet (right).

4. Results and Discussion

In this chapter, we present the results of the algorithm described in the previous chapter. The experiments that follow below were performed on a logo from the University of Illinois at Urbana Champaign of resolution 1255 x 1255. The image itself is shown below.

Figure 4.1: Base image of the University of Illinois logo used for the experiments that follow.

4.1 Determining the Offset

To determine the offset used in the algorithm outline in the previous chapter, the base image was modified by a message that contained all 1's. This image was modified by the Daubechies 10 sample wavelet and in 30 different locations. No error correction codes were applied to the retrieved messages. The message was then retrieved by the retrieval code after modification and photographing. In the following table, the percentage of the number of 1's bits retrieved correctly is measured against the structural similarity of the base image to the modified image. From the table, we see that with an offset of K = 200, the modified image is still very similar to the base image and that the percent of 1's correctly retrieved is at a maximum.

K	Percent of 1's Correctly Retrieved	SSIM
100	47.99	.9532
200	49.33	.9111
300	48.00	.8822

Table 1: Offset and SSIM

4.2 Repetition Coding Message Retrieval Results

We compare our method for encoding the base image with a message of 30 bits at a rate of 1/3 using Repetition Codes for five different wavelets which are the Haar, Daubechies 10 sample, Daubechies 20 sample, Biorthogonal 3 sample, and Biorthogonal 9 sample wavelets.

Wavelet	Percentage
Haar	.6200
Daubechies 10 sample	.6366
Daubechies 20 sample	.6100
Biorthogonal 3 sample	.6633
Biorthogonal 9 sample	.6533

Table 2 Percentage of Message Recovered Using Repetition Coding

4.3 Hamming Coding Message Retrieval Results

In addition, we tested how well-suited Hamming Codes were on reclaiming data that had been imbedded within the photo. In this section, 4 bytes were encoded using 7,4 Hamming Code to generate the payload to be encoded within the images. The wavelets used are the Haar, Daubechies 10 sample, Daubechies 20 sample, Biorthogonal 3 sample, and Biorthogonal 9 sample wavelets.

Wavelet	Percentage
Haar	.5563
Daubechies 10 sample	.5375
Daubechies 20 sample	.6250
Biorthogonal 3 sample	.5938
Biorthogonal 9 sample	.6188

Table 3 Percentage of Message Recovered Using Repetition Coding

4.4 Reed Solomon Coding Message Retrieval Results

We also compare the results found for the Repetition Error Correction code with the results of the Reed Solomon Error Correction code. In this section, 24 bits were used to as the original message to be encoded into the base image. The choice to encode the 24 bits instead of 30 bits into the image for this section comes from the fact that the Reed Solomon Coding implementation appends parity bytes to the original message to be used for decryption. So, the message length was shortened to the greatest length that contained an integer number of bytes.

The coding rate remains the same though at 1/3 and the wavelets used are the Haar, Daubechies 10 sample, Daubechies 20 sample, Biorthogonal 3 sample, and Biorthogonal 9 sample wavelets.

Wavelet	Percentage
Haar	.6458
Daubechies 10 sample	.6823
Daubechies 20 sample	.6083
Biorthogonal 3 sample	.7001
Biorthogonal 9 sample	.6167

Table 4 Percentage of Message Recovered Using Reed Solomon Coding

5. Conclusion and Future Work

In this thesis, we provided and implemented a new approach to transmitting and retrieving a digital watermark from a photographed image. In this approach, the locations of the watermark were pseudo randomly generated and the watermarked bits were retrieved by comparing the correlation of the modified image with two other images: the base image which is modified by a sequence of all 1's bits and an image that is not modified (to match how a 0 does not modify the image to be printed and photographed). As seen in the results chapter above, this implementation is able to retrieve the watermark with around 60% accuracy using Repetition Error Correction Codes and increases in accuracy to around 65-70% accuracy using the Reed Solomon Error Correction Codes.

Future work on this algorithm can be divided into three distinct modifications which may increase the percentage of correctly retrieved bits and one modification to increase the amount of information that can be encoded in the image. The first modification would be to modify how the locations of the coefficients to be modified were found. In the current implementation, the entire square image was chosen to be the "logo" to be watermarked. However, it is clear from looking at the image that there is a large amount of white space around the "I" in the University of Illinois logo. As found in the results earlier, any watermarks that were placed in the white space around the "I", were not able to be recovered from the camera since those watermarks did not stand out from the white background enough. In future work, on the algorithm, logo should be redefined as only the region of the image with either a non-zero-pixel intensity or a pixel intensity that is greater than an arbitrary percent of the intensity of the greatest pixel value. This would be done to ensure that the watermark is only applied to locations of the logo that would retrievable after a camera takes a picture of it. The second modification to the algorithm could involve, applying a type of deblurring filter to the retrieved camera image before retrieving the message encoded in the image. In the current implementation, the watermark is very sensitive to blurring effects caused by swaying the camera during the moment when the picture is taken. This causes much of the information contained in the high frequencies such as sharp edges or high contrast between neighboring pixel values in the image to be lost which are the frequencies which contain the information found in the orthogonal wavelet watermarks. A deblurring filter such as the unsharp filter or Wiener filter may allow those watermarks to be retrieved with greater accuracy since they have been used to get a greater retrieval percentage of encoded bits from the blurred print scan attacks [15]. The third modification of the algorithm could involve an extension of the algorithm to colored images. This would involve performing the current algorithm on each of the color channels of the colored image. This could be used as an added level of error correction. It may also be possible to encode three times the amount of information as the current implementation allows since there are three color channels to encode information in. However, encoding different watermarks in each of the color channels may degrade the visual integrity of the image much more than encoding the same information into all three of the channels. The fourth modification would be a relatively simple modification to how the zeros are encoded in the original message. Currently if there is a zero in the original message, the watermarked image coefficients are not modified at the location that correspond to that bit. This means that the current implementation can encode only half of the information that it possibly could encode. To correct this, the zeros could be encoded into the base image by adding a negative constant to the wavelet coefficients that correspond to a 0.

References

[1] F. Wasilewski, "Daubechies 10 wavelet (db10) properties, filters and functions - Wavelet Properties Browser", *Wavelets.pybytes.com*, 2019. [Online]. Available: http://wavelets.pybytes.com/wavelet/db10/. [Accessed: 24- Apr- 2019].

[2] H. Gerhard, F. Wichmann and M. Bethge, "How Sensitive Is the Human Visual System to the Local Statistics of Natural Images?", *PLoS Computational Biology*, vol. 9, no. 1, p. e1002873, 2013. Available: 10.1371/journal.pcbi.1002873 [Accessed 24 April 2019].

[3] T. Filiba, "Print-Scan Resilient Watermarking in the Wavelet Domain", Tel Aviv University, 2019.

[4] Y.-C. Chiu and W.-H. Tsai, "Copyright Protection by Watermarking for Color Images ...," 2004. [Online]. Available: https://people.cs.nctu.edu.tw/~whtsai/Conference_Papers_PDF/Chiu_Tsai_CVGIP_2004.pdf. [Accessed: 24-Apr-2019].

[5] Lo, H., Topiwala, S. and Wang, J. (2019). Wavelet Based Steganography and Watermarking. [online] Cs.cornell.edu. Available at: http://www.cs.cornell.edu/topiwala/wavelets/report.html [Accessed 1 May 2019].

[6] V. S. Jabade and S. R. Gengaje, "Literature Review of Wavelet Based Digital Image Watermarking Techniques," International Journal of Computer Applications, vol. 31, pp. 28–35, Oct. 2011.

[7] "Free Vector Of The Day #141: Globe Logo Concept Vector Art & Graphics | freevector.com," FreeVectors. [Online]. Available: https://www.freevectors.net/free-vector-of-the-day-141-globe-logo-concept-39183. [Accessed: 01-May-2019].

[8] L. Yu, X. Niu (IEEE member), and S. Sun, "Print-and-scan model and the watermarking countermeasure," Image and Vision Computing, vol. 23, no. 9, pp. 807–814, Sep. 2005.

[9] C. Lin and S. Chang, "Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process", International Symposium on Multimedia Information Processing, vol. 99, 1999. Available: https://pdfs.semanticscholar.org/088c/5c13179710d2bc576321ad768addc8a9d413.pdf. [Accessed 5 May 2019].

[10] V. Solachidis and L. Pitas, "Circularly symmetric watermark embedding in 2-D DFT domain", IEEE Transactions on Image Processing, vol. 10, no. 11, pp. 1741-1753, 2001. Available: 10.1109/83.967401.

[11] Zhou Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," IEEE Signal Processing Magazine, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[12] T. Kimoto, “Error-Correction Coding of Spatial Image Watermarks in Gray Code,” in 2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems, 2011.

[13] David J.C., “Repetition codes,” Inference, 10-May-1997. [Online]. Available: http://www.inference.org.uk/mackay/itprnn/1997/l1/node6.html. [Accessed: 08-May-2019].

[14] M. Riley and I. Richardson, “Reed-Solomon Codes,” CMU, 1999. [Online]. Available: https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html. [Accessed: 08-May-2019].

[15] R. Riad, H. Douzi, M. El, R. Harba and F. Ros, "A Print-Scan Resilient Watermarking based on Fourier Transform and Image Restoration", International Journal of Computer Applications, vol. 128, no. 15, pp. 13-17, 2015. Available: 10.5120/ijca2015906192.

[16] N. Shep and P. Bhagat, "Implementation of Hamming code using VLSI", International Journal of Engineering Trends and Technology, vol. 4, no. 2, pp. 186-190, 2019. Available: https://pdfs.semanticscholar.org/9c21/38dd3529183a85a360bf19c9ba4bdefec0c8.pdf. [Accessed 10 May 2019].

[17] EE 387, Notes 4, Handout #6. Stanford, 2015, pp. 1-12.