Project Report
EE 371R Digital Image Processing
December 6, 2001

Onur Mutlu

An Overview of Image Watermarking Algorithms

1. Introduction

Digital watermarking is the process of conveying information by imperceptibly embedding it into the digital media. The purpose of embedding such information depends on the application and the needs of the owner/user of the digital media. Current main applications of watermarking include the following:

1. Copyright protection: The objective is to embed information about the source/owner of the digital media in order to prevent other parties from claiming the ownership of the media.

2. Fingerprinting: The objective of fingerprinting is to convey information about the recipient of the digital media (rather than the owner) in order to identify every single distributed copy of the media. This concept is very similar to serial numbers of software products.

3. Copy protection: Watermarking can be used to control data copying devices and prevent them from copying the digital media in case the watermark embedded in the media indicates that media is copy-protected.

4. Image authentication: The objective is to check the authenticity of the digital media. This requires the detection of modifications to the data.

This project does not specifically focus on a single application of watermarking. Rather, it implements several different watermarking algorithms which may or may not be desirable for a variety of applications.

However, I only focus on watermarking of images and leave the problem of video watermarking as future work.

2. Motivation and Objectives

As described in the introduction, image watermarking algorithms is the main focus of this project. I got interested in the topic due to the first application of watermarking as described above. I wanted to know about how one can embed information in an image such that he can later claim the ownership of that image by extracting back the embedded information. Hence, “copyright protection” of images was my main motivation in starting this project.

As I had no background in image watermarking before I started the project, I set the following as my objectives before starting the project:

1. Understanding the requirements of image watermarking based on its applications. A good understanding of these requirements is the first step in designing algorithms for different watermarking applications.

2. Familiarizing myself with the watermarking literature that has been developing fast in the last decade.

3. Understanding how the robustness of the image watermarks can be improved.

4. Implementing several of the watermarking algorithms and examine them in terms of how they meet the requirements of different applications and general requirements of watermarking.

5. Most importantly, applying and extending the information and techniques I learned in this course (such as edge detection, discrete fourier transform, discrete cosine transform, linear filtering, etc).

3. Requirements of Image Watermarking

An image watermarking system needs to have at least the following two components:

1. A watermark embedding system
2. A watermark extraction (recovery) system

The watermark embedding system takes as input the watermark bits, the image data, and optionally a secret or public key. The output of the watermark embedding system is the watermarked image. The watermark extraction system takes as input an image that possibly contains a watermarks and possibly a secret or public key. Depending on the type of watermarking system used, it may also take as input the original image or the watermark. The watermark extraction system determines whether a watermark is present or absent in the image. It may also output a confidence measure that indicates the probability with which the watermark is present in the image.

Besides these two requirements, a useful watermarking scheme also has the following properties:

1. Imperceptibility of the watermark: The watermarking system must embed the watermark in the image such that the visual quality of the image is not perceptibly distorted. Hence, a measure of distortion needs to be used when determining the imperceptibility of the watermarking algorithm. In this project, I did not use any mathematical metric (such as MSE or PSNR) to quantify the distortion due to watermarking. Instead, I commented on the visual quality of images by comparing how the original image and watermarked image look.

2. Robustness of the watermarking scheme: Most of the watermarking applications require that the watermark should still be recovered even if the image is distorted. Perhaps we can call a watermarking algorithm “robust” if recovery of the watermark cannot be made impossible without perceptibly distorting the image (This definition is good for the purposes of my project). Robustness is not required for all applications. For example, a fragile watermark that has to prove the authenticity of the host data does not have to be robust against alterations of the image. This is due to the fact that, in this application, failure to detect the watermark proves that the host data has been modified and the image is therefore not authentic.

3. Security: The security of watermarking techniques is very similar to the security of the encryption techniques. A watermarking technique is truly secure if knowing the algorithms to embed and extract the watermark does not help an unauthorized party to detect the presence of the watermark [1].

4. Payload of the watermark: The amount of information that can be stored in an image for watermarking depends on the application and the image. Usually, the robustness of the watermark is increased if the payload of the watermark is bigger.

5. Oblivious vs. non-oblivious watermarking: Some applications (copyright protection) can use the original image to extract the watermark from another image. This is called non-oblivious watermarking. However, many applications (copy protection) need to extract the watermark or detect the existence of a watermark without access to the original image. This is called oblivious watermarking and it is a much harder problem. (Oblivious watermarking algorithms also do not have access to the embedded watermark bits.)

4. Focus of This Project: Imperceptibility and Robustness of the Watermark

For the purposes of this project, I focused on two requirements of image watermarking. I implemented different watermarking algorithms and observed the imperceptibility of the watermark embedded using each algorithm. I also report results on the robustness of the examined algorithms. I examine the robustness of the images based on different transformations applied to images. There are many possible transformations that can be applied to watermarked images and which might possibly render watermark extraction impossible. Examples include addition of noise, filtering, lossy

compression, affine transformations (rotation, scaling, etc), cropping, and multiple watermarking. More serious modifications might intentionally try to transform the image such that watermark will not be extracted. As there are many possible attacks that can be performed on the image, in this project, I mainly focus on how the examined watermarking schemes perform with respect to scaling, filtering, and compression.

5. Implementation and Evaluation of Several Watermarking Algorithms

This section describes the watermarking algorithms I implemented and tested. I will comment on the imperceptibility and robustness of each watermarking algorithm and the experiments I performed to determine this. The discussion of each algorithm is not extensive. More details can be found in the Matlab code.

5.1. Least Significant Bit Substitution

Using Least Significant Bit manipulation, a huge amount of information can be hidden with very little impact to image quality. This technique is performed in the spatial domain.

The embedding of the watermark is performed choosing a subset of image pixels and substituting the least significant bit of each of the chosen pixels with watermark bits. Extraction of the watermark is performed by extracting the least significant bit of each of the selected image pixels. If the extracted bits match the inserted bits, then the watermark is detected. The extracted bits do not have to exactly match with the inserted bits. A correlation measure of both bit vectors can be calculated. If the correlation of extracted bits and inserted bits is above a certain threshold, then the extraction algorithm can decide that the watermark is detected.

The implementation of this algorithm is quite simple. However, some policy decisions should be made. For example, how should the set of pixels to be modified be selected? One way to select these elements is by using a pseudorandom number generator [2]. Also, the watermark extractor should have access to these selected elements.

Imperceptibility and Robustness of the Algorithm

The visual quality of the image does not change significantly because the watermark bits only change the least significant bits of some pixels. Hence, the addition of the watermark to an image using this algorithm is quite imperceptible. On the other hand, this algorithm is not very robust, due to the same reason. As the least significant bits of pixels do not contribute to the image much, some attacker can possibly zero out several least significant bits of all pixels of the image and hence clear the watermark. *This suggests that it may not be a good idea to insert the watermark bits to non-significant parts of the image*. An image that is watermarked using this algorithm is shown in Figure 1 (a,b,c). This algorithm also will not be robust against JPEG compression because it is performed in the spatial domain and involves least significant bits of the image pixels. I will show that DCT-domain-based watermarking techniques are more robust to JPEG compression.

5.2. Patchwork Algorithm

This algorithm is an extension of the algorithm proposed by Bender et. al. [3]. During the insertion process, n pixel pairs are selected pseudorandomly using a secret key K. The luminance values (a_i, b_i) of the n pixel pairs are then modified slightly such that

$\underline{a}_i = a_i + 1$ and $\underline{b}_i = b_i - 1$

Extraction process retrieves the n pixel pairs which were used in the encoding step. Then, the sum

$S = \Sigma (\underline{a}_i - \underline{b}_i)$ over i=1 to n

is computed. If the image actually contains a watermark, then the expected value of the sum is 2n. Otherwise, it should be approximately 0. This reasoning is based on the statistical assumption that

$$E[S] = \Sigma (E[\underline{a}_i] - E[\underline{b}_i]) = 0$$

This assumption only holds if the pixel pairs are randomly chosen and if they are independently and identically distributed. However, this assumption is not quite true. Even though it is not quite true, in my implementation of the algorithm I saw that this is a good approximation. Hence, S will be close to 2n if the image actually does contain the watermark (The provided code shows this).

Imperceptibility and Robustness of the Algorithm

As seen in Figure 2, this algorithm also has imperceptible effects. This is due to the fact that it does not significantly modify the image pixels. However, robustness of this algorithm is not high. Robustness of this algorithm depends on the assumption based on E[S] being true. However, very basic pixel operations can invalidate this assumption. My experiments with this algorithm showed that movement and translation of pixels, basic filtering operations such as erosion and dilation change the values of $\underline{a}_i$ and $\underline{b}_i$ such that the assumption on E[S] does not hold any more. Hence, the watermark becomes undetectable.

5.3. Correlation-based Watermarking in the Spatial Domain

This is a generalized algorithm that relies on correlational techniques for the extraction of the watermark. Algorithms of this class add some pseudorandom noise to the image as the watermark. This noise (W(i,j)) is generated based on a secret key. The only requirement for this noise is that it should be uniformly distributed and the noise pattern should not be correlated with the image content. Watermarked image, WI(x,y) is obtained using the following equation:

$WI(i,j) = I(i,j) + k.W(i,j)$ (I is the original image and k is a gain factor)

To detect a watermark in an image J(i,j), the correlation between J(i,j) and the watermark (noise pattern) W(i,j) is calculated. If this correlation is greater than some predetermined threshold, then the watermark detector concludes that the given watermark W is present in image J. Otherwise, the image is deemed to be non-watermarked. My implementation of this algorithm estimates this correlation using a fast algorithm provided in [4].

As it is the case with any watermark detector, a correlation-based watermark detector can make two types of errors: It can detect the existence of a watermark, although there is none, or it can reject the existence of a watermark although there is one. Using probability theory [4] it can be shown that, probability of making both errors decreases by increasing the gain factor k. However, increasing the gain factor k also degrades the visual quality of the image.

Imperceptibility and Robustness of the Algorithm

Figure 3 shows the watermarked lena image (k=1, range(W)={-1,0,1}). We can see that this algorithm does not impact the visual quality if gain factor is kept small and noise pattern does not contain large values. However, this algorithm suffers the same robustness problem described for the patchwork algorithm. In fact, patchwork algorithm is a special case of this generalized model. Translation, rotation, scaling significantly affect the correlation values obtained and hence cause the watermark to go undetected or destroyed. Similarly, JPEG compression will also destroy the correlation between the watermarked image and the watermark bits. Therefore, we still need better algorithms that endure these operations.

Using Filtering to Improve the Detectability of the Watermark

One problem I found with this technique is that image content can interfere with the watermark and the correlation between the image and the watermark may be rendered meaningless. This is especially true for low frequency image components. [5] suggests the application of filtering to reduce this interference before the calculation of the correlation. Hence, I filtered the image using the convolution kernel suggested in [4] and shown in Figure 4. Applying this filter significantly improved the detectability of the watermark when the interference between image content and watermark was high.

5.4. CDMA Watermarking

This technique is actually intended to increase the payload of the watermark. Increasing the payload of the watermark intelligently increases the probability that the watermark will be detected using a correlation-based technique. This technique is based on the use of Direct Sequence Code Division Multiple Access (CDMA) spread spectrum communications as proposed by [6]. For each bit b_i of the watermark, a different independent pseudorandom pattern P_i is generated that has the same size as the image to be watermarked. This pattern is dependent on the bit value b_i. For example if b_i is 0, P_i is added to the image, else P_i is subtracted from the image. Mathematically, the watermarked image can be expressed as follows:

$WI(i,j) = I(i,j) + k.\Sigma ((-1)^{bi} P_i)$ where k is the gain factor.

Hence, each watermark bit contributes a positive or negative random pattern to the image to form the watermarked image.

Each bit b_i of the watermark can be extracted by calculating the correlation between normalized image J(i,j) and the corresponding random pattern P_i. If the correlation is positive, the watermark extraction algorithm decides that b_i is 0, otherwise b_i is assumed to be 1.

Imperceptibility and Robustness of the Algorithm

I found that this algorithm does not affect the visual quality of the image if small gain factors are used. Figure 5 shows an example. One problem I found with this algorithm is that random patterns P_i should actually be selected carefully, otherwise watermark extraction process is bound to have many errors. Let's say, if random patterns are selected such that their sum is a zero image, the watermarked image and the original image will be identical. Therefore, it will not be possible to detect the watermark by taking the correlations of the watermarked image and each bit pattern. This implies P_i that should not be random images but should be carefully selected to impose correlations.

I found that this algorithm is quite robust against cropping. It is also somewhat resilient to JPEG compression, however the probability that all watermark bits will be recovered after compression is usually low. Although it is also more robust to scaling and filtering compared to previously discussed algorithms, the computation time required for this algorithm can be quite high, especially if the number of watermark bits is high.

5.5. Watermarking Based on DFT Amplitude Modulation

In the spatial domain, if the image is shifted a little bit, the watermark extraction process will be disturbed greatly because the pixels will now be translated to different locations. Embedding the watermark in the DFT amplitude of the image overcomes this problem. *Due to the periodicity of the image implied by DFT, cyclic translations of the image in the spatial domain do not affect the DFT amplitude*. A watermark embedded in this domain is therefore translation invariant. The embedding process consists of selecting which amplitudes to modify to embed the watermark and modifying them in such a way that image quality doesn't degrade. After selecting the DFT amplitude

coefficients to embed the watermark, these coefficients can be modulated using the following equation [7]:

$$|WDFT(u,v)| = |DFT(u,v)|. (1 + k.W(u,v)) \quad \text{(Equation 1)}$$

where k is the gain factor, WDFT is the DFT of the watermarked image and W is the watermark image. This equation makes the watermark image content dependent. The larger DFT coefficients are affected more severely by this equation whereas smaller coefficients are not modified by much.

Imperceptibility and Robustness of the Algorithm

In my implementation, I decided to embed the watermark in mid-frequency components of the DFT amplitude. If a heavy watermark is embedded in low frequency components, then the image quality is slightly degraded. On the other hand, if the watermark is embedded in high frequency components, it is very vulnerable to noise, filtering and lossy compression. Therefore, I found it better to embed the watermark in mid-frequency components of the DFT amplitude. Figure 6 shows the watermarked image. We can see that watermarked image is a little bit brighter than the original and perhaps it even looks better.

Using this algorithm, I found that watermark is not affected by shifting the image as expected. However, this algorithm was still not strong against JPEG compression or geometric transformations. Therefore, I implemented the following algorithm to increase the robustness of the watermark against JPEG compression.

One other problem with using DFT amplitude modulation is the fact that DFT amplitude does not contribute too much to the image quality. This suggests that using *DFT phase modulation* will probably be more robust due to its high contribution to image quality. I also explored this possibility and will explain my experience in Section 5.8.

5.6. Watermarking Based on DCT Coefficient Modulation

None of the previously mentioned techniques are resilient enough to JPEG compression. This technique embeds the watermark in the DCT domain to increase the robustness of the watermarking scheme against JPEG compression. The idea in this algorithm is very similar to DFT amplitude modulation. The watermark bits are embedded in each 8x8 DCT block of the image. The embedding algorithm needs to carefully choose where to embed the watermark bits in the 8x8 block. My argument is similar to DFT amplitude modulation. It is not wise to embed the watermark bits in the low frequency components of the DCT block, because these coefficients are subject to heavy quantization during JPEG compression. Hence, it is better to embed the watermark in mid or high-frequency DCT components. If the gain factor k is chosen small, embedding the watermark in lowest-frequency components will be more desirable, because these components are the ones that are least likely to be quantized in JPEG compression. The actual modification of the selected DCT coefficients is done using Equation 1.

Imperceptibility and Robustness of the Algorithm

This algorithm also does not affect the visual quality of the image much if the gain factor is chosen as a small value. Figure 7 shows the watermarked image where the watermark is embedded in mid-frequency components. This figure shows the disadvantage of embedding the watermark in mid-frequency components. Complete quantization (clearing) of the coefficients in which the watermark is inserted does not degrade image quality much. However, the watermark will be irrecoverable. The watermarking scheme preserves its robustness against JPEG compression if the watermark bits are embedded in the lowest frequency DCT coefficients. Figure 8 shows the watermarked image in this case. There are only a few DCT coefficients in which the watermark can be embedded. If we want to increase the payload of the watermark, then these coefficients may need to be modified significantly

which probably will impact the quality of the image. Therefore, it might be better to use a scheme which embeds watermark bits into both low and mid-frequency DCT coefficients.

5.7. Watermarking Based on HVS (Human Visual System)

We would normally like to increase the energy of the watermark (or payload of the watermark) in order to increase its robustness. However, increasing the payload of the watermark degrades the visual quality of the image such that human eye will notice the degradation. A dual reasoning leads us to think that it might be better to increase the payload of the watermark by embedding the watermark bits into places where human eye will not detect the changes to the image. Several watermarking schemes were proposed by researchers that aim to exploit the characteristics of the human visual system. For example, [8] suggests to make the gain factor luminance dependent. This is because of the fact that Human Visual System (HVS) is less sensitive to changes in regions of high luminance.

I implemented another algorithm that exploits the fact that HVS is less sensitive to distortions around edges and textured areas of the image compared to distortions in smooth areas. We can exploit this property by increasing the payload (energy) of the watermark in those specific areas. We can create a mask image that consists of those areas that are less sensitive to distortions and modulate the watermark bits using this mask image. This can be mathematically expressed as:

$$WI(i,j) = I(i,j) + Mask(i,j).k.W(i,j)$$

W is the watermark pattern (image), k is the gain factor, and Mask is the mask image as mentioned above. In my implementation, I generate the Mask image using an edge detection algorithm. I convert the edge image into a binary image. I amplify the effect of watermark bits by k on pixels where edge image is '1' and keep the effect of the watermark bits minimal on pixels where edge image is '0'. This increases the energy of the watermark along the edges in the image. I use the canny edge detector to extract the edge information out of the image. Results of this algorithm are shown in Figure 10.

Imperceptibility and Robustness of the Algorithm

Due to the exploitation of HVS, this algorithm does not affect the visual quality of the image much. In fact it might even sharpen some parts of the edges in the image (Figure 9). By increasing the energy of the watermark, the algorithm becomes more robust compared to other spatial-domain watermarking techniques. I have tested the robustness of this algorithm with respect to JPEG compression and saw that it can endure higher levels of JPEG compression compared to other spatial-domain techniques such as CDMA Watermarking (as discussed in Section 5.4).

Unfortunately, due to its implementation in the spatial domain, this scheme is not robust against translation and shifting of the image pixels. However, we might be able to make this algorithm more robust by modifying the DFT of the image based on DFT amplitudes of the edge image obtained using the edge detector. I have not implemented this algorithm due to time considerations, but it sounds promising.

5.8. Watermarking Based on DFT Phase Modulation

Phase information of the DFT provides information about how different sinusoids form an interference pattern to form an image. This interference pattern is quite significant, slight changes in this pattern can destroy the image. Hence, DFT phase of an image is very important compared to the DFT amplitude. [9] suggests that DFT phase modulation is a good candidate for image watermarking.

If the watermark is introduced to the phase components of the image DFT with high redundancy, unauthorized parties would probably need to cause visually visible damage to the image to destroy the watermark. This is due to the great significance of DFT phase information in the

structure of the image. One algorithm proposed by [9] performs watermarking on an NxN image by modifying the frequency components as follows:

WatermarkedPhase(u,v) = Phase(u,v) + m

WatermarkedPhase(N – u, N – v) = Phase(u,v) – m

where Phase is the Phase matrix of the DFT and m is the watermarking level desired. Due to the symmetry of the DFT, watermark should be subtracted from one phase coefficient whereas it should be added to its symmetric counterpart. We would like to mark only those DFT phase coefficients which have significant contributions to the image structure. The selection of the DFT coefficients is done based on the corresponding DFT amplitudes. Phase(u,v) will be marked if the following holds:

Amplitude(u,v)/ΣΣAmplitude(i,j) > T , where T is a predetermined threshold.

Imperceptibility and Robustness of the Algorithm

If the watermarking level is not too high, the watermark is not perceptible in this algorithm. This can be seen in Figure 10. This algorithm is quite robust against modifications to image. DFT phase information cannot easily be destroyed by noise or changes to image contrast. Therefore this algorithm survives the kind of attacks that change the image contrast and those that employ filtering on the image.

5.9. Watermarking Based on DCT Coefficient Reordering

I also implemented and evaluated watermarking based on DCT coefficient reordering as proposed in [10] (Figure 11). Due to space limitations I will not discuss the details in this report. Details and discussion on the visual quality of watermarked images are provided in the code I submitted (koch_zhao.m). Many other watermarking algorithms were proposed and I read about some of those but did not have enough time to evaluate each of them.

6. Key Learnings

I believe the following are the most important insights I got out of this project:

1. It is not a good idea to hide the watermark in the perceptually insignificant portions of the image. For example, in the DFT domain, it is not really desirable to embed the watermark in high frequency coefficients. This is due to the fact that an unauthorized third party can easily clear those coefficients and hence wipe out the watermark without significantly affecting the quality of the image. Therefore, a watermark that is hidden in low frequency DFT components (of course, without significantly affecting the quality of the image) will be more robust.

2. Frequency domain techniques are usually more robust than spatial domain techniques due to their shift and translation invariant properties. Especially, use of DCT domain techniques increases the resilience of the watermarking algorithm against JPEG compression. In the DFT domain, it is more desirable to use phase modulation rather than amplitude modulation, because phase information contributes more to the image than amplitude information.

3. Exploitation of the properties of Human Visual System can increase the robustness and imperceptibility of the watermark. Especially techniques that would exploit HVS in DFT and DCT domains would lead to robust watermarking systems.

4. I familiarized myself with the current watermarking research and algorithms and saw the tradeoffs in watermarking by implementing and evaluating some of these algorithms. Overall, I believe this project was very interesting and I really learned a lot about watermarking. I also got a good chance to use the information I learned during this semester, such as DFT, DCT transforms, JPEG compression, linear filtering, edge detection, and thresholding.

References:

[1] M. D. Swanson, M. Kobayashi, and A.H. Tewfik, “Multimedia Data-Embedding and Watermarking Technologies”, Proceedings of the IEEE, Vol. 86(6) pp. 1064-1087, June 1998.

[2] S. Moller, A Pfitzmann, and I. Stirand, “Computer Based Steganography: How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense, At Best.”, in Information Hiding: First International Workshop Proceedings, vol. 1174 of Lecture Notes in Computer Science, pp. 7-21, Springer, 1996.

[3] W. Bender, D. Gruhl, and N. Morimoto, “Techniques for Data Hiding”, Proceedings of the SPIE 2420, Storage and Retrieval for Image and Video Databases III, pp. 164-173, 1995.

[4] A. Hanjalic, G.C. Langelaar, P.M.B. van Roosmalen, J. Biemond, and R.L. Lagendijk, Image and Video Databases: Restoration, Watermarking, and Retrieval, Elsevier, 2000.

[5] G. Depovere, T. Kalker, J.-P. Linnartz, “Improved watermark detection using filtering before correlation”, Proceedings of the 5th IEEE Conference on Image Processing, pp. 430-434, 1998.

[6] J.J.K. O Ruanaidh, S. Pereira, “A secure robust digital image watermark”, Electronic Imaging: Processing, Printing, and Publishing in Colour, SPIE Proceedings, May 1998.

[7] I.J. Cox, J. Kilian, T. Leighton, T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia”, Technical Report 95-10, NEC Research Institute, 1995.

[8] M. Kutter, F. Jordan, F. Bossen, “Digital Signature of Color Images Using Amplitude Modulation”, Proceedings of the SPIE Electronic Imaging, Storage and Retrieval for Image and Video Databases V, February 1997.

[9] J.J.K. O Ruanaidh, W.J. Dowling, and F. M. Boland, “Phase Watermarking of Digital Images”, Proceedings of the IEEE International Conference on Image Processing, pp. 239-242, September 1996.

[10] E. Koch and J. Zhao, “Towards Robust and Hidden Image Copyright Labeling”, in IEEE Workshop on Nonlinear Signal and Image Processing, pp. 452-455, October 1995.

[11] S. Katzenbeisser and F.A.P. Petitcolas, editors, “Information Hiding: Techniques for steganography and digital watermarking”, Artech House, 2000.

[12] A. Bovik, editor, “Handbook of Image and Video Processing”, Academic Press, 2000.