Watermarking Technologies - Analysis and Design Report (December 2005)

Patrick Lam, Orion Winkelmeyer, Syed Abbas Mehdi, Nima Kamoosi

Abstract— **With so many forms of media taking on the digital format in recent times, it is necessary to provide security for ones valuable intellectual property (IP). Watermarking is the process of inserting information into a piece of digital data in an unperceivable manner while retaining this information's recoverability. Watermarking is at the forefront of IP protection and is constantly under development due to its proven ability to deliver security of digital data. This paper first analyzes two of the most popular domains of watermarking algorithms, namely Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). Then it gives an overview analysis of Perceptual Masking, a technique that effectively utilizes the characteristics of the Human Visual System (HVS) in order to embed a watermark. Then, finally, this paper describes a technique of using Perceptual Masking in conjunction with DCT: allowing for high strength watermarks to be inserted with low image distortion. This technique evidently preserves exceptional image quality and robustness under a variety of attacks. For the purpose of this research, which later turned into a design, all three algorithms based on DCT, DWT and DCT with PM were realized in MATLAB, and the results were analyzed and compared.**

I. Introduction

With data and multimedia taking on digital format, there is a need to protect digital property. There are two ways of accomplishing this: encryption and watermarking. Encryption protects information during transmission, but after its arrival at its destination, it is decrypted and is no longer protected. Watermarking is meant to compliment encryption in an effort to protect data after it has been decrypted. A watermark should be imperceptible (either visually or audibly depending on the application), yet robust enough to withstand intentional and/or unintentional attacks. There are many applications of watermarking [1], including:

Copyright Protection: The owner of the digital property embeds his/her copyright information in the digital data. This can prove security for copyright infringement, and prove ownership in court.

Copy Protection: This features allows developers and manufacturers of electronic data copying devices to develop there products so they recognize watermarked material, and not allow the user to make unauthorized copies of it.

Fingerprinting: This allows the owner of the digital property to pinpoint the source of illegal copies of there product. The owner embeds a unique watermark for each customer, and upon finding an illegal copy, the owner can trace the customer who leaked the property to third party. Letting customers know about this feature has proven as an effective measure for property security.

Medical Safety: As a safety measure for the medical industry, patients' digital medical images are watermarked with the patient's information.

Data Authentication: Fragile watermarks are used to detect altered data and provide information about the altering source. They have proven to be a sound deterrent against attempts to deface digital data, and are therefore an effective security mechanism.

II. Assumptions

Throughout the following section(s) we discuss processing images and assume the image to be of monochromatic nature and to have a square size of N×N. Each pixel's assigned data is a byte indicating monochromatic color intensity. The concepts can be extended to color and non-square images by respectively separating color components and cropping down to a square image.

III. Watermarking Using DCT and Sub-sampling

DCT watermarking involves sub sampling images from the original picture to be able to create the watermark for that image. In these sub samples a select fixed number of highest magnitude DCT coefficients and randomly perturbed. Therefore the watermark is placed to the perceptually significant components of the image. This method is quite robust against several known manipulations.

The original image is first decomposed into sub images which are transformed via DCT to obtain a set of coefficients. The coefficients are examined in pairs to verify whether they are appropriate for insertion. When a coefficient pair has been deemed appropriate for insertion a watermark insertion sequence is needed to determine how the different coefficients from the sub images are employed during encoding. Since the number of possible ordering sequences is huge, a person with no knowledge of the exact order of the sequence cannot recover the watermark in a reasonable amount of time. When generating an insertion sequence it should be done in such a way that a regular pattern will not appear on the watermarked image. The basic requirement is that four pixels in a 2x2 block of the original image must be assigned to different sub images. Once we have the set of watermarked images they are transformed via the inverse DCT, then they put together to compose the final watermarked image.

For the decoder, the input image is decomposed and transformed in the same manner as the insertion. The same watermark insertion order sequence is used to select the pairs of coefficients. Given the original sequence and the recovery sequence we use a distance measure to establish the closeness between the two. By using this closeness between the two we can establish a threshold limit. To decide whether the watermarked image was the original image, we establish a way to determine the threshold which if greater than the threshold level, a valid detection is declared.

A. Insertion Algorithm

First step is to divide the original image into 4 sub-images with interlaced pixels. Figure 3.1 shows the relative positioning of original image and sub-images. The following shows how the pixels are positioned:

$$r,s = 0,\ldots, N/2-1$$
$$v_1[r,s] = v[2r,2s],\ v_2[r, s] = v[2r+1,2s]$$
$$v_3[r,s] = v[2r,2s+1], v_4[r,s] = v[2r+1,2s+1] \tag{1}$$

Given an image of size $\boldsymbol{N}\times\boldsymbol{N}$ the sub-images will be of size $\boldsymbol{N}/2\times\boldsymbol{N}/2$.

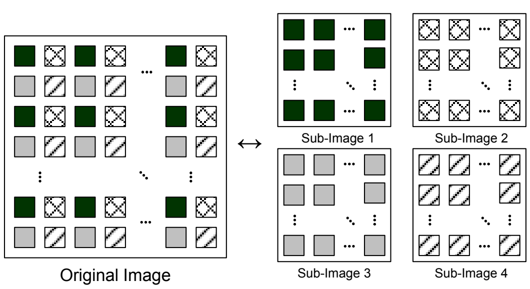

Figure 3.1: image (de)composition into/from sub-images

Next step is to transform the sub-images using the DCT method. The transform takes an image of size $\boldsymbol{N}/2\times\boldsymbol{N}/2$ and yields a matrix of equal size containing transformation coefficients.

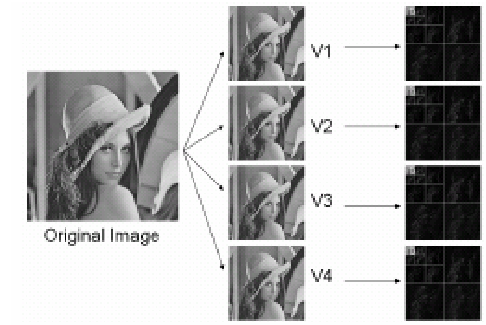

Figure 3.2: Image decomposition and transformation

Next we use these coefficient matrices to insert the watermark. Considering that the water mark is a sequence of $\boldsymbol{M}$+1 floating values $\boldsymbol{W}$: ($\boldsymbol{W}$[0], …, $\boldsymbol{W}[\boldsymbol{M}]$), each ranging in interval (-3, +3), we will modify $\boldsymbol{M}$+1 corresponding pairs of coefficients as shown below. $\boldsymbol{W}$ is required to have standard normal distribution.

$\boldsymbol{\alpha}$ is defined as a positive float value denoting the watermark strength coefficient. We also define a watermark insertion order sequence $\boldsymbol{O}$: ($\boldsymbol{O}$[0], …, $\boldsymbol{O}[\boldsymbol{M}]$), consisting of ordered pairs of values chosen from {1, 2, 3, 4}. These indicate the order in which coefficients are processed for insertion. We assume no four consecutive pairs are the same.

Example of such a sequence: $\boldsymbol{O}$[0] = (1,2) ; $\boldsymbol{O}$[1] = (2,4) ; $\boldsymbol{O}$[2] = (1,3) ; … $\boldsymbol{O}[\boldsymbol{M}]$ = (4,2) ;

For the insertion we index $\boldsymbol{O}$ and $\boldsymbol{W}$ using $\boldsymbol{i}$: 1…$\boldsymbol{M}$; Each sub-image's coefficients are indexed following a zigzag pattern yielding the four coefficients $\boldsymbol{V}_1[\boldsymbol{i}]$, $\boldsymbol{V}_2[\boldsymbol{i}]$, $\boldsymbol{V}_3[\boldsymbol{i}]$, $\boldsymbol{V}_4[\boldsymbol{i}]$ for a given $\boldsymbol{i}$. Figure 3.3 demonstrates the zigzag pattern and how the coefficient pairs are selected.

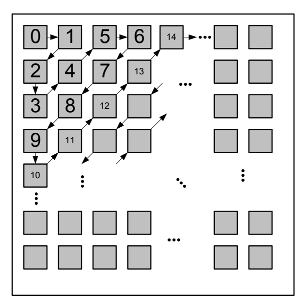

Figure 3.3: zigzag pattern in a square matrix

The process of inserting the watermark bits is performed by iterating though $\boldsymbol{i}$: 1…$\boldsymbol{M}$ and modifying the sub-image coefficients in the following way:
Given $\boldsymbol{i}$ and $\boldsymbol{O}[\boldsymbol{i}] = (\boldsymbol{a},\boldsymbol{b})$, the two coefficients $\boldsymbol{V}_{\boldsymbol{a}}[\boldsymbol{i}]$, $\boldsymbol{V}_{\boldsymbol{b}}[\boldsymbol{i}]$ are chosen denoted for short as $\boldsymbol{V}_{\boldsymbol{a}}$, $\boldsymbol{V}_{\boldsymbol{b}}$. Find the average:

.
$$V = \frac{V_a + V_b}{2} \tag{2}$$

When the following is true:

$$\left|\frac{V_a - V_b}{V}\right| \geq 6\alpha \quad (3)$$

We move further in the zigzag pattern without increasing index $\boldsymbol{i}$. Otherwise these modifications are made to $\boldsymbol{V_a}$, $\boldsymbol{V_b}$:

$$V'_a = V(1+\alpha W) \quad V'_b = V(1-\alpha W) \quad (4)$$

After the watermark is inserted into the transformed sub-images, each sub-image must be transformed back through DCT inverse transform. Consequently the sub-images are recombined in the same way they were decomposed, giving us a watermarked image of size $\boldsymbol{N}\times\boldsymbol{N}$.

B. Recovery Algorithm

In order to ensure an image is watermarked with a known watermark sequence $\boldsymbol{W}$: ($\boldsymbol{W}$[0], …, $\boldsymbol{W}$[$\boldsymbol{M}$]), we need to recover a watermark sequence $\boldsymbol{W'}$: ($\boldsymbol{W'}$[0], …, $\boldsymbol{W'}$[$\boldsymbol{M}$]) from the image and compare $\boldsymbol{W}$ with $\boldsymbol{W'}$. The comparison is done using probability techniques and the watermark is considered valid only of the comparison satisfies certain criteria.

The watermark recovery algorithm's first step is image decomposition. As in the insertion algorithm the image (of size $\boldsymbol{N}\times\boldsymbol{N}$) is divided into 4 sub-images. Each sub-image (of size $\boldsymbol{N}/2\times\boldsymbol{N}/2$) is transformed using DCT into a coefficient matrix of the same size.

Similar to the insertion algorithm, we iterate though $\boldsymbol{i}$: 1…$\boldsymbol{M}$ and index the sub-image's coefficients following a zigzag pattern yielding the four coefficients $\boldsymbol{V'}_1[\boldsymbol{i}]$, $\boldsymbol{V'}_2[\boldsymbol{i}]$, $\boldsymbol{V'}_3[\boldsymbol{i}]$, $\boldsymbol{V'}_4[\boldsymbol{i}]$ for a given $\boldsymbol{i}$. Given $\boldsymbol{O}[\boldsymbol{i}]$ = ($\boldsymbol{a}$,$\boldsymbol{b}$) the two coefficients $\boldsymbol{V'_a}[\boldsymbol{i}]$, $\boldsymbol{V'_b}[\boldsymbol{i}]$ are chosen denoted for short as $\boldsymbol{V'_a}$, $\boldsymbol{V'_b}$. We find the average:

.

$$V' = \frac{V'_a + V'_b}{2} \quad (5)$$

We skip further in the zigzag pattern without increasing index $\boldsymbol{i}$ if:

$$\left|\frac{V'_a - V'_b}{V'}\right| > 6\alpha \quad (6)$$

Otherwise the water mark bit $\boldsymbol{W'}[\boldsymbol{i}]$ (denoted by $\boldsymbol{W'}$ for short) is calculated as follows:

$$W' = \frac{1}{\alpha}\left(\frac{V'_a - V'_b}{V'_a + V'_b}\right) \quad (7)$$

The resulting watermark sequence $\boldsymbol{W'}$ is compared with the original watermark sequence $\boldsymbol{W}$ using the sim() function. Since $\boldsymbol{W}$ is assumed to have a standard normal distribution, and $\boldsymbol{W'}$ is expected to be treated as a random output, their independence implies that sim($\boldsymbol{W}$, $\boldsymbol{W'}$) will be close to zero. However sim($\boldsymbol{W}$, $\boldsymbol{W'}$)>6 disproves their independence with high certainty. Thus if,

sim($\boldsymbol{W}$, $\boldsymbol{W'}$)>6, (8)

Then it is concluded that the watermarked image has watermark $\boldsymbol{W}$.

IV. Watermarking using DWT and Sub-sampling

In the DWT method of implementation it is very similar to DCT. The only difference is that we use DWT instead of DCT to transform the sub images to obtain the set of coefficients. Then we use the inverse DWT instead of inverse DCT to compose the final image. The advantage of using DWT over DCT is that you can use a stronger watermark while still being able to maintain a high image quality unlike DCT.

A. Insertion Algorithm

As in the DCT watermark insertion algorithm, first step is to divide the original image into sub-images. Then the sub-images are each transformed using the DWT method. The transformed sub-images are then processed exactly as in the DCT watermark insertion algorithm and the coefficients are modified. The resulting sub-images are then transformed back using DWT. Consequently the sub-images are recombined in the same way they were decomposed, giving us a watermarked image of size $\boldsymbol{N}\times\boldsymbol{N}$.

B. Recovery Algorithm

The watermarked image is decomposed into 4 sub-images as before and each sub-image is transformed using DWT. Consequently the transformed sub-images are scanned using the same zigzag pattern and pairs of coefficients are selected as in the DCT watermark recovery algorithm; $\boldsymbol{W'}$ is also calculated the same way.

The resulting watermark sequence $\boldsymbol{W'}$ is compared with the original watermark sequence $\boldsymbol{W}$ using the sim() function and the retrieved watermark is considered to originate from $\boldsymbol{W}$ if sim($\boldsymbol{W}$, $\boldsymbol{W'}$)>6.

V. DCT and Perceptual Masking

When imposing a watermark onto a digital piece of work, there are several methods. Spread Transform Scalar Costa Scheme watermarking is a more traditional scheme which projects a watermarking candidate onto a pseudo-random vector. The shortfall of this scheme is that the watermark is generally perceivable by the naked eye. To attain imperceptibility, perceptual masking derives a unique perceptual mask sequence from the original work and applies it back to the original work to produce the watermark.

Perceptual masking is derived from the Watson's perceptual model. The model divides an image into independent DCT blocks. Each block is of fixed size and the perceptibility of changes is estimated by the changes in coefficient in each block. In order to achieve the Perceptual Masking with DCT, we simply split the DCT matrix of the given image into many 8x8 sub-matrices, and then multiplied each of these sub-matrices with the matrix of table 4.1. Then we put the multiplied components back together to achieve the watermarked image.

The Watson's perceptual model is based on 3 components namely the frequency sensitivity function, luminance and contrast masking components.

A. Frequency Sensitivity

In Watson's model, frequency sensitivity is the amount of frequency change a block can withstand before it is perceivable by the eye. A frequency table contains entries denoted **fs**[*i, j*] (i, jth pixel in a block) as seen in Table 4.1. Each entry in the table represents the smallest magnitude of the corresponding DCT coefficient in a block that can be perceived by the eye also known as the Just Noticeable Difference (JND).

(j) \ (i)	0	1	2	3	4	5	6	7
0	1.404	1.011	1.169	1.664	2.408	3.433	4.796	6.563
1	1.011	1.452	1.323	1.529	2.006	2.716	3.679	4.939
2	1.169	1.323	2.241	2.594	2.988	3.649	4.604	5.883
3	1.664	1.529	2.594	3.773	4.559	5.305	6.281	7.600
4	2.408	2.006	2.988	4.559	6.152	7.463	8.713	10.175
5	3.433	2.716	3.649	5.305	7.463	9.625	11.588	13.519
6	4.796	3.679	4.604	6.281	8.713	11.588	14.500	17.294
7	6.563	4.939	5.883	7.600	10.175	13.519	17.294	21.156

TABLE 4.1 Frequency Sensitivity Table

B. Luminance Masking

Luminance masking accounts for the effect of the DC-component (average brightness of the block) on the frequency sensitivity table. A higher DC-component is a DCT coefficient that can be changed by an amount larger than that indicated in Table 3.1. If this is the case, the table needs to be adjusted using the DC term for that block.

C. Contrast Masking

Contrast masking accounts for the effect of visibility of a change in one frequency due to the energy present in that particular frequency. A contrast masking threshold is generated from the luminance masking threshold and it represents the JND a coefficient can withstand.

VI. MATLAB Result Comparison

This section will compare the different algorithms used for watermarking. Discrete Cosine Transform, Discrete Wavelet Transform as well as DCT with perceptual masking were all implemented using MATLAB and results were giving in the form of images and similarity score.

The value alpha (α) is a variable used to determine the strength of the watermark. For the purposes of this analysis report, we have chosen the α values to be 0.2, 0.5 and 0.9. This yields some very noticeable results however; typically in industry this α value is quite low; 0.2 or lower. From Figure 5.1, as the watermark strength coefficient increases, the picture appears to have more horizontal distortion lines. This is the effect of the watermark strength and it becomes a tradeoff. Because the requirement of a watermark is for it to be imperceptible, yet still remain robust to processing, it is necessary to find a well balanced coefficient.

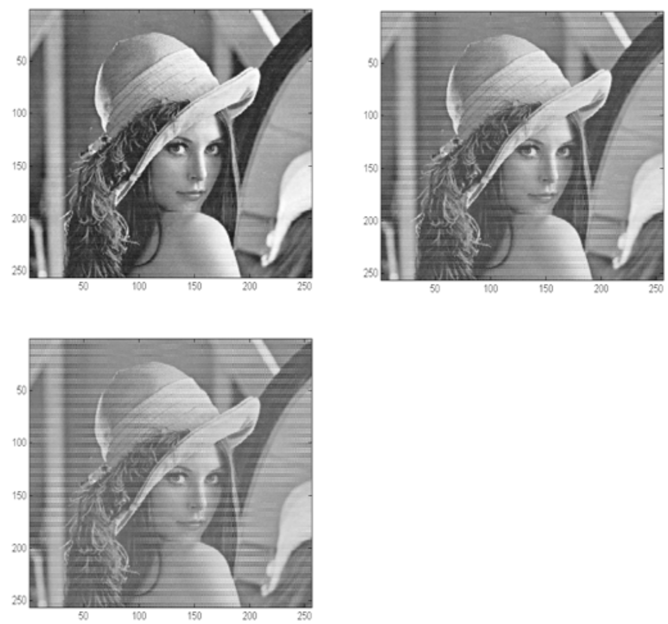

Figure 5.1: Effect of α (0.2, 0.5, and 0.9 respectively)

Our implementations of the algorithms and its usage on images yielded some fairly interesting results. The following pictures are watermarked using DCT, DWT and DCT with Perceptual masking while constraining the α to be the same throughout the sample space. In Figure 5.2, DCT yields an image with some defined watermarking lines. The distribution of the watermark is quite evenly spread out. In the DWT implementation, the image reveals a much chunkier watermark leaving stronger changes in different areas. Figure 5.3 shows there are some blocks with much heavier watermarking effects. With perceptual masking applied to DCT, the results are much less obvious. In Figure 5.4, the watermarking lines are much less visible. This is because of the contrast masking of the Watson's model used in perceptual masking.

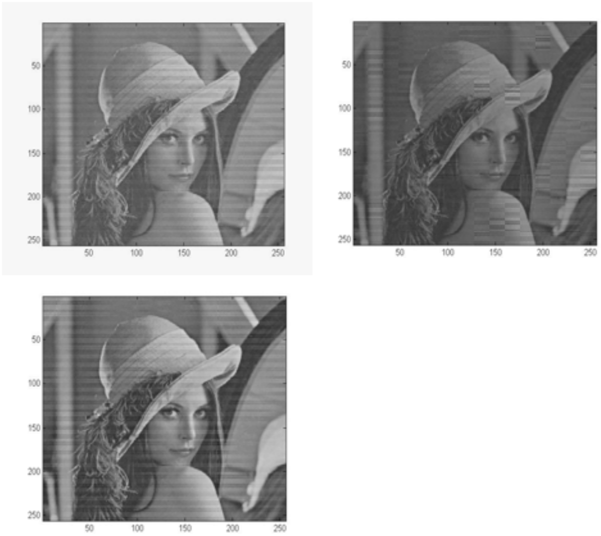

Figure 5.2: DCT, DWT and DCT with PM

To compare the experimental results, we have devised a system to compare the algorithms effectiveness in embedding a watermark which is recoverable. For each image that we watermark, we assign a score based on how similar the recovered watermark is to the embedded one. This value is called a Similarity Score; the higher the value the better. Table 5.1 shows a similarity score table with the experimental results we were able to derive from our tests.

	α = 0.2	α = 0.5	α = 0.9
DCT	*0.1925*	*0.378*	*0.251*
DWT	*15.4677*	*15.7182*	*15.8435*
DCT + PM	*31.4129*	*59.1437*	*78.0076*

Table 5.1 Similarity Score Table

VII. Conclusion

Note that all the values in the similarity score table are relative. Alone, these numbers do not mean anything. For example, when the value of α = 0.1, it can be seen that DCT had a score of 0.1925 while DWT had a score of 15.4677 and DCT with Perceptual Masking scored 31.4129. This shows that DWT is about 80 times better than DCT. With the scoring of DCT with Perceptual Masking, it is about 2 times better than DWT and about 160 times better than regular DCT. Similarly, the scores increase as the value of α is increased. One value that seems to be an outlier is DCT while the value of the coefficient is 0.5. Notice the score apexes at this value and there is a gradual decline of the value as α continues to increase.

We can confidently conclude, that the method of DCT with Perceptual Masking that we proposed is better than both, DCT and DWT.

References

[1] Gerhard C. Langelaar,Iwan Setyawan, and Reginald L. Lagendijk, “Watermarking Digital Image and Video Data” *IEEE Signal Processing Magazine*, Sept 2000.

[2] Min-Jen Tsai and Hsiao-Ying Hung, “DCT and DWT-based Image Watermarking by Using Sub-sampling,” *Institute of Information Management*, National Chiao Tung University.

[3] Wai C. Chu, “DCT-Based ImageWatermarking Using Subsampling”, *IEEE Transaction On Multimedia*, Vol. 5, No. 1, Mar. 2003.

[4] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia,” *IEEE Trans. on Image Processing*, vol 6, No. 12, pp 1673-1687, Dec. 1997.