Digital Watermarking by DCT Coefficient Manipulation

by

Xiaodi Sun

Ph.D., University of British Columbia, 1998

M.Sc., Najing University, 1987

B.Sc., Najing University, 1984

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard

The University of British Columbia

August 2001

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
Vancouver, Canada

Date Aug. 20, 2001

Abstract

The advent of digital multimedia and worldwide area networks such as the Internet facilitates efficient distribution, reproduction and manipulation over networked information systems. To discourage the unauthorized copying and distributing of electronic documents, new tools are needed for their tracking and copyright enforcement. Digital watermark, which is an invisible mark embedded in a digital medium, for example a serial number or copyright information and can be retrieved even after attacks such as image processing and lossy compression, is one of the most promising ways for this purpose.

This thesis is concerned with analyzing and developing several digital watermarking schemes based on manipulating DCT coefficients of images. In the first scheme, a variant of the watermarking scheme proposed by Langelaar, et al. for JPEG/MPEG streams is implemented based on selectively discarding high frequency DCT coefficients. The watermark bits can be recovered by comparing the high frequency energy in difference DCT blocks. The second scheme embeds a watermark into the DCT coefficients of an image during the process of quantization; by introducing a self-reference pattern, the robustness of an watermark can be improved. In the third scheme, pairs of mid-frequency DCT coefficients are selected and modified to embed a watermark. The last scheme is built on the foundation of spread-spectrum communication. After presenting a generic spread spectrum based watermarking scheme, we derive the expectation and variance of the correlation between the investigated image and the watermark, from which the performance of our spread spectrum watermarking algorithm is analyzed and is found to be consistent with the experiment results.

The watermarking schemes can detect the embedded watermarks in the DCT domain without the help from the original images. Experiment results

show that the embedded watermarks in images can be retrieved successfully even after the JPEG compression and certain other image processing attacks. In addition, our spread spectrum watermarking scheme is capable of embedding multiple watermarks into an image and is robust to cropping attack as well as the JPEG compression with a very low quality factor.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Son Vuong. Without his advice and support, this thesis would never have been completed. The help from Vuong was not limited to the science; he also supported my personal goal. I also wish to thank Dr. Alan Wagner for his careful reading and insightful comments on my thesis.

XIAODI SUN

The University of British Columbia
August 2001

To my parents, LongGuan and XiaoLong, my wife, Yanping and my daughter, Amanda.

Chapter 1

Introduction

1.1 Watermarking and Steganography

In the last decade, there has been an explosion in the use of multimedia data. Computers and high rate digital transmission facilities are becoming less expensive and more widespread. Digital networks and CD-ROM provide an efficient cost-effective means of distributing digital media. However, all of these advanced technologies have made duplication of original artwork much easier with an unlimited number of copies. In order to protect intellectual property rights, new methods for signing and copyrighting digital data are in demand by artists and publishers. As a result, digital watermarking technology has become a very popular area of research.

The concept of digital watermarking is to sign images or other multimedia by introducing small changes that are imperceptible to the human eye but easily recoverable by a computer program. Generally, these small changes

represent the signature of the owner of the image, such as a serial number, or other identifications of the author. Another important requirement of a digital watermark is robustness. That is, it should be possible to retrieve the signature or watermark from an alternated image. Possible alternations of watermarked images include lossy compression, blurring, cropping, geometry transformation, etc. There alternations are called attacks.

Digital watermarking is an important and new sub-discipline of communication security. In an ideal world, communication may be secured by encrypting the traffic, however this is not always adequate in reality. For example, the company you are working for may not allow encrypted email, and even so, an encrypted email message between a known criminal and someone else not under suspicion yet does have obvious implications. This is where steganography comes into play. The word *steganography* derives from Greek language and means "cover writing". It simply takes one piece of information and hides it within another. So while cryptography is about *protecting* the content of messages, steganography is about *concealing* their very existence.

Digital watermarking is a technique of steganography which place a hidden copyright message in images, music, video, books, etc. Computer files and digital multimedia have redundant or insignificant areas of data, or holes. By taking advantage of these holes, we can replace them with hidden information. For example, a recording of a song might contain a plan of escaping from prison, and an image might contain a train whistle or a letter to a friend.

Steganographic techniques can trace their history back to antiquity. For

example, Gaspar Schott [41] explained how to embed information in music notes: each note represents a letter; David Kahn [12] explained how to use the acrostic method to put a name in the first letters of successive chapters of a book. First publication focusing on watermarking digital images were by Caronni [6] and Tirkel et al [47] in 1993. Since then, digital watermarking has received much attention from the research community and industry and played many important roles in a number of application areas.

1.2 A Generic Digital Watermarking Process

There are some common terminologies for digital watermarking which were proposed at the first international workshop on Information Hiding [29] in 1996. A generic digital watermarking process was stated as follows. Every watermarking system consists of the same generic building components: a watermark embedding system and a watermark detection system. Figure 1.1 shows the structure for the watermark embedding process.

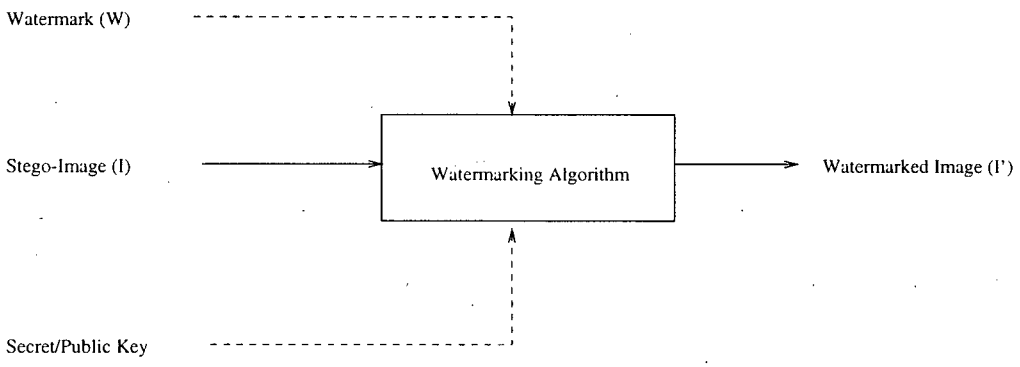


Figure 1.1: Generic digital Watermarking encoding process.

The input to the embedded scheme includes the watermark, the stego-image (also often referred as cover or host data) and optionally a secret or public key (usually a seed for a pseudo-random number generator). The watermark can be a serial number, copyright information or a logo of a company. Under the assumption that the watermarking algorithm is public and known by interested parties, the public or secret key is necessary to enforce the security of the watermark. The output of the watermarking embedding scheme is a modified, that is watermarked, image.

The generic detection process is shown is Figure 1.2. Using the public or secret key, the detection scheme can determine if there is a watermark in the test-image and recover it, or provide some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the test-image under inspection. The original image or the watermark may be used in the extraction process depending on the method.

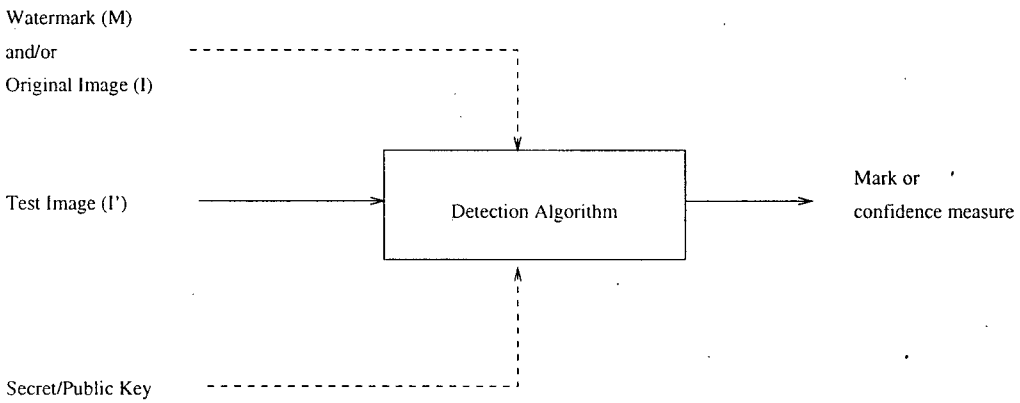


Figure 1.2: Generic digital Watermarking decoding process.

There are several types of robust copyright watermarking systems. Given

an image I, a watermark W and a key K, the embedding process can be defined as a mapping of the form: $I \times K \times W \rightarrow \hat{I}$ and is common to all marking methods. They are differentiated by the detection process as follows.

- **private watermarking** system requires at least the original image. There are two types of these systems. The first type of systems use the original image as a reference to find out where the watermark could be in the possibly distorted image $\hat{I}$ and then extract the watermark W from $\hat{I}$. The second type of systems also require the embedded watermark for extraction and just give a "yes" or "no" answer to the question: does the test-image $\hat{I}$ contain the mark W? $(\hat{I} \times I \times K \times W \rightarrow \{0,1\})$. It can be expected that this type of systems are more robust than the others since they convey very little information and more data can be used to embed the one bit mark. Examples of private watermarking include [7, 8, 16].

- **Semi-private watermarking** answers the same question as the second type of private watermarking, but doesn't require the original image for extraction. It is defined as $\hat{I} \times K \times W \rightarrow \{0,1\}$. The only practical application of the private and semi-private watermarks we can identify is in a court as evidence to prove ownership. Specifically, the number of the original image wants to distinguish among many apparently copies of the cover image in order to identify users who have given away illegal copies. Many of the currently proposed watermarking schemes fall into this categories (cf. [17, 42, 51]).

- **Public watermarking** requires neither the original image I nor the embedded watermark W. It is also called *blind* or *oblivious* watermarking, meaning that the embedded watermark can be read without prior knowledge of the cover image. Indeed such systems can extracts n bits of information (watermark) from the marked image: $\hat{I} \times K \rightarrow W$. Examples are given in [14, 20, 44, 45].

We consider the private watermarking schemes are of limited interest due to their narrow range of applications. Since the embedded watermark can only be recovered by one who owns the original, the embedded message can't be extracted by a user. For instance, a user's web browser would not be possible to extract and display a caption such as "do not copy" warning embedded in a download image. In addition, the need of the original image also make oblivious, batch extraction impractical. One might desire a web crawler or a search engine to automatically identify all illegal copies of anyone of the images belong to a particular photo archive, but this is not feasible for the private watermarking system. Even worse, the proof value of such systems is sometimes questionable since it is possible to construct an "original" image a posteriori to make any image appear to contain any watermark. So in this thesis, we are only interested in the public watermark schemes.

Another application of digital watermarking is for content and/or author authentication. An example of this scenario is images taken by a prestigious photographer using a digital camera. The images must be watermarked upon capture so that the clients can be sure that the images they want to

purchase have not been altered. Here the unrestricted distribution of copies of images is much less a concern than verifying an image's origin and content. This is an important issue in the protection of historical artwork and those used in courts of law as evidence.

It is critical that very slight changes to the image can be detected and localized, and it is not desirable for the watermark to remain in the image after any attacks on the images such as filtering. In contrast to copyright protection watermarks, which are also referred as *robust* watermarks, this type of watermarks are known as *fragile* watermarks. Although most of recent research is about robust watermarks, the issue of fragile watermarking, with focus on content authentication, remains a subject of active research (cf. [18, 51, 54]).

Different applications pose different requirements on watermarking schemes. However watermark imperceptibility is a common requirement and independent of the application purpose for all watermarking systems. The embedding process should not introduce any perceptible artifacts in the cover data, otherwise the commercial value of the images or other multimedia will depreciate. This is similar to designing lossy compression algorithms since the human visual/auditory system is used in both cases.

For all watermarking applications, except authentication watermarking, the robustness is one of the most important design issues because it determines the algorithm behavior toward data manipulation and signal processing operations on the host data. The robust watermarks should have these properties:

invariance to noise, covariance to geometry transformation and localization. Specifically, the following distortions and attacks should be taken into consideration:

- Noise (additive, multiplicative, lossy compression, etc.)
- Linear and nonlinear filtering (low pass, high pass, bandpass, etc.)
- Affine transformation (rotation, translation, scaling, shearing, etc.)
- Data reduction (cropping, clipping)
- A/D and D/A data conversion (print-scan)
- File format conversion (JPEG $\rightarrow$ GIF, H.265$\rightarrow$ MPEG-2)

The other watermarking design issues include: multiple watermarks, collusion attacks, trustworthy detection and automatic searching. After the owner of an image embedded a watermark in it and then sold it to another person. The second owner should also be able to embed his own watermark and extract it afterwards. The watermark should be characteristic of an author, but "collusion attacks" should not be able to detect the watermark by comparing several signals belong to the same author. As mention before, for the second type of private watermarking and semi-private watermarking, the embedded watermark has to be available in the detection process. The system detects if an image is "trustworthy" by verifying if the given watermark is present in the image under inspection. If the original watermark is not required, then the detection process can extract the embedded information and

such systems are for example useful for automatic searching on the Internet with a web crawler or intelligent agent. Here it might not only be of interest to find images, but also to clarify them through the embedded watermarks as their identification number.

1.3 Organization of this thesis

The goal of the thesis is to survey and develop digital watermarking schemes for data hiding in digital images for the purpose of copyright protection. The techniques implemented in this thesis research fall into two categories: DCT coefficient manipulation and spread spectrum techniques. The organization of this thesis is as follows.

Chapter 2 starts with an introduction into the field of digital image watermarking. There are two basic categories for image watermark encoding: spatial domain watermarking which embeds watermarks into the spatial domain of an image and spectral domain watermarking which embeds watermarks into the spectral domain of an image. So we first introduce the techniques of spatial domain watermarking including least significant bits, patchwork, spread spectrum techniques, etc. Then various digital watermarking schemes in DCT domain and wavelet domain are presented. Other miscellaneous watermarking algorithms such as fractal, Fourier-Mellin transform and several watermarking algorithms for image authentication will also be described.

In Chapter 3, a spatial domain watermarking scheme is implemented

using DCT coefficient removal technique. This technique is first proposed in Langelaar, et al [21], but our scheme overcomes its disadvantage that a watermark bit sometime can't be embedded depending on the local image properties. Two other related watermarking based on DCT coefficient manipulation are also presented, including quantizing and pairing schemes. The scheme based on quantization encodes a watermark into the DCT coefficients of an image during the process of quantization, and its robustness can be improved if a self-reference pattern is used. In the pairing scheme, pairs of mid-frequency DCT coefficients are selected and modified to embed a watermark; a key file is generated during the encoding phase and is used in the decoding phase to retrieve the watermark. Using this approach, it is possible that the original image is not touched and thus no image distortion occurs.

In Chapter 4, we start with an introduction to the basic idea of direct sequence spread spectrum. Then we present a generic spread spectrum based watermarking scheme. For our spread spectrum based watermarking scheme the watermark is modulated and spreaded in the 8×8 DCT domain as in the JPEG compression. we will also derive the expectation and various of the correlation between the investigated image the watermark, from which the performance of our spread spectrum watermarking algorithm is analyzed and is found to be consistent with the experiment results. The major advantage of this scheme is capable of embedding multiple watermarks into an image and is robust to cropping attack as well as the JPEG compression with a very low quality factor.

Finally, a summary of the major results in this thesis and the potential problems for future research are presented in Chapter 5.

Chapter 2

Background of Image Watermarking

This chapter presents background material on the state of the art for image watermarking. There are two basic modalities for image watermarking encoding: spatial domain techniques which yield spatial watermarks, and frequency domain techniques which yield spectral watermarks depending on the domain of watermark insertion. While most of the spatial watermarking schemes provide simple and effective ways for embedding an invisible watermark into the cover image, they are usually not robust to common image alternations. The frequency domain schemes first transform an image into the frequency domain using Fourier, DCT, wavelet, etc.) and then embed the information directly into the frequency coefficients of the image. The inversed coefficients form the watermarked image. Generally speaking, casting watermarks in the frequency domain can provide more protection under most signal processing and high

ratio compression attacks. It can be recognized that most of the current watermarking algorithms are based on some kind of spread spectrum modulation in the spatial, frequency or space-frequency domain. Specifically, small, pseudo-random changes are applied to selected coefficients in the spatial or spectral domain and are later on identified by correlation or correlation-like similarity measures.

2.1 Spatial domain watermarking

For most existing commercial products, the watermark is cast in the spatial domain (cf. [1, 7, 46, 47, 42]. Generally speaking, the watermark is embedded in the least significant bits (LSB) of image pixels. For example, in their 1993 publication entitled *Electronic Watermark*, Tirkel et al [47] proposed two watermarking methods for gray scale images. In their first approach, the watermark is in form of an m-sequence derived pseudo-noise code and is embedded in the least significant bits of the image pixels. To avoid introducing much visual distortion, the image is first compressed to 7 bits through adaptive histogram manipulation. This method is in fact an extension to simple LSB coding schemes in which we insert the code information directly into the LSB bit plane. The watermarking decoding is straightforward by comparing the LSB bit pattern with a stored counterpart. The second approach uses LSB addition for embedding the watermark, again in the form of an m-sequence derived code. The decoding process makes use of the unique and optimal auto-correlation function of m-sequence [23]. A modified version of this pa-

per was published in 1994 [42], which was the first publication that explicitly mentioned and hence defined the term "digital watermarking"

In [48], the idea of using m-sequence and LSB addition was extended and improved through the use of two dimensional m-sequences which resulted in more robust watermarks. Let X be a $N \times M$ gray-scale image, and a bipolar extended m-sequence from a small key file is uniquely associated with an owner. The sequence fills multiple 8×8 pixel blocks to eventually cover the entire original image. The collection of blocks forms the full watermark W. W is then arithmetically added to X to form the watermarked image, Y:

$$Y = X + W \ .$$

As part of the encoding procedure, the inner product between each watermark block and corresponding marked image block is computed. To verify a possibly forged image Z, the spatial cross-correlation function between Z and W is computed and compared with the previously stored value. A user defined threshold on the magnitude of the change determines whether a block is altered from the original one.

Another example is known as the *patchwork* [1]. In this method, n pairs of points (a_i, b_i) are selected randomly to hide bit 1 by increasing the brightness of a_i by one and decreasing the brightness of b_i by one simultaneously. Provided that the image satisfies certain statistic properties, the expected value of the sum of the differences between the a_i's and b_i's is given by $2n$,

$$S_n = \Sigma_{i=1}^{n} a_i - b_i = \begin{cases} 2n & \text{for watermarked image} \\ 0 & \text{for unwatermarked image} \end{cases}$$

See Figure 2.1 for the distribution of S_n. A second method presented in [1] is called *texture block coding*. The watermark is embedded by copying one image texture block to another area in the image with a similar texture. The watermark can be recovered by computing the autocorrelation function. A remarkable feature of this scheme is the high robustness to any kind of distortion, since both image areas are distorted in a similar way which means that the watermark recovery by autocorrelation still works. This scheme, however, requires that the image contain relatively large areas of texture; the techniques is also vulnerable to low pass filtering. So the transparency requirement comes at the expense of robustness.

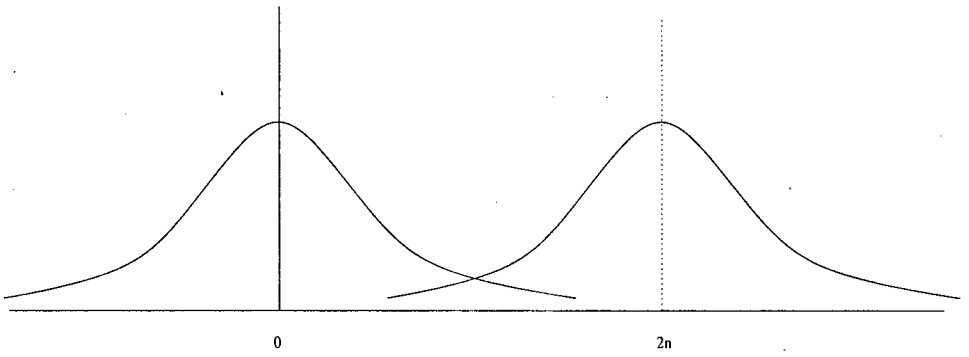


Figure 2.1: As n increases, the distribution of S_n shifts further to the right.

Similar idea as those by Bender et al [1] was proposed by Pitas et al [25, 30, 31]. Consider a $N \times M$ gray image $I = \{x_{nm}\}$ and assume that watermarks $S = \{s_{nm}\}$ is a binary pattern of size $N \times M$ where the number of ones equals the number of zeros. The original image I is first splited into two subsets of equal size $p = N \times M/2$ as follows

$$A = \{x_{nm} \in I\ ,\ \ s_{nm} = 1\ \}$$

$$B \;=\; \{x_{nm} \in I\,,\;\; s_{nm} = 0\,\}$$

The watermark S is superimposed by changing the elements of the subset A by the positive integer factor k, i.e., $C = \{x_{nm} + k\,,\;\; x_{nm} \in A\}$ and the signal image is given by $I_S = I \oplus C$. To verify if the image is watermarked, the mean $\tilde{c}$ of set C and the mean $\tilde{b}$ of B can be calculated first to obtain the test statistic

$$q = \frac{\tilde{c} - \tilde{b}}{\sigma_c^2 + \sigma_b^2},$$

where σ_c and σ_b are the standard deviations of set C and B respectively. Then the Null and the Alternative Hypotheses ([22])

$\mathbf{H_0}$: There is no watermark in the image ($q = 0$)

$\mathbf{H_1}$: There is a watermark in the image ($q = 1$)

can be applied to verify the presence of a watermark by comparing the test statistic with a pre-defined threshold. This scheme is robust to JPEG compression up to a compression ration of $4:1$ as well as sub-sampling and multiple watermarks.

In [20], amplitude modulation was used to embed a watermark into color images. Let s be a single bit to be embedded in an image $I = (R, G, B)$, $p = (i, j)$ is pseudo-random position within I depending on a secret key. The bit s is embedded by modifying the blue channel B at position p as

$$B_{ij} \leftarrow B_{ij} + (2s - 1)L_{ij}q,$$

where q is a constant determining the signature strength and L_{ij} is the luminance at P. In order to retrieval the embedded bit, a prediction $\hat{B}_{ij}$ of

the original value of the pixel is computed based on a linear combination of pixel values in a neighborhood around p. Then the difference δ between the prediction and the actual value of the pixel is taken

$$\delta = B_{ij} - \hat{B}_{ij} .$$

The sign of the difference δ determines the value of the embedded bit. Since the embedding and the retrieval functions are not symmetric, the correct retrieval is not guaranteed though it is very likely. To reduce the probability of false retrieval, the bit can be embedded multiple times at different locations. The method was shown to be resistant to both classical attacks, such as filtering, and geometrical attacks after determining what operations (translation, rotation, etc.) have been applied to produce the tampered image. Moreover, the signature can be extracted without the original image.

The spectrum spreading techniques used in RF communications [10, 43] was first employed by Smith, et al [44] in digital watermarking. In their modulation scheme, each bit b_i is represented by some basis function ϕ_i multiplied by either positive or negative one, depending on the value of the bit. The modulated message $S = \sum_i b_i \phi_i(x, y)$ is added pixel-wise to the cover image $N(x, y)$ to create the stego-image $D(x, y) = S(x, y) + N(x, y)$. The basis functions will always be chosen to be orthogonal to each other, so that the embedded bits do not equivocate. In addition, we assume that the basis functions are also uncorrelated with the cover image, although they are not always so in reality. If they were, we could hide our signal using arbitrarily little energy and still

be able to recover it later as follows

$$(D, \phi_i) = \sum_{x,y} b_i \phi_i(x,y) N(x,y) + \sum_{x,y} b_i \phi_i^2(x,y) \sim b_i \,.$$

Different spread spectrum schemes can be obtained by choosing different basis functions ϕ_i. In direct sequence spread spectrum, basis function ϕ_i is a constant G multiplied by a pseudo-random block of $+1$ and -1 values. Each block ϕ_i has a distinct location in the (x, y)-plane. The blocks ϕ_i are non-overlapping and therefore trivially orthogonal. They tile the (x, y) plane without gaps. The embedded bits can be recovered by demodulation with the original modulating function. This scheme is oblivious and robust to various noise attacks.

This modulation idea was further extended by Kutter [19]. He investigated various efficient ways to watermark digital gray and color images, based on the foundations of spread spectrum communication in the spatial domain. Two different watermark detectors are introduced, the alternative sign detector and the linear correlator detector. A pre-processing step is introduced prior to the watermark extraction process to increase the robustness of the scheme. Furthermore, it is shown that the use of M-ary modulation, instead of binary signaling, is advantageous in the context of digital watermarking. In addition a technique is developed to detect if an image under investigation is watermarked based on the theory of detection of weak signals in non-Gaussian noise, and the concept of self-reference is introduced to identify geometrical image transformations and hence provide a functionality which allows for the design of watermarking schemes resilient to geometrical alternation.

Beside spatial domain watermarking related to modulation, it is possible

to insert a watermark by modifying certain special characteristics of an image. Knox, et al [13] has developed a watermarking algorithm for half-tone image, which is based on the fact that many different half-tone patterns will produce a perceptually similar gray field in an image. By modifying the half-tone pattern of an image, a watermarking can be incorporated into the image and can be then recovered as follows: a transparent sheet with a certain half-tone pattern is overlaid on a printed version of the watermarked image. Upon sliding the testing sheet into alignment with the printed image, a visible watermark appears, but it is invisible in the printed image itself. Another method is to modify the geometry features of an image [24]. This method is based on a dense line pattern which is generated pseudo-randomly and represents the watermark. Using an edge detector, a set of salient points is obtained in the image. These points are then warped such that most of them are within the vicinity of lines. In the extraction process, the method verifies if a significant large amount of points are in the vicinity of lines.

2.2 Watermarking in DCT domain

A first efficient watermarking scheme in spectral domain was introduced by Koch, Burgett, Zhao and Rinfrey [5, 14, 15]. After the image is divided into square blocks of size 8×8 for which the DCT is computed, a pair of mid-frequency coefficients from a pseudo-randomly selected block determined by a secret key. The pairs are modified to embed a watermark bit such that the difference of them is either positive or negative depending on the bit value.

To survive JPEG compression, the quantization matrix is taken into account when altering the DCT coefficients. This method is oblivious.

A frequency domain method for digital watermarking of image proposed in [8] is also based on the idea of spread spectrum communication. The key insight of this work is the realization that in order for watermark to be robust, it must be embedded in the perceptually significant regions of the image despite the risk of protected fidelity distortions. In its most basic implementation, a watermark X consists of a sequence of normally distributed, zero-mean unit-variance random number V to produce the watermarked image V'. Three naturally formulae for watermarking insertion are

$$\begin{aligned} v_i' &= v_i + \alpha x_i\,, \\ v_i' &= v_i(1+\alpha x_i)\,, \quad \text{or} \\ v_i' &= v_i e_{\alpha x_i} \end{aligned}$$

where α is the watermark strength and the v_i's are the perceptually significant spectral components. Inversely transforming v_i' to form the watermarked image completes the encoding procedure. The authors propose an empirically derived value of 0.1 for α. The scheme can be generalized by introducing multiple scaling parameters α_i as to adapt to the different spectral components and thus reduce visual artifacts. To verify if a watermark is present in the image, one can evaluate the similarity between the recovered watermarked vector X^* and the original watermark vector X. The similarity of X and X^* is defined by

$$sim(X, X^*) = \frac{X \cdot X^*}{\sqrt{X^* \cdot X}}$$

where the recovered watermark vector X^* is calculated by using formulae

$$\begin{aligned} X^*(i) &= \frac{1}{\alpha}(v_i' - v_i)\,, \\ X^*(i) &= \frac{1}{\alpha}\left(\frac{v_i'}{v_i} - 1\right)\,, \quad \text{or} \\ X^*(i) &= \frac{1}{\alpha}\log(v_i'/v_i)\,. \end{aligned}$$

In this method, the original image is needed in decoding phase, so it isn't oblivious. If an image has not been watermarked with X, sim is distributed as a zero mean random variable. If X^* is slightly different from X (i.e., V is watermarked with X, although slightly altered), then $E(sim) >> 0$. A hypothesis test on sim determines if X is present in the image. Robustness tests showed that the method resists to JPEG compression, dithering, fax transmission, printing-photocopying-scanning, multiple watermarking and collusion attacks.

Another global method by modulating DCT coefficients is presented in [3]. One first calculates the DCT of the image and then sort them in the order of their absolute magnitude. A percentage of total energy, p, is defined to identify the largest n coefficients that makes up p percent of the total energy. The watermark sequence which is a one-dimensional bipolar binary sequence is added to all the AC coefficients in the list

$$v_i' = v_i + x_i\,.$$

The percentage p can be adjusted to trade off robustness and imperceptibility. The decoding phase first extracts X^* from the test image $\hat{V}$:

$$X^*(i) = \hat{V}_i - V_i\,.$$

Then the similarity between X and X^* can be used to verify X^*. Note that this method requires the original image to extract the watermark.

A similar method to [3] is given by Piva, et al [32]. The watermark consists of a pseudo-random sequence of M real numbers with normal distribution $X = \{x_i, \ldots, x_M\}$. The DCT coefficients of an entire original image I are reordered in the zig-zag fashion of JPEG. To decrease the chance of the watermark being perceptible, the first L coefficients are not marked and M coefficients starting at position $L + 1$ are selected to generate the vector $T = \{t_1, \ldots, t_M\}$. The watermark X is then embedded into T as follows:

$$t'_i = t_i + \alpha |t_i| x_i \,, \quad i = 1, \ldots, M \,,$$

where α is a user-defined scaling factor. The modified coefficients replace the non-modified coefficients before the intermediate image I' is reconstructed. Pixels in the watermarked image Y are linear combinations of the pixels in I and I' as follows:

$$y''_{ij} = I_{ij}(1 - \beta_{ij}) + \beta_{ij} I'_{ij} = I_{ij} + \beta_{ij}(I'_{ij} - I_{ij}) \,,$$

where β_{ij} is a weighting factor factor that takes account the characteristics of the human visual system. To detect the presence of a watermarked X in a test image I^*, the correlation Z between the possibly corrupted signed DCT coefficients T^* and the watermark is calculated as

$$z = \frac{X \cdot T^*}{M} = \frac{1}{M} \sum_{i=1}^{M} x_i t_i^*$$

and compared with a predefined threshold s_z. The threshold s_z is evaluated

directly on the marked image and given as

$$s_z = \frac{\alpha}{3M} \sum_{i=1}^{M} |t_i^*| .$$

Experiment results demonstrate that the watermark is robust to several image processing techniques and geometry distortions.

Liu, Podilchuk and Zeng [33, 55] introduce perceptual watermarking using the Just Noticeable Difference (JND) to determine an image dependent modulation mark. The watermark modulation in either DCT or wavelet transform domain can be described as

$$X_{u,v}^* = \begin{cases} X_{u,v} + J_{u,v} w_{u,v} , & X_{u,v} > J_{u,v} , \\ X_{u,v} , & \text{otherwise,} \end{cases} \tag{2.1}$$

where $X_{u,v}$ refers to the frequency coefficients of the original image samples $x_{i,j}$, $X_{u,v}^*$ refers to the watermarked image coefficients, $w_{u,v}$ is the sequence of watermark values, and $J_{u,v}$ is the computed just noticeable difference based on visual models. Watermark detection is based on the correlation between the difference of the original image and the image under inspection, the watermark sequence and watermark sequence. Experiments showed that the watermark is extremely robust to JPEG compression, cropping, scaling additive noise, gamma correction and print-scanning. This scheme requires the original image in detection. A revision is also proposed to avoid the use of the original image in the verification procedure. In this technique, it is assumed that the original image has already been JPEG compressed. A subset called feature vector is denoted by $\{X_D\}$. If a DCT coefficient x_D is larger that half of its

corresponding quantization table value, Q, it is included in $\{X_D\}$:

$$x_D \in \{X_D\}\ , \quad \text{if } x_D > \frac{Q}{2}\ .$$

The watermark w is a sequence of $N(0,1)$ random numbers that is added to $\{X_D\}$:

$$y_D = x_D + w\ , \quad x_D \in \{X_D\}\ ,$$

$$y_D = x_D\ , \quad \text{otherwise}\ .$$

The IDCT of Y_D forms the marked image Y. To verify the presence of w in a test image Z, the feature vector $\{Z_D\}$ is obtained. A correlation measure $x = c$ is found between $\{Z_D\}$ and w:

$$c = \frac{\mu\sqrt{n}}{\sigma}\ ,$$

where μ and σ are the mean and variance of the point-wise multiplication of $\{Z_D\}$ and w. c will be distributed roughly according to $N(0,1)$ if w is not in $\{Z_D\}$, otherwise it will be much higher.

2.3 Watermarking in Wavelet Domain

Various wavelet based schemes have been proposed [16, 17, 53, 50]. The difference between them usually lies in the way the watermark is weighted in order to decrease visual effects.

A robust digital image watermarking method using wavelet-based fusion is proposed in [16]. This method assumes that the binary watermark is of

length N_ω and consists of elements from the set $\{-1, 1\}$. The watermark is embedded into the detail wavelet coefficients of the host image with the use of a key. The number of ones in the key must be greater than or equal to the size of the watermark. The watermark values can be repeatly embedded in different coefficients if the length of the watermark is less than the number of ones in the key. The encoding procedure has three stages. Stage I computes the L-th discrete wavelet decomposition of the host image to produce a sequence of $3L$ detailed images, corresponding to horizontal, vertical and diagonal details at each of the L resolution levels, and one gross approximation. Stage II considers each resolution level l and coefficient location (m, n). The detailed coefficients are sorted in ascending order ($f_{k_1,l}(m,n) \leq f_{k_2,l}(m,n) \leq f_{k_3,l}(m,n)$. If the associated value of the key is one, then some middle wavelet coefficient must be quantized appropriately to embed the binary watermark (See Figure 2.2, where Q is a user pre-defined variable and $\Delta = \frac{f_{k_3,l}(m,n) - f_{k_1,l}(m,n)}{2Q-1}$). Stage III forms the watermarked image by the inverse wavelet transformation of the fused image components. The detection is performed by estimating the watermark bit value from the relative position of the middle wavelet coefficient. It was shown that this method is robust to some common image distortions, such as JPEG compression, additive noise and linear filtering.

Xia, et al [53] implemented a multi-resolution watermark for digital images. In the encoding part, an image is first decompose into several bands with a pyramid structure as shown in Figure 2.3 and then a pseudo-random sequence (Gaussian noise) is added to the large coefficients not in the lowest

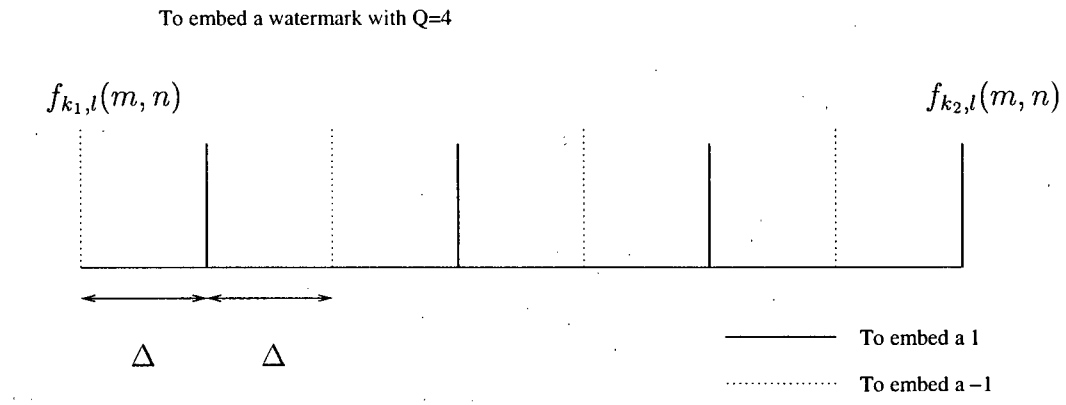


Figure 2.2: Quantization process to embed a watermark. The middle wavelet coefficient $f_{k_2,l}(m,n)$ must be quantized to the nearest vertical bold bar to embed a one and to the nearest dotted line to embed a negative one.

resolution. Let $y(m,n)$ denote the discrete wavelet transformation (DWT) coefficients not in lowest frequency band. The Gaussian noise $N(m,n)$ with mean zero and variance one is added to $y(m,n)$:

$$\tilde{y}(m,n) = y(m,n) + \alpha y^2(m,n)N(m,n)\,,$$

where α is the watermark strength. The DWT coefficients at the lowest resolution are not changed. The inverse DWT forms the watermarked image. The decoding method is hierarchical and the original image is required. One first decomposes the test image and the original one with DWT into four bands (LL1, LH1, HL1, HH1), then calculates the cross correlation between the signature added in HH1 band and the difference of the DWT coefficients in HH1

bands of the test and the original images. If there is a peak in the cross correlation, the signature is then detected. Otherwise, compare the signature added in HH1 and LH1 bands with the difference of the DWT coefficients in the corresponding bands. If there is a peak, the signature is detected. Otherwise, consider the signature added in the HH1, LH1, HL1, and so on. An important advantage of this scheme is that the watermarking method has multi-resolution characteristic and is hierarchical. In the case when the received image is not altered significally, the cross correlation with the whole size of the image may not be necessary, and thus much of the computation can be saved. This scheme is robust to some common image distortion, such as wavelet transform based image compression and image half-toning.

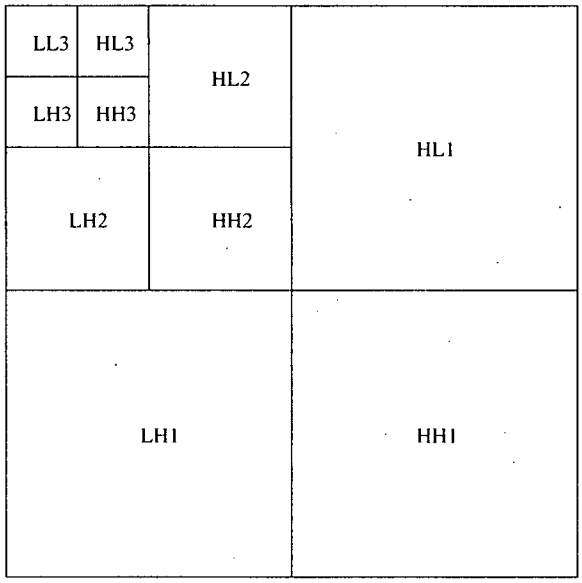


Figure 2.3: DWT pyramid decomposition of an image.

The scheme in [53] fails under a DCT-based compression attack (e.g. JPEG), since JPEG quantization table sets most coefficients of high frequency

components in each block to zero so that the watermark cast in these coefficients is lost. To overcome this problem, a new scheme to search perceptually significant wavelet coefficients for effective digital watermark casting is proposed in [50]. An adaptive watermark hiding method is first developed to determine significant wavelet sub-bands and to select a couple of significant wavelet coefficients in these sub-bands. Then a blind watermark retrieval technique that can detect the embedded image's watermark in the wavelet domain without the help from the original image is described. The basic idea in this blind watermarking algorithm is to truncate selected significant coefficients to some specific values. Experiments showed that the cast watermark can be successfully retrieved after various attacks including signal processing, geometry transform, noise, JPEG and wavelet compression methods. With the help of original image, the watermark can be detected after more serious attacks.

2.4 Miscellaneous watermarking algorithms

There are other miscellaneous watermarking algorithms (cf. [9, 18, 35, 37, 40, 38, 39, 49], etc.).

One of them is related to spatial domain watermarking schemes based on fractal image compression proposed by Puate and Jordan [35]. In general term, a fractal coder exploits the spatial redundancy within the image by establishing a relationship between its different parts. They described a way to use this relationship as a means of embedding a watermark. Specifically, the original image is divided into square block R_b, called range blocks, and

similarly, into square blocks D_b, called domain blocks. The domain blocks are larger than range blocks. The goal of the encoding algorithm is to establish a mapping in such a way that any R_b can be expressed as a set of transformations to be applied on a particular D_b. For each range block R_{b_j}, the mapping function consists of a vector V_j, which has its original in R_{b_j} and points to the corresponding D_{b_j} which becomes its matching block (M_{b_j}), and an appropriate transformation T_j which minimizes the difference between the range block R_{b_j} and the mapped domain block. Decoding is complished by iterating over the coded mapping function using any initial image. The partition of domain blocks where the search is performed is commonly taken as a square region surrounding the R_{b_j}, denoted by LSR (local searching region). Signing an image, consists of a coding-decoding process with variable searching regions. Consider two different LSR, A and B (see Figure 2.4), and a third one, C, defined as their union. To sign a bit s_i, a range block is randomly selected, and denoted by $\{R_b\}_j$. To sign one, R_{b_j} is coded by searching $\{M_{b_j}\}$ in region $\{A\}_j$. To sign zero, R_{b_j} is coded by searching $\{M_b\}_j$ in region $\{B\}_j$. Otherwise, $\{R_{b_j}$, which is not signed, is coded by searching $\{M_{b_j}\}$ in $\{C\}_j$. The rule to decide if a range block has been signed with a zero, one, or not signed, is determined by examining if V_j belongs to region A_j, B_j or C_j. The algorithm was tested against JPEG compression and showed good robustness down to a compression quality of 50%. A drawback of this technique is the slow speed due to the fractal compression.

Ruanaidh, et al [38, 39] proposed a phase based method of conveying

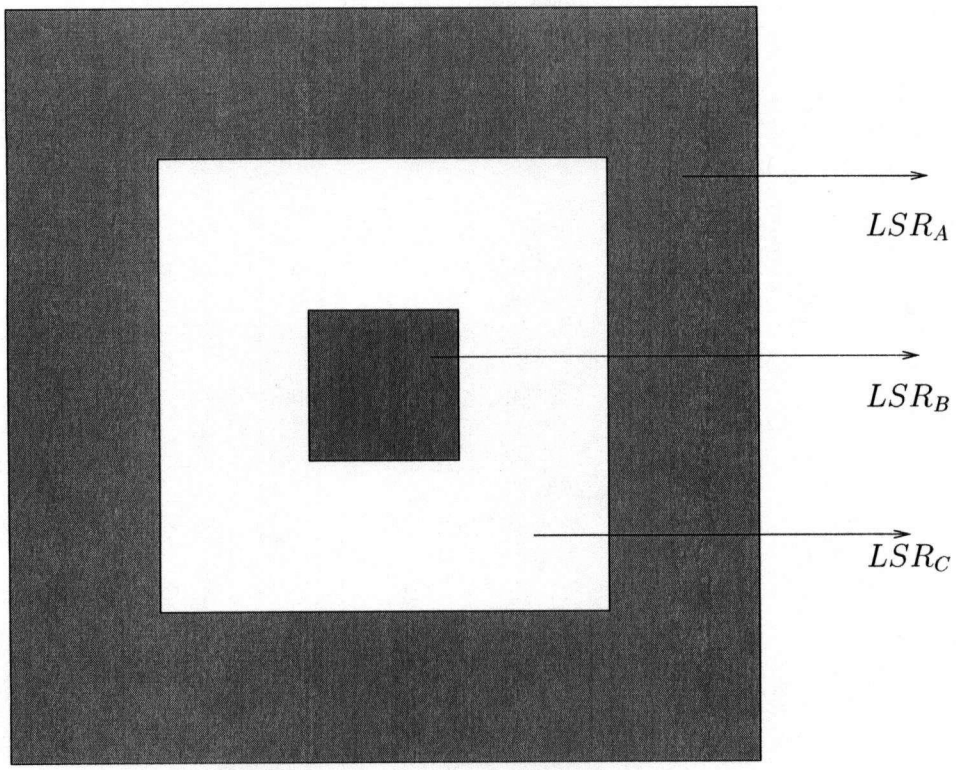


Figure 2.4: A range block, its LSR_A and its LSR_B. LSR_C is defined as their union.

the watermark information in gray scale digital images. To embed a bit, the phase of a selected coefficient $F(k_1, k_2)$ of an $N_1 \times N_2$ discrete Fourier transform (DFT) is modified by adding a small value δ:

$$\langle F(k_1, k_2) \leftarrow \langle F(k_1, k_2) + \delta \,.$$

The condition that the image be real imposes the following additional modification

$$\langle F(N_1 - k_1, N_2 - k_2) \leftarrow \langle F(N_1 - k_1, N_2 - k_2) - \delta \,,$$

i.e., the phase must satisfy negative symmetry. A DFT coefficient is marked only if its relative power is above a given threshold. Two distinct methods for watermark detection were described. Assuming the original image is available,

the first one is simply to compare the phase. The second one which doesn't require the original is to pre-quantize the original phase prior to encoding and use the deviations from these quantized states to convey information. Experiments showed this scheme survives 15:1 JPEG compression.

Another publication by Ruanaidh and Pun [40] proposed a Fourier-Mellin transform-based watermarking method which is robust to any combination of rotation, scale and translation transformations. The key idea is to obtain an invariant of an image, which is unaffected by these transformation. The watermarking process is described in Figure 2.5. The first step is to perform a DFT on an image. One of the DFT properties is that spatial shifts affect only the phase shifts in the frequency domain, but not the amplitude. So keeping only the amplitude for further processing makes the image translation invariant. The second step is to achieve a rotation and scaling invariance by mapping the amplitude from the Cartesian grid to a log-polar grid defined as

$$x = e^{\mu} \cos\theta \,,$$

$$y = e^{\mu} \sin\theta \,,$$

where $(x, y) \in R^2$, $\mu \in R$, and $0 \leq \theta < 2\pi$. It is easy to observe that for every point (x, y) there exists a corresponding point (μ, θ) and in this new coordinate system scaling and rotation are converted to a translation of μ and θ coordinates respectively. So one can implement a rotation and scale invariant by applying the DFT of the log-polar Map (LPM) and keeping only the amplitude. Taking Fourier transform of a LPM is equivalent to computing the Fourier-Mellin transform. These two steps results in a rotation, scale and

translation (RST) invariant in which a watermark may be safely embedded. After watermark insertion the inverse transform of the previous two step yield a watermarked image. The watermark takes the form of a 2-dimensional spread spectrum signal in the RST transformation invariant domain. This scheme resists to the JPEG compression and is the first published one which was especially designed to resist to geometry attacks.

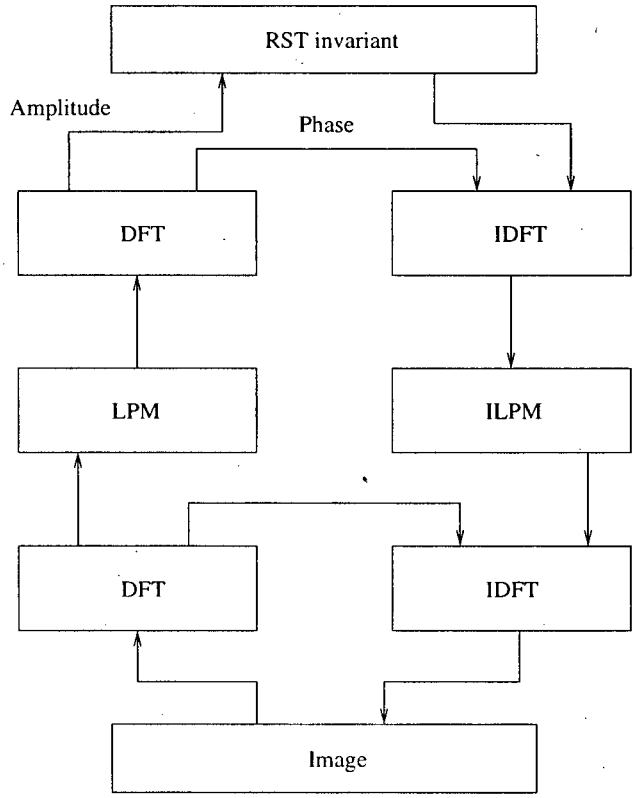


Figure 2.5: Rotation, scale and translation invariant scheme.

An earlier watermarking algorithm, for the purpose of image authentication and tamper detection, is known as the check-sum technique [49]. It is formed from the checksum value of the seven most significant bits of all pixels

in an image. This method randomly selects the locations of pixels to contain one bit of the checksum. The last bit of each selected pixel is changed to equal the corresponding checksum bit. To verify if a test image is authentic, the checksum of the image is computed and compared to the values in the locations where the checksum is embedded. Any discrepancy means that the image is not an exact copy the original.

Another fragile watermarking scheme for authentication was proposed in [18]. There are three main stages to the watermark embedding procedure. The first stage computes the L-th level discrete wavelet decomposition of the host image to produce a sequence of $3L$ detail images, corresponding to the horizontal, vertical and diagonal details at each of the L resolution levels, and a gross approximation at the coarsest level. In the second stage, the watermark bit stream is embedded by modifying selected wavelet coefficients through an appropriate quantization procedure. The selection of the coefficients is random and well-spread spatially and throughout each resolution level. In the final stage, the corresponding L-th level inverse wavelet transform of the marked image components is computed to form the tamper-proofed image. During watermark extraction, the L-th level discrete wavelet transform is applied to the given image and a quantization function is applied to the key-determined coefficients to extract the watermark value $\tilde{w}_i$. To assess the extent of tampering, the tamper assessment function (TAF) is defined by

$$TAF(w, \tilde{w}) = \frac{1}{N_w} \sum_{i=1}^{N_w} w_i \oplus \tilde{w}_i \,,$$

where w is the true watermark, $\tilde{w}$ is the extracted watermark, N_w is the length

of the watermark, and $\oplus$ is the exclusive-OR operator. The value of TAF is between 0 and 1. The presence of tampering is determined by comparing TAF with a threshold.

Chapter 3

Watermarking algorithms by DCT coefficient removal

In this chapter, we first implement a variant of the watermarking scheme proposed by Langelaar, et al. in [21]. In their paper, they proposed a watermarking algorithm for JPEG/MPEG streams that is based on selectively discarding high frequency DCT coefficients. They used a statistic model to derive the probability that a label bit can't be embedded and this model is used for maximizing the robustness against re-encoding and for developing adequate error correcting codes for the label bit string. A disadvantage of this scheme is that we sometime can't embed a label bit depending on the local image properties. The reason is that we can't guarantee that an energy difference between the *lc*-subregions in either positive or negative. We here propose a scheme such that this difference is always away from zero by a predefined threshold. Consequently, a label bit can always be embedded not depending

on the image properties. We further implement two other related watermarking schemes based on DCT coefficient manipulation, including quantizing and pairing schemes.

The organization of this chapter is as follows. In § 3.1, we first introduce the watermarking algorithm based on discarding DCT coefficients proposed in [21] and a modification of this scheme such that a label bit can always be encoded and decoded correctly. Then we describe the quantizing scheme and pairing scheme in § 3.2. Finally, experiments and analysis are given in § 3.3.

3.1 Watermarking scheme based on discarding DCT coefficients

Langelaar, et al. have proposed in [21] a watermarking algorithm for JPEG/MPEG streams that is based on selectively discarding high frequency DCT coefficients in the compressed data stream. The watermark bits are encoded in the pattern of DCT blocks in which high frequency DCT coefficients are removed.

Specifically, we can first represent a watermark or label as a label bit string L consisting of label bits L_j $(j = 1, 2, \ldots, l)$. This label bit string is embedded in an JPEG still image or in the I-frame of an MPEG video stream bit by bit. To provide better security, all 8×8 DCT-blocks are shuffled randomly before encoding as shown in Figure 3.1. Each bit out of the bit string is embedded in a label bit-carry-region, or *lc*-region, in a shuffled image or a shuffled I-frame. The size of a *lc*-region, i.e., the value of n, determine

the maximal number of label bits that can be encoded in the image. The larger the size of n, the smaller the maximal number of label bits that could be embedded, but the more robust the embedded watermark.

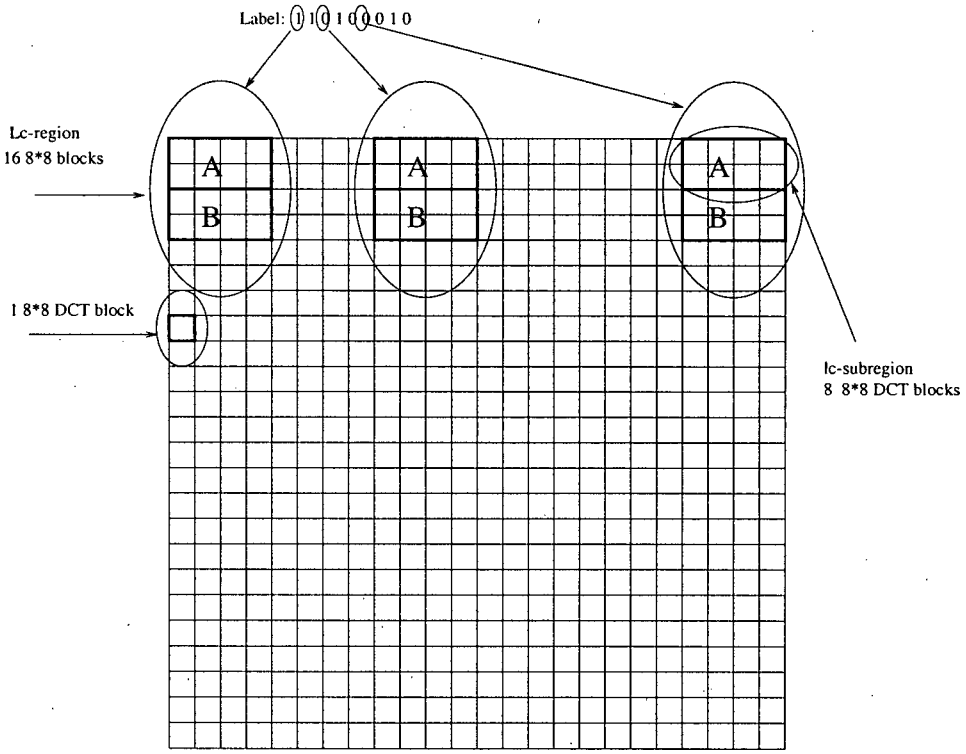


Figure 3.1: Bit positions and block definitions in a still image or an I-frame of a video stream

A label bit is embedded in a *lc*-region by introducing an "energy" difference between the high frequency DCT-coefficients of the top half of the *lc*-region (denoted by *lc*-subregion A) and the bottom half (denoted by B). The energy is computed over a subset of zip-zag scanned DCT-coefficients indicated by $S(c)$:

$$S(c) = \{i \in \{0, \ldots, 63\} | i > c\} \ ,$$

where c is the cut-off point, indicating that the DCT coefficients after c will be discarded. The selection of a suitable cut-off point c is essential for the robustness and the visibility of the watermark. The larger the cut-off points are chosen, the less distortion the watermark embedding will introduce. But this will decrease the robustness of the watermark. The energy in lc-subregion A is then defined as

$$E_A(c, n, Q_{jpeg}) = \sum_{b=0}^{n/2-1} \sum_{i \in S(c)} \left([\theta_{ib}]_{Q_{jpeg}} \right)^2$$

Here θ_{ib} denotes the i-th non-weighted DCT coefficient in the b-th block of the lc-region A (see Figure 3.2). The notation $[\cdot]_{Q_{jpeg}}$ indicates that, prior to the calculation of E_A, the DCT-coefficients are re-quantized using the standard JPEG quantization procedure with quality factor Q_{jpeg}. The energy E_B in lc-subregion B can be defined similarly.

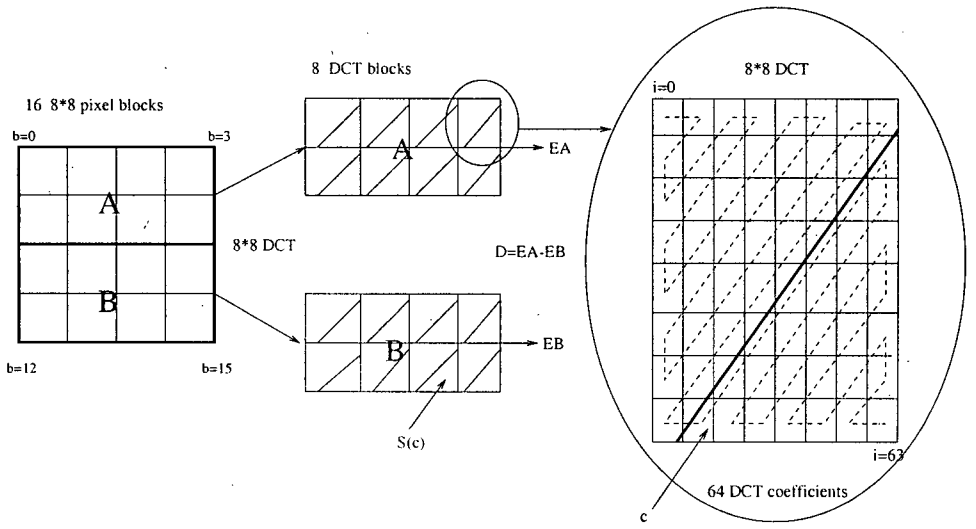


Figure 3.2: Calculating the energy difference D.

Now define the energy difference between the lc-subregions A and B as

$$D(c, n, Q_{jpeg}) = E_A(c, n, Q_{jpeg}) - E_B(c, n, Q_{jpeg})$$

Then the value of a label bit can encoded as the sign of the energy difference D. A label bit 0 is defined if D is positive and bit 1 is defined if D is negative. So we can embed a label bit string by manipulating the energy difference D. If label bit 0 must be embedded, all energy after the cut-off point c in the DCT-blocks of lc-subregion B is eliminated by setting the corresponding DCT-coefficients to zero, yielding

$$D = E_A - E_B = E_A - 0 = +E_A$$

If label bit 1 must be embedded, all energy after the cut-off point in the DCT-blocks of lc-subregion A is eliminated, yielding $D = -E_B < 0$.

There is one obvious problem which may occur in the previous algorithm. That is, what if the energy difference is zero? As a simple example, for a constant luminance gray image, the energy difference is always zero no matter what the value of cut-off point or the value of n and Q_{jpeg} you use. To minimize the possible failure of the watermarking scheme, Langelaar, et al. [21] have used a statistic model to derive the possibility that a label bit cannot be embedded. The resulting model is used for maximizing the robustness against re-encoding and for developing adequate error correcting codes for the label bit string.

3.1.1 A modified watermarking scheme

While it is true that the statistic model can reduce the label bit error probability, it is still possible that the failure will occur during watermark encoding or decoding due to a very small energy difference. To overcome this problem, we propose a variant of the watermarking scheme in [21] by introducing some additional noises in the high frequency DCT coefficients such that the energy difference can always be kept away from zero by some threshold, not depending on the local image properties.

Let τ be a threshold of the energy difference, i.e., we require that energy difference between lc-subregions A and B is greater than τ. Since we set the high frequency DCT coefficients zero in one lc-subregion (say A), we need that the energy in the other lc-subregion (say B) be greater than τ. There are $\frac{n}{2}$ DCT blocks in lc-subregion B, so we can add an amount of $2\tau/n$ energy into each of these blocks if $E_B < \tau$. To do so, assuming the cut-off point is c, we can increase each high frequency DCT coefficient after c by

$$QTable[u]\sqrt{\frac{\tau}{n/2(63-c)}} \quad (u = c+1, \ldots, 63) .$$

Here $QTable$ is the quantization table determined by quality factor Q_{jpeg} using the standard JPEG quantization procedure. It is easy to show that by discarding high frequency DCT coefficients in one lc-subregion and introducing some amount of noise in the high frequency DCT coefficients in the other lc-subregion, it can be guaranteed that the energy difference D is greater than the threshold τ. Thus a label bit can always be embedded.

3.2 Algorithms based on quantization and coefficient comparison

In addition to watermarking by DCT coefficient removals, we propose two other schemes based on quantization and comparison of DCT coefficients in a progressive transmission scheme.

The first algorithm embeds the watermark bits into an image by modifying the rounding rule for the quantized coefficients such that the resulting coefficients are odd or even, depending on the values of the watermark bits. Specifically, as in § 3.1, the image is first divided into square blocks of size 8×8 for which the DCT is computed. Each bit out of the label bit string corresponds to a label bit-carrying-region (lc-region) in a shuffled image. The mid-frequency coefficients of each 8×8 DCT block is indicated by $\tilde{S}$ defined by

$$\tilde{S}(c_m, c_M) = \{i \in \{0, 1, \ldots, 63\} \mid c_m \leq i \leq c_M\},$$

where c_m and c_M are the margin coefficients between which the zig-zag scanned DCT coefficients will be modified to embed a watermark bit. The parameters of c_m and c_M are to be selected to obtain a trade-off between perceptual invisibility and robustness to image processing techniques. We perform quantization the DCT coefficients which are real number in the following manner. Let θ_{ib} be the i-th non-weighted DCT coefficient in the b-th block of a lc-region, L_j be the corresponding watermark label bit and $\triangle \equiv \alpha QTable[i]$, where $QTable[i]$ is the i-th element of the quantization table determined by the quality factor

Q_{jpeg} and α is a scale parameter. Assume the coefficient θ_{ib} satisfies

$$r\triangle \leq \theta_{ib} \leq (r+1)\triangle\ ,\quad r = 0, \pm 1, \pm 2, \ldots\ .$$

Then in order to embed the label bit L_j, we quantize θ_{ib} for $i \in \tilde{S}(c_m, c_M)$ and $0 \leq b < n$ as follows:

$$\theta_{ib} = \begin{cases} r\triangle & \text{if } L_j = 0,\ r \text{ odd} \quad \text{or } L_j = 1,\ r \text{ even} \\ (r+1)\triangle & \text{if } L_j = 1,\ r \text{ even} \quad \text{or } L_j = 1,\ r \text{ odd.} \end{cases}$$

The inverse of the quantized DCT coefficients forms the watermarked image.

To extract the label bit L_j, which corresponds to a lc-region, we first calculate the DCT coefficients θ_{ib} ($c_m \leq i \leq c_M$ and $0 \leq b < n$) of the blocks in the lc-region. Then we identify whether the quantized coefficient $[\theta_{ib}]_{Q_{jpeg}}$ is close to an odd number or an even number by checking if $\lfloor \frac{\theta_{ib}}{\triangle} + 0.5 \rfloor$ is odd or even, where $\lfloor x \rfloor$ represents the largest integer less than or equal to x. Finally, we calculate L_j as follows:

$$L_j = \begin{cases} 1 & \text{if more than half of } [\theta_{ib}]_{Q_{jpeg}} \text{ are even} \\ 0 & \text{otherwise} \end{cases}$$

In order to enhance the performance of the watermark extraction process, we can introduce a concept called self-reference. This concept embeds an known pattern into the multiple mid-frequency coefficients with the goal of indicating whether the recovered bits are trusted. For example, in the encoding stage, we can make the quantized coefficients $[\theta_{ib}]_{Q_{jpeg}}$ even for $i = c_m, c_m + 2, c_m + 4, \ldots, c_M$, assuming the difference between c_m and c_M is even. And the label bit L_j is embedded in the rest coefficients. In the decoding

stage, only those DCT coefficients $[\theta_{ib}]_{Q_{jpeg}}$ whose two direct neighborhoods are even (i.e., only when the neighborhoods are correctly recovered) are taken into account. When their neighborhoods are not even, i.e., they are corrupted, we don't expect the DCT coefficients θ_{ib} can be trusted. So those coefficients are discarded in the decoding stage. Experiments showed that this approach often gives better robustness to JPEG image compression.

The second watermarking algorithm is totally different from the other watermarking schemes. Usually, a watermark is embedded by changing data portion of an image in such a way that the watermark is invisible and can be retrieved by a decoding algorithm. Our algorithm tries to remember some characteristics of an image depending on the watermark in a separate file and tries to keep the original image intact. Specifically, from a pseudo-randomly selected DCT block, a pair (p_0, p_1) of mid-frequency coefficients in $\tilde{S}(c_m, c_M)$ is selected such that the difference of them is maximized. The goal of this maximization is to provide better robustness to watermark attacks. If this maximum is still less than a pre-defined threshold τ, then this pair is modified as follows:

$$p_0 = p_0 - \frac{\tau}{2}\,, \quad p_1 = p_1 + \frac{\tau}{2}\,,$$

where we assume $p_0 \leq p_1$. To embed a bit one, we write the positions of the zig-zag scanned DCT coefficients p_0 and p_1 into a file; to embed a bit zero, the positions of p_1 and p_0 are written. During the recovery process, the file is used to locate the positions of the pairs and the watermark bits can be determined by checking if the difference between the pairs is positive or negative. In this

approach, it is often the case that the original image is untouched.

3.3 Experiment results

3.3.1 Watermarking scheme with additional noises

In this sub-section, we implement the watermarking algorithm introduced in § 3.1.1 using C and evaluate the following performance of our watermarking algorithms:

- the robustness against attacks such as JPEG compression, and other image processing
- the visual impact of the watermark
- the size of the watermark.

These performance factors are controlled by four parameters:

- Q_{jpeg}, which is the minimum JPEG quality setting up to which the watermark is resistant against re-encoding,
- n, which is the number of DCT blocks used to embed a single watermark bit,
- c, which represents the lowest DCT coefficient that we permit to be discarded during the label embedding,
- τ, which is a threshold that the energy difference must be greater than.

Now we analyze how these parameters affect the performance of the watermarking algorithm. In all of our experiments, we assume the watermark to be encoded is a string *Secret is "CPSC 549"!* We first analyze the number n of blocks for encoding a label bit by running the program with $c = 32, Q_{jpeg} = 75$ and $\tau = 0.5$. Let *length* be the number of chars in the watermark. Then the image has to have $8 \times length \times n$ DCT blocks to embed the watermark. Generally, if we select a larger number value of n, then the watermark will be more robust since the label bits are embedded in more blocks and thus less fragile. For example, if we take $n \doteq 16$, then we can retrieve the watermark correctly; on the other hand, if we take $n = 2$ or $n = 4$, then there is one bit error in the retrieved watermark. This number n of blocks can also affects the visibility of a watermark. To see this effect, we set $\tau = 10$ and run the program with $n = 2$ and $n = 16$ to obtain the watermarked images shown in Figure 3.4. We can notice that the distortion is less noticeable in the image with $n = 16$ than that with $n = 2$ since the energy (for a larger n) is distributed in more DCT blocks so that the noise in each DCT block is smaller. Please note that we cut a portion of the image out (see Figure 3.3) such that you can see the embedded noise more clear. For the same purpose, we set the threshold τ very large (10, *in real world, you can use some* $\tau < 0.01$).

We then study the relationship between the cut-off point c and the performance of the watermark. Since there are two types of blocks in an image, the situation is little bit complicated. For the first type of blocks where we may put some noise in them such that energy difference is greater

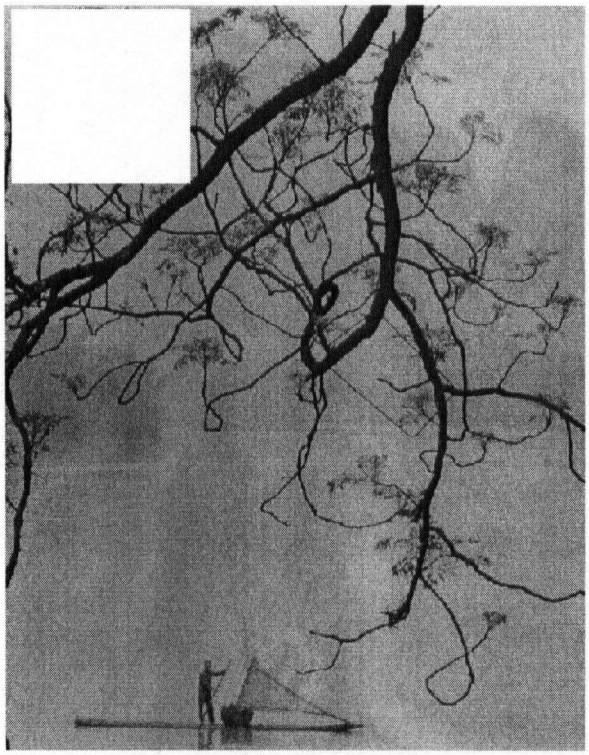


Figure 3.3: Plot of the original image, where we cut a portion of it such that any embedded noise will be more perceptible.

than a threshold, if c is small, then the distributed energy in each block may be too small to survive the rounding operation to the nearest integer in the DCT transformation. In this case, the embedded watermark is usually less visible, but is not very robust. On the other hand, if c is large, then the watermark is generally more visible, but is also more robust. See Figure 3.5 for comparisons.

For the second type of DCT blocks, where we discard the high frequency DCT coefficients, small value of c can yield more robustness, but less invisibility. On the other hand, large value of C can give less robustness, but also less distortion of the image. See Figure 3.6 for some comparisons. Therefore, smaller values of c will make the noise in those blocks discarding DCT

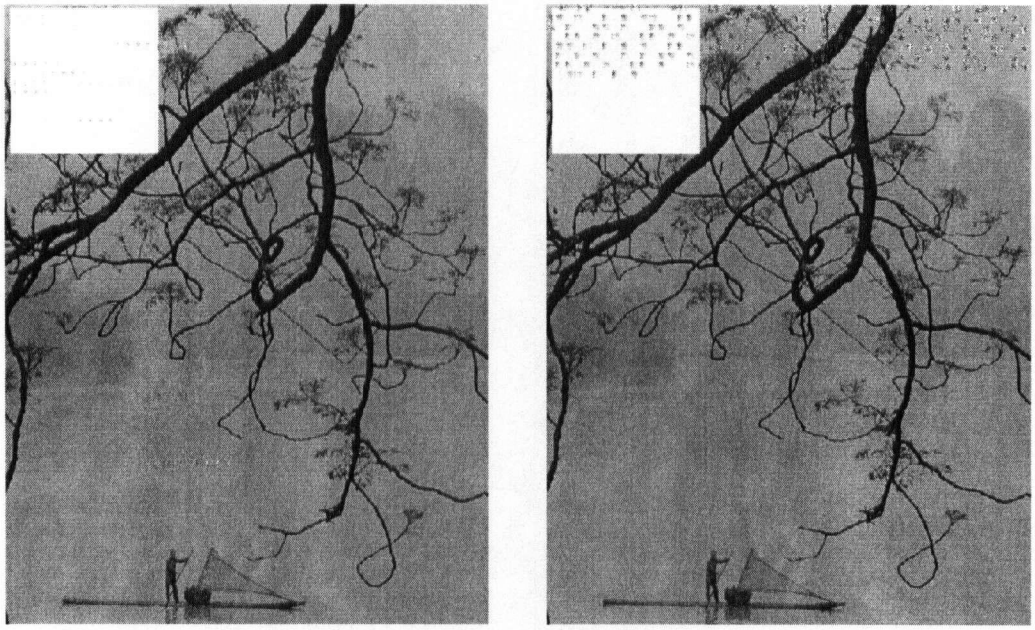


Figure 3.4: Plot of the watermarked images with $n = 16$ (left) and $n = 2$(right).Here $c = 32$, $Q_{jpeg} = 75$ and $\tau = 10$. Noises are more noticeable for $n = 2$ than for $n = 16$.

coefficients more perceptible and those blocks putting additional energy less perceptible; for larger values of c, the effect is opposite. Generally, we can take $c = 55 \sim 60$ for most of the cases.

We now discuss how the JPEG quality factor Q_{jpeg} is related to the performance of the watermarking algorithm. This factor is the minimum JPEG quality setting up to which the watermark is resistant against re-encoding. We select $\tau = 0.2, n = 16$ and $c = 55$, and run the program with various values of Q_{jpeg} for the image "palace.pnm" shown in Figure 3.7. The results are shown in Table 3.1 and Figure 3.8. In the table, given Q_{jpeg}, we encode the watermark, then re-compress the image using JPEG with different values of JPEG quality factor and record the minimum of them for which the watermark

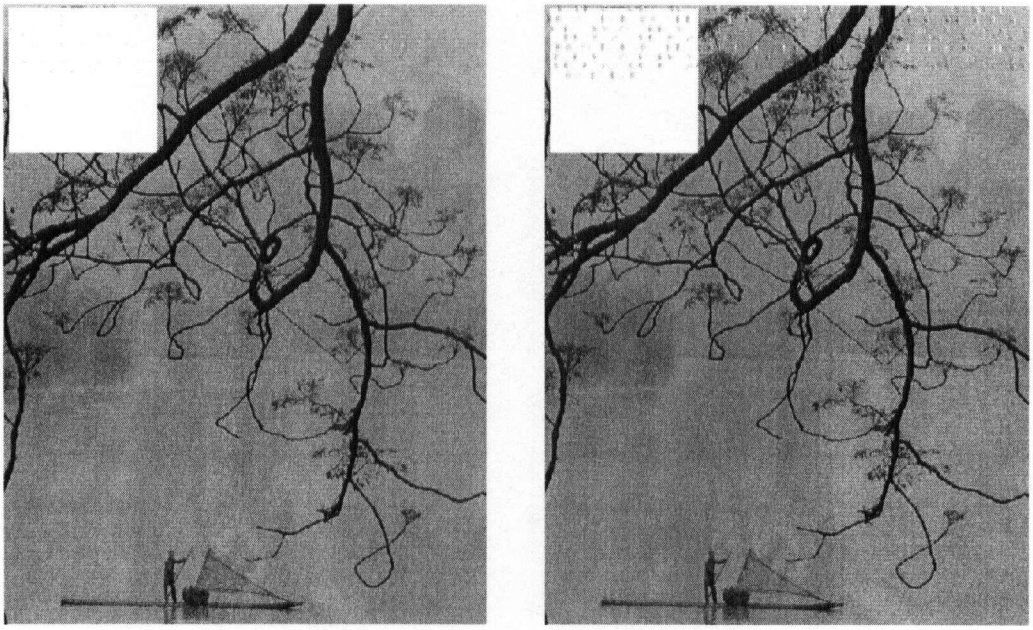


Figure 3.5: Plot of the watermarked images with $c = 32$ (left) and $c = 60$(right).Here $n = 2$, $Q_{jpeg} = 75$ and $\tau = 1$. Noises are more noticeable for $c = 60$ than for $c = 32$.

can still be decoded correctly. We can notice that a small value of Q_{jpeg} can provide more robustness of the watermarking scheme, but will degrade the image quality.

Q_{jpeg}	90	80	75	70	65	60	50	40	30	20
mini. factor	86	74	74	65	65	56	38	33	27	0

Table 3.1: The minimal re-encode quality factor with which the watermark based on DCT coefficient removal can still be decoded correctly, given a JPEG compression quality factor Q_{jpeg}. The watermark can survive a wider range of JPEG quality factors for a smaller Q_{jpeg}.

The relationship between the threshold τ and the performance of a watermarking algorithm is obvious: larger the value of τ is, more robust but more visible the embedded watermark is. This is illustrated in Figure 3.9 and

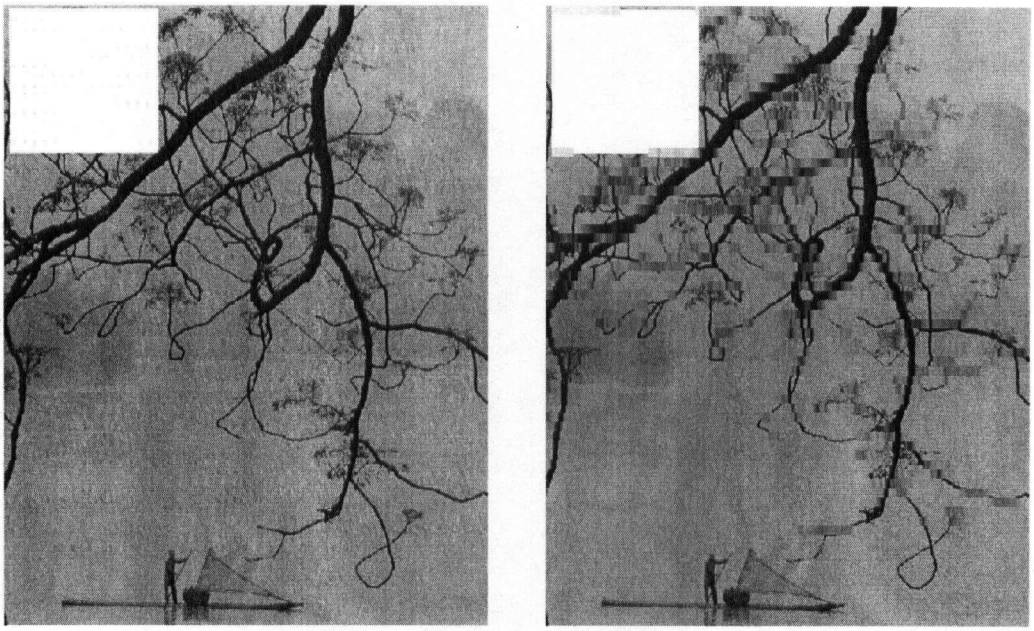


Figure 3.6: Plot of the watermarked images with $c = 55$ (left) and $c = 5$(right).Here $n = 16$, $Q_{jpeg} = 75$ and $\tau = 1$. Block-effect is more noticeable for $c = 5$ than for $c = 55$.

Table 3.2, where we watermark the image "palace.pnm" using $n = 16, c = 55, Q_{jpeg} = 75$ with various values of τ. In the table, we first encode the watermark in the image, then compress this file with JPEG encoding using different values of quality factors and record the minimum of them for which the watermark can still be decoded correctly. From this table, we can see that with a larger value of τ, the watermark can survive JPEG compression using a higher compression rate.

We have also tested if our watermarking algorithm in § 3.1.1 can survive other image processing attacks. Our experiments demonstrated that the algorithm may be robust to some image processing attacks such as blur, motion blur, supernova, sparkle, sharpening, NL filter, destripe, applycanvas, noisify,

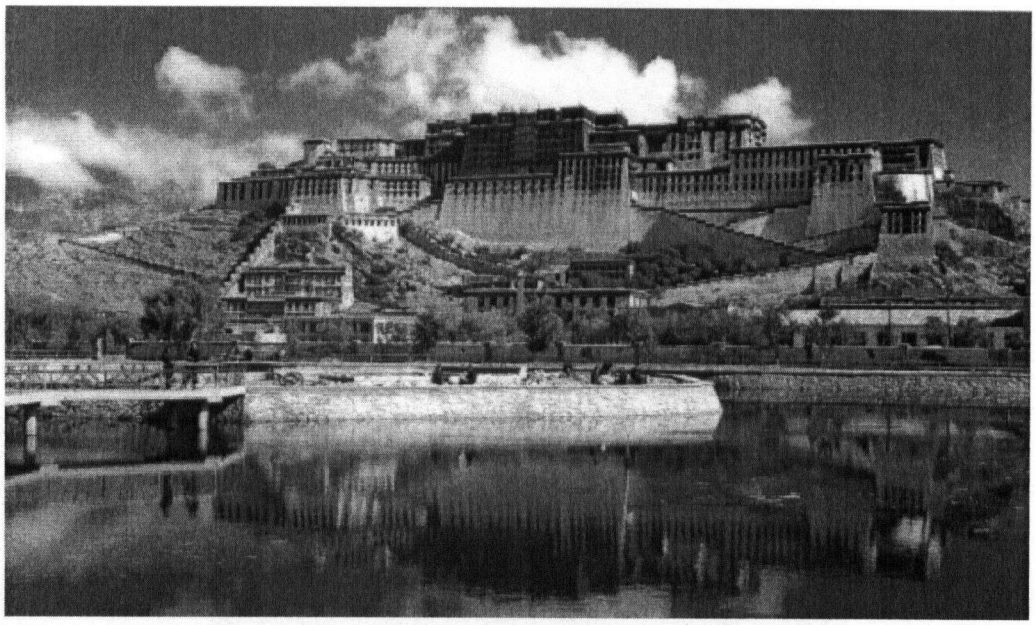


Figure 3.7: Plot of the original image "palace".

τ	2	1	0.5	0.1	0.05	0.01
mini. factor	28	35	60	77	85	94

Table 3.2: The minimal re-encode quality factor with which the watermark based on DCT coefficient removal can still be decoded correctly, given a threshold τ. The watermark can survive a wider range of JPEG quality factors for a larger τ.

etc. (These filters are given in Gimp [56]).

3.3.2 Other schemes

In this sub-section, we implement the watermarking algorithms introduced in § 3.2 to study how these schemes survive JPEG compression attack and if the self-reference pattern can improve the performance of the watermark extraction.

For the algorithm based on quantization rule, we first analyze how the JPEG quality factor Q_{jpeg} is related to the performance of the watermark again JPEG compression attack. We encode the image in Figure 3.7 with different values of Q_{jpeg}. We choose $c_m = 21$, $c_M = 35$, $\alpha = 1$ and the size of lc-region is four. The encoded watermark message is "UBC". The result is given in Table 3.3, in which given the value of Q_{jpeg}, we encode the watermark in the image, then compress the watermarked image using JPEG with different values of JPEG quality factor and record the minimum of them for which the watermark can still be recovered correctly. We can notice that the small value of Q_{jpeg} can provide more robustness of the watermarking scheme, but at the expense of image quality. Also we can notice that the algorithm with self-reference pattern gives better performance. We also test the relationship between the parameters α and the robustness. The results are shown in Table 3.4, where $c_m = 21$, $c_M = 35$, $Q_{jpeg} = 75$ and the size of lc-region is four. We again notice that self-reference gives better robustness again JPEG compression. And the value of α can increase the performance of the watermark but it will degrade the image quality as it can be expected.

Q_{jpeg}	90	80	75	70	65	60	50	40	30	20
mini. factor: SF	81	60	51	43	36	32	26	20	16	0
mini. factor: NSF	86	72	64	57	49	44	35	28	21	16

Table 3.3: The minimal re-encode quality factor with which the watermark based on quantization can still be decoded correctly, given a JPEG compression quality factor Q_{jpeg}. Here SF means using self-reference and NSF means no self-reference. The watermark can survive a wider range of JPEG quality factors when using self-reference.

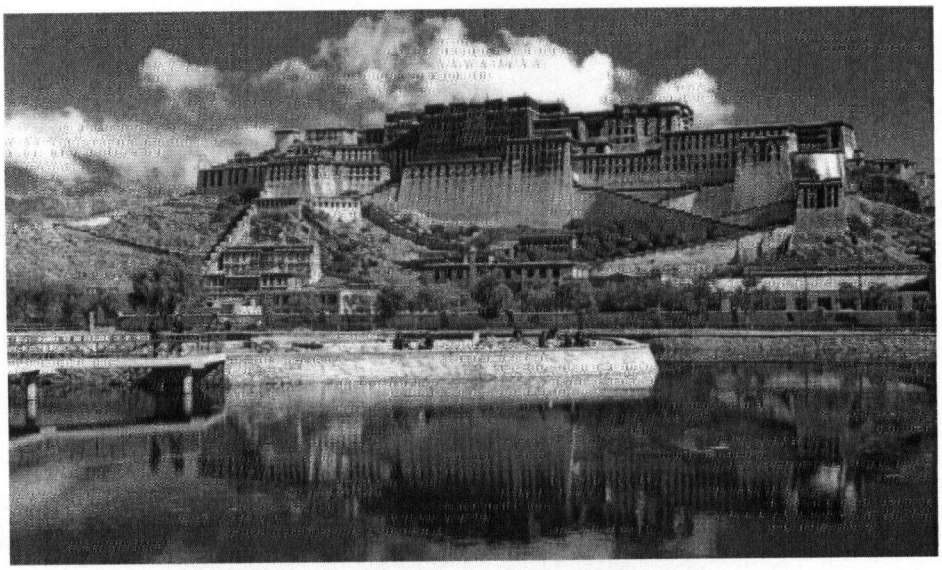
$1/\alpha$	1	1.5	2	4
mini. factor: SF	51	68	76	88
mini. factor: NSF	64	76	83	91

Table 3.4: The minimal re-encode quality factor with which the watermark based on quantization can still be decoded correctly, given a watermark strength scale parameter α. Here SF means using self-reference and NSF means no self-reference. The watermark can survive a wider range and JPEG quality factors for a larger α.

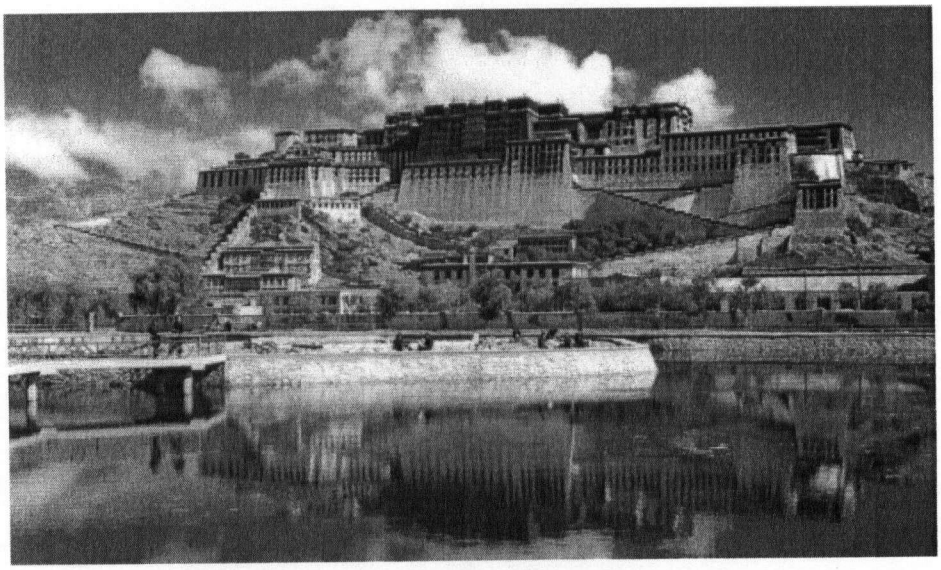
For the watermarking algorithm based on coefficient comparison in § 3.2, we first study how robust this algorithm is against the JPEG compression attack. We choose $c_m = 21$, $c_M = 35$, $Q_{jpeg} = 75$ and the pre-defined threshold $\tau = QTable[c'_m] + QTable[c'_M]$, where c'_m and c'_M are the zig-zag positions corresponding c_m and c_M. The encoded message is "computer". We list the results in Table 3.5. Comparing this table with Tables 3.1 and 3.3, we notice that this scheme yields better performance than the watermarking algorithms based on DCT coefficient removal and quantization. In addition, the distortion in the watermarked images is usually less noticeable since most of the *lc*-regions are untouched. A disadvantage of this scheme is that a secret file which keeps the position and pairing information is needed, since this is not convenient for some applications such as searching images with a specific watermark on Internet.

Q_{jpeg}	90	80	75	70	65	60	50	40	30	20
mini. factor	78	54	41	38	34	29	25	21	17	16

Table 3.5: The minimal re-encode quality factor with which the watermark based on DCT coefficient comparison can still be decoded correctly, given a JPEG compression quality factor Q_{jpeg}.

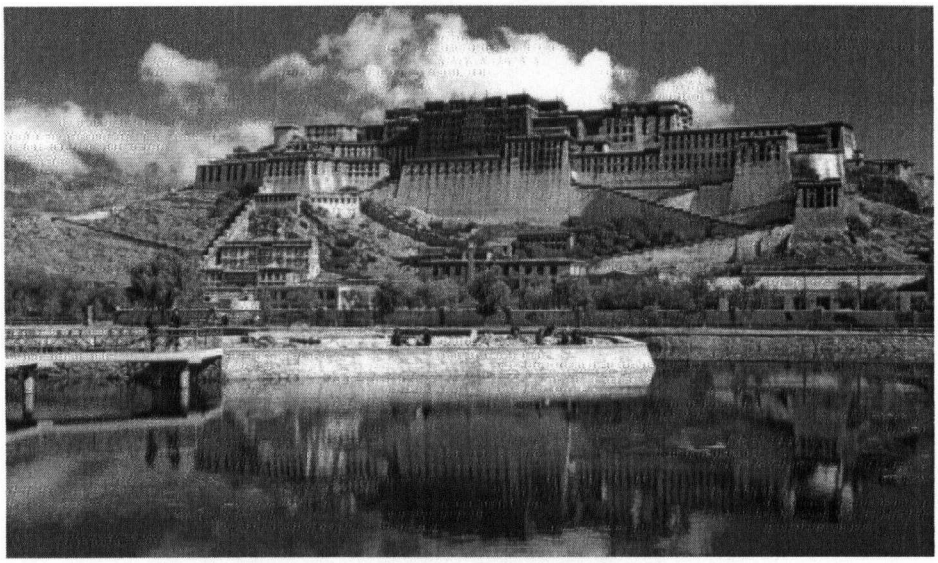


$Q_{jpeg} = 20$

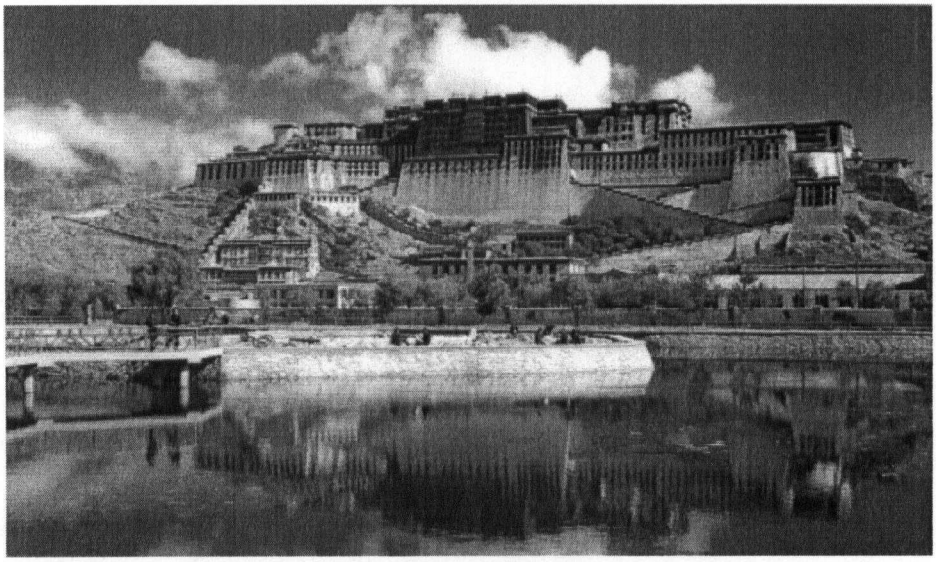


$Q_{jpeg} = 75$

Figure 3.8: Plots of the watermarked image. Here $n = 16$, $c = 55$ and $\tau = 0.2$. Notice that larger values of Q_{jpeg} can give better image quality, but less robustness to JPEG compression.



$\tau = 2$



$\tau = 0.1$

Figure 3.9: Plots of the watermarked image. Here $n = 16$, $c = 55$ and $Q_{jpeg} = 75$. Notice that smaller values of τ can yield better image quality, but less robustness to JPEG compression.

Chapter 4

Watermarking algorithms using spread spectrum technique

The spread spectrum techniques used in RF communications (cf. [10, 43]) are frequently applied in digital watermarking development (cf. [44, 55, 32, 45, 53, 8]). Through the spread spectrum techniques, signal-to-noise ratio (SNR) is traded for bandwidth: the signal energy is spread over a wider frequency band at low SNR such that it is hard to detect, intercept, or jam. In the context of information hiding or watermarking, the goal is to send a message, the watermark, over a very noisy channel, the image. The explosive interest in developing the spread spectrum based watermarking schemes is due to the fact that spread spectrum signals, which are distributed over a wide range of frequencies and then collected onto their original frequencies at the receiver, are so inconspicuous as to be "transparent". Just as they are unlikely to be intercept, detect or jam in RF communication by a military opponent, so are

the watermarks unlikely to be detected or destroyed.

One big difference is that in watermarking communication, the channel is very noisy (unless the image is uniform) and largely non-Gaussian. Another difference is that signal transmission in watermarking process doesn't have any connection to the physical world because modulation, transmission and demodulation are usually performed in a purely digital environment.

The general model for a watermarking system can be depicted as in Figure 4.1. The input is an N-bit binary information S. The information is modulated and added to the image in some modulation space. The modulation of the input is based on certain spread spectrum techniques such as direct sequence, frequency hopping, chirp, etc. But we will only employ a direct sequence spread spectrum approach in the thesis. After the watermark embedding, the image is released and thus subject to various attacks and alternations. The watermarked or probably distorted image is then the input to the demodulator which performs the following two tasks: detect if the image under investigation is watermarked; if so, demodulate the embedded information.

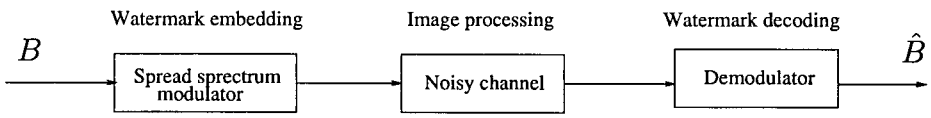


Figure 4.1: A generic watermarking process.

The organization of this chapter is as follows. In § 4.1, we introduce the basic idea of direct sequence spread spectrum. Then the watermarking algorithm based on spread spectrum is proposed in § 4.2, and the experiment results are presented in § 4.3.

4.1 Introduction to direct sequence spread spectrum

In this section, we give a introduction the direct sequence spread spectrum. Let's consider a binary discrete time communication system with the received signal as a time sequence $\{s_i\}_{i=0}^{\infty}$ defined by

$$s_i = \Upsilon b_i + n_i \,,$$

where Υ is the unit chip energy, b_i is the sequence of two input symbols which are antipodal binary, i.e., $b_i \in \{-1, 1\}$, and n_i is additive white Gaussian noise with zero mean, i.e., the expectation values satisfy

$$E(n_i) = 0 \,, \quad E(n_i n_{i+l}) = \sigma^2 \delta(l) \,.$$

Here σ is the standard variance and $\delta(l)$ is the dialet function. To determine whether a positive one or a negative one was transmitted, we assume that the transmitter is connected to an information source which yields +1's and -1's with equal probability. In this circumstance, such a receiver is a simple level detector using the theory of Hypothesis Testing [22, 26]:

$\mathbf{H_0}$: +1 was sent if $s_i > 0$;

$\mathbf{H_1}$: -1 was sent if $s_i < 0$.

Here $y_i = s_i$ is called the decision variable. Its statistic determine the performance of the receiver. It is not hard to show that y_i is a normal (Gaussian) random variable with mean Υb_i and variance σ^2.

Now we modulate each input symbol b_i with another ± 1-valued sequence called a spreading sequence $\{c_j\}_{j=0}^{N-1}$. Depending on the value of b_i, each symbol b_i results in the transmission of either

$$c_0, c_1, \ldots, c_{N-1},$$

or

$$-c_0, -c_1, \ldots, -c_{N-1}\,.$$

Thus each bit of duration T is encoded into a sequence of N chips of duration $T_c = T/N$. And the received sequence can be written as

$$s_j = \Upsilon_c b c_j + n_j\,, \quad j = 0, 1, \ldots, N-1\,,$$

where $\Upsilon_c = \frac{\Upsilon}{N}$, $E(n_j^2) = \frac{\sigma^2}{N}$ and we omit the subscript i for simplicity. Now we make two assumptions about the spreading sequence $\{c_j\}$:

- its mean is approximately zero, i.e., $\sum_{j=0}^{N-1} c_j \simeq 0$;
- its autocorrelation is given by

$$\sum_{j=0}^{N-1} c_j c_{j+k} \simeq \begin{cases} N\,, & k = 0\,, \\ 0\,, & k \neq 0\,. \end{cases}$$

These two conditions are ideal, but can be closely approached in practice when N is large. This type of sequences are noise-like, thus called pseudo-noise (PN)

sequences. To retrieve the transmitted signal, we use a correlation receiver to determine whether a $+1$ or -1 was transmitted. The correlation receiver performs the following operation to obtain the decision variable y:

$$y = \sum_{j=0}^{N-1} s_j c_j \,,$$

i.e.,

$$y = \sum_{j=0}^{N-1} (\Upsilon_c b c_j + n_j) c_j \,.$$

Using the properties of the spreading sequence, it is easy to show that

$$y = N\Upsilon_c b + \sum_{j=0}^{N-1} n_j c_j \,,$$

which is normal with mean $N\Upsilon_c b = \Upsilon b$ and variance σ^2. Compared with the non-spreading system above, this result shows that the spreading yields no improvement in the ideal white Gaussian noisy channel. This can be intuitively explained by the fact that the signal bandwidth is increased by a factor of N, although the unit chip energy is decreased by a factor of N.

As we will show, however, the power of spectrum spreading is its effect on narrow band or correlated signals. These includes interference, multi-path or signals from other transmitters. Now we assume that there is an interferer in the communication channel, i.e., an unknown constant is added to the transmitted signal. Then we have

$$s_j = \Upsilon_c b c_j + I_j + n_j \,, \quad j = 0, 1, \ldots, N-1 \,,$$

where $I_j = I$ is a real unknown constant. Then we calculate the decision variable for our correlation receiver

$$y = \sum_{j=0}^{N-1} (\Upsilon_c b c_j + I_j + n_j) c_j \,.$$

It is easy to show that

$$\begin{aligned} y &= N\Upsilon_c b + I\sum_{j=0}^{N-1} c_j + \sum_{j=0}^{N-1} n_j c_j \\ &\simeq N\Upsilon_c b + 0 + \sum_{j=0}^{N-1} n_j c_j \,. \end{aligned}$$

Again the decision variable is normal with mean $N\Upsilon_c b = \Upsilon b$ and variance σ^2, so the interference is suppressed by the despreading/correlation operation. On the other hand, the decision variable in a non-spread system is normal with a mean of $\Upsilon b + I$, which will render the system useless for $|I|$ sufficiently large. Similar results can be obtained for a multi-path channel with a direct path and a specular (reflected) path which causes another copy of the signal to arrive at a delay of l with unknown attenuation β:

$$s_j = \begin{cases} b_i c_i + \beta b_{i-1} c_{N-l+j}\,, & j = 0, 1, \ldots, l-1, \\ b_j c_j + \beta b_j c_{n-l}\,, & j = l, \ldots, N-1 \end{cases}$$

where we assume that $l < N$, i.e., the delay is less than one symbol duration. Specifically, we can calculate the decision variable

$$\begin{aligned} y_i &= \sum_{j=0}^{N-1} s_j c_j \\ &= N b_i + \beta b_{i-1} \sum_{j=0}^{l-1} c_{N-l+j} c_j + \beta b_j \sum_{j=l}^{N-1} c_{j-l} c_j + \sum_{j=0}^{N-1} n_j c_j \,. \end{aligned}$$

This yields, using the properties of c_j,

$$y_i \simeq N b_j + 0 + 0 + \sum_{j=0}^{N-1} c_j b_j \,.$$

Again the multi-path signal is suppressed by the despreading/correlation, but for the unspread system, this system has severe inter symbol interference (ISI) and will result in a performance loss.

Based on the above observation, we can anticipate that spread spectrum techniques can yield a more robust and more imperceptible digital watermark at the expense of less watermark capacity (the number of bits that may be hidden and then recovered).

4.2 Watermark embedding and decoding

For our image watermarking, we employ the above direct sequence spread spectrum modulation technique. The basic idea is to spread the signal over all or part of frequencies of an image to increase robustness and resilience to noise. In direct sequence spread spectrum modulation, a binary information is modulated with a binary pseudo-noise sequence (also called chirp sequence). This leads to a spreading of the frequency spectrum of the input signal.

Our generic watermark encoding system is described in Figure 4.2. The goal is to embed an N bit long watermark $B = \{b_0, b_1, \ldots, b_{N-1}\}$ into an image I. Assume $I = \{I_{mn}\}$ is an gray image, where $(m, n) \in Z^2$ is the spatial location in the Cartesian coordinate system. In general, the pixel values I_{mn} are continuous, however in digital imaging, they are usually coded with 8 bits, which means that they are any integer values between 0 and 255.

The watermark embedding process takes place in the watermarking space Ω. To project an image into the watermarking space, a transformation χ is applied to the image, i.e., $\chi : I(k, l) \rightarrow C(m, n)$. After the watermark is embedded in the watermarking space, an inverse transformation is performed to obtain the watermarked image, i.e., $\chi^{-1} : \hat{C}(m, n) \rightarrow \hat{I}(k, l)$. There is no

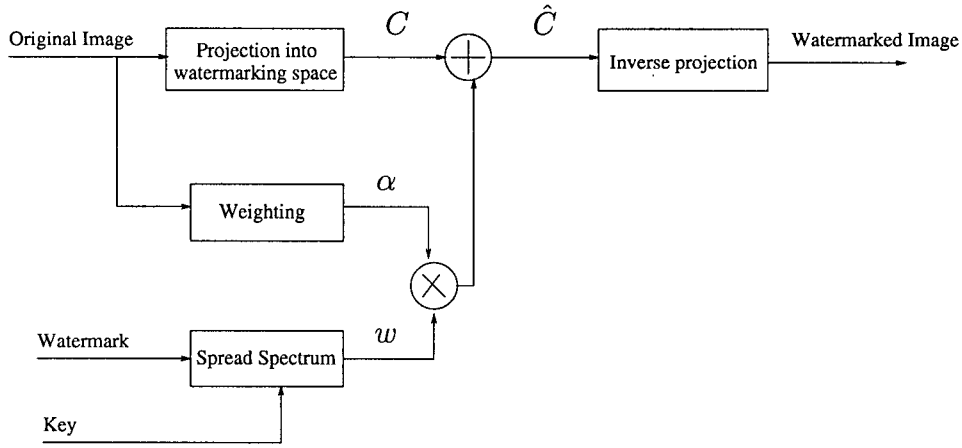


Figure 4.2: A generic watermark embedding system.

special constraints on the projection except that the image representation in the watermarking space has to be two dimensional.

To embed an N bit binary watermark B, we employ a set Φ^k of N two dimensional orthogonal functions ϕ_i, $i \in \{1, \ldots, N\}$, where k defines the secret key used as initializing seed to generate the set. Each function ϕ_i in the set is used to represent one bit value of the watermark. These functions can be defined as the pseudo-noise sequences depending on different keys. To not introduce the inter symbol interference (ISI), i.e., to make the functions orthogonal, it is convenient to design them to be not overlapping. That is, if $\Phi_i = \{(m,n), \forall \phi_i(m,n) \neq 0\}$ is the set of all locations for which the function ϕ_i is not zero, then the intersection of all these set Φ_i is an empty set, $\Phi_i \bigcap \Phi_j =$

$0, \forall i = j$.

Now the watermark can be defined as

$$w(m,n) = \sum_{i=0}^{N} b_i' \alpha(m,n) \phi_i(m,n)\,, \tag{4.1}$$

where $\alpha(m,n)$ is a local scaling factor which adapts the watermark as robust and imperceptible as possible, and b_i' is defined by

$$b_i' = \begin{cases} -1\,, & \text{if } b_i = 0\,, \\ 1\,, & \text{if } b_i = 1\,. \end{cases} \tag{4.2}$$

By adding the watermark to the image representation in the watermarking space and applying the inverse projection, we obtain the watermarked image

$$\hat{I} = \chi^{-1}(C + w)\,, \tag{4.3}$$

The best way to determine an optimal scaling factor $\alpha(m,n)$ such that the image distortion is minimized whiling maintaining strongest robustness is to employ the human visual system. We, however, use a simple interpolation for $\alpha(m,n)$ in next section.

To extract the embedded watermark, the correlation is calculated between the data under investigation $\hat{C}$ and the modulation function ϕ_i, as shown in Figure 4.3. The sign of the correlation is used to determine the embedded watermark bit. A positive correlation indicates an embedded bit value of 1, while a negative correlation indicates an embedded bit value of 0.

We now analyze the correlator statistics by investigating a watermarked and undistorted image $\hat{C} = C + w$ projected into the watermarking space. For

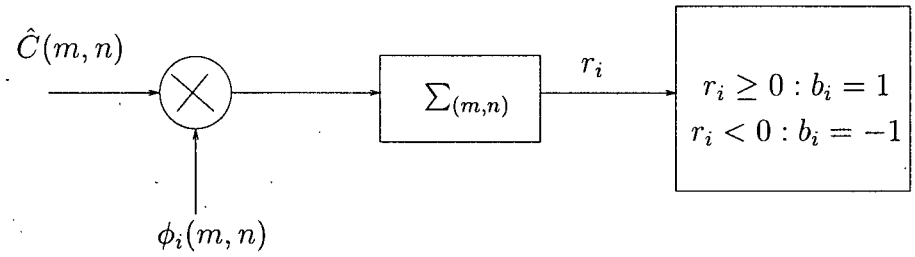


Figure 4.3: A generic detector based on correlation with the spread spectrum sequence.

a bit b_i, the detector statistic r_i conditioned on a fixed set Φ^k is given by

$$\begin{aligned} r_i &= <\hat{C}, \phi_i> = <C, \phi_i> + <w, \phi_i> \\ &= \sum_{(m,n)} C(m,n)\phi_i(m,n) + \sum_{(m,n)} w(m,n)\phi_i(m,n)\,, \end{aligned} \tag{4.4}$$

where $<\cdot,\cdot>$ is the inner product operator. To study the performance of the watermark detector, we calculate the mean and the variance of the random variable at the output of the detector. It is easy to show that

$$E[r_i] = E[\sum_{(m,n)} w(m,n)\phi_i(m,n)] + E[\sum_{(m,n)} C(m,n)\phi_i(m,n)]\,. \tag{4.5}$$

Substituting (4.1) into (4.5) and using the fact that s_i and $s_j (i \neq j)$ are not overlapping give

$$E[r_i] = E[\sum_{(m,n)} b_i' \alpha(m,n)\phi_i^2(m,n)] + E[\sum_{(m,n)} C(m,n)\phi_i(m,n)]\,. \tag{4.6}$$

Assuming that the distribution of s_i is symmetric and has zero mean, the second item in (4.6) becomes zero. If we further assume the variance $\sigma_{s_i} = Var[\phi_i]$ to be 1, then the expectation value becomes

$$E[r_i] = b_i' \sum_{(m,n)} \alpha(m,n)\,. \tag{4.7}$$

To calculate the variance σ_{s_i}, we compute the second moment of the detector statistic and subtract the square of its expectation (4.7). The second moment is defined as follows

$$\begin{aligned} E[r_i^2] &= E\left[\left\{\sum_{(m,n)\in\Phi_i} w(m,n)\phi_i(m,n) + \sum_{(m,n)\in\Phi} C(m,n)\phi_i(m,n)\right\}^2\right] \\ &= E\left[\left\{\sum_{(m,n)\in\Phi_i} b_i'\alpha(m,n)\phi_i^2(m,n)\right\}^2 + \left\{\sum_{(m,n)\in\Phi_i} C(m,n)\phi_i(m,n)\right\}^2\right. \\ &\quad \left. +2\sum_{(m,n)\in\Phi_i} b_i'\alpha(m,n)\phi_i^2(m,n)\cdot\sum_{(m,n)\in\Phi_i} C(m,n)\phi_i(m,n)\right]. \end{aligned} \tag{4.8}$$

It is not hard to derive that if we we assume $E[\phi_i^2] = 1$, then

$$E\left[\left\{\sum_{(m,n)\in\Phi_i} C(m,n)\phi_i(m,n)\right\}^2\right] = \sum_{(m,n)\in\Phi_i} C(m,n)^2, \tag{4.9}$$

$$\begin{aligned} &E\left[2\sum_{(m,n)\in\Phi_i} b_i'\alpha(m,n)\phi_i^2(m,n)\sum_{(m,n)\in\Phi_i} C(m,n)\phi_i(m,n)\right] \\ &\quad = 2b_i'\sum_{(m,n)\in\Phi_i}\alpha(m,n)C(m,n)E[\phi_i^3(m,n)] \end{aligned} \tag{4.10}$$

and

$$E\left[\left\{\sum_{(m,n)\in\Phi_i} b_i'\alpha(m,n)\phi_i^2(m,n)\right\}^2\right] = \sum_{(m,n)\in\Phi_i}\alpha^2(m,n)E\left[s_i^4\right]. \tag{4.11}$$

Combining (4.8) to (4.11), we obtain

$$\begin{aligned} E\left[r_i^2\right] &= \sum_{(m,n)\in\Phi_i} C^2(m,n) + \sum_{(m,n)\in\Phi_i}\alpha^2(m,n)E\left[s_i^4\right] \\ &\quad +2b_i\sum_{(m,n)\in\Phi}\alpha(m,n)C(m,n)E\left[s_i^3(m,n)\right], \end{aligned} \tag{4.12}$$

i.e., the variance of the detector is

$$\begin{aligned} Var\left[r_i\right] &= E\left[r_i^2\right] - E\left[r_i\right]^2 \\ &= \sum_{(m,n)\in\Phi_i} C^2(m,n) + \sum_{(m,n)\in\Phi_i}\alpha^2(m,n)\left(E\left[s_i^4\right] - 1\right), \end{aligned} \tag{4.13}$$

where we assume again that the distribution of s_i is symmetric and has zero mean (i.e., $E[s_i^3] = 0$).

In order to increase the performance of the watermarking scheme, we should make the expectation of the correlation as large as possible while keep its variance as small as possible. By inspecting the expectation (4.7) and the variance (4.13), we can make the following observations

- The expected value depends exclusively on the watermark embedding strength α, which suggests that finding optimal large scaling values is important under the restriction of invisibility of the watermark.

- The four-th moment of any probability density function (pdf) is larger than or equal to 1, i.e., $E[s_i^4] \geq 1$ if $E[s_i^2] = 1$ ([11]). The lower bound is achieved only when the random variables are bilevel. In this case, the last term of (4.13) vanishes.

- The first term of (4.13) depends exclusively on the image in watermarking space or the second order moment. So one way to decrease the variance (4.13) is to subtract out the mean of the image, i.e., $C - E[C] \rightarrow \hat{C}$. In this circumstance, the second order moment is equivalent to the variance since $E[x^2] = Var[x] + E^2[x]$ and here $E[x] \doteq 0$. This modification doesn't change the expected value of r_i, but may substantially increase the detector performance. We will examine this behavior in next section.

The previous discussion is based on a generic watermarking space Ω. There can be difference choices of this space. In [1] and [44], the watermark-

ing space is just correspondent to the spatial domain, i.e., a spatial domain technique. In [32], the DCT domain of the whole image is used as Ω, which corresponds to a spectral domain. Specifically, they first compute the DCT coefficients of an $N \times N$ image the re-order them into a zig-zag scan. Let a pseudo-random sequence (size of M) represent a watermark bit. After skipping the first L lowest DCT coefficients, they embed the watermark into the next M DCT coefficients as follows (see Chapter 2 for details).

In our implementation, we also choose the DCT domain as the watermarking space Ω. However, we take a different approach as that in [32]. Specifically, the image is first divided into square blocks of size 8×8 for which the DCT is computed as in the JPEG compression scheme. The DCT coefficients are re-ordered into a zig-zag scan and we denote the total number of the blocks by K. In order to obtain a trade-off between perceptual invisibility and robustness to image processing techniques, we also skip the first L DCT coefficients and only embed our binary watermark bits into the next M DCT coefficients in each DCT block, i.e., $T = \{t_{L+1}, \ldots, t_{L+M}\}$, where $1 \leq L$, $L + M \leq 64$ and $N \leq M$. Here we restrict the number N of the the watermark bits is less than M, i.e., N can't be greater than 64. Now given a key generator KEY which generates an integer between $L + 1$ and $L + M$ without duplicate for each integer $i : 0 \leq i \leq N - 1$, we embed our watermark $B = \{b_0, b_1, \ldots, b_{N-1}\}$ into the DCT coefficient as follows, for each block $K(k = 0, 1, \ldots, K - 1)$,

$$t_{i'}(k) = t_{i'}(k) + \beta(i)|t_i(k)|b_i' s(k),\ i = 0, 1, \ldots, N - 1\,, \qquad (4.14)$$

where b'_i is defined in (4.2), $i' = KEY(i)$ is determined by the key generator and i, $t_{i'}(k)$ is the i'-th scanned DCT coefficient in k-th DCT block, $\beta(i)$ is the scaling parameter, and $(s(0), s(1), \ldots, s(K-1))$ is a pseudo-random sequence which can be normal, uniform, bilevel, etc. More precisely, we embed bit b_i into the i'-th DCT coefficients of all the DCT blocks, i.e., we spread a bit all over the domain, and the pseudo-random sequence $s(k), k = 0, 1, \ldots, K-1$ can be obtained by a pseudo-random generator with certain seed. If we write (4.14) in the form of (4.1), then we have $\alpha(m,n) = \beta(i')|t_{i'}(k)|$ and $\phi_i(m,n) = s(k)$. Here $\phi_i(m,n)$ is identical for each (m,n), but you can make ϕ_i depend on the coefficient location which is unnecessary. To retrieval the embedded watermark, we calculate the correlation r_i (4.4), which is

$$r_i = \sum_{k=0}^{K-1} t_{i'}(k)s(k)\,. \tag{4.15}$$

Then a positive correlation indicates a bit 1 and a negative correlation indicates a bit 0.

Compared with Piva's DCT-based watermarking scheme [32], our spread spectrum based watermarking algorithm has several advantages. First, since our transformation is performed on 8×8 DCT blocks, it is faster and more computational economic when the size of the image is large. This is true even when the fast Fourier transformation is employed. Secondly, our algorithm can embed multiple watermark bits into an image while the scheme in [32] is only limited to one single bit. More importantly, since we employ the spread spectrum technique and the bit is spread over the whole image, the embedded watermark using our scheme may easily easily be recovered if the image has

been cropped or translated as well as JPEG compressed.

4.3 Experiment results

To illustrate the effectiveness of the proposed watermarking system, we will evaluate its robustness against attacks such as JPEG compression, cropping and multiple watermarking. We will also study the performance of the watermarking system with respect to the choices of the scale parameter β, basis functions ϕ_i, etc.

One major difference of this spread spectrum watermarking algorithm with those in previous chapter is that it is not only survives JPEG compression, but also cropping attack. In addition, we can embed multiple watermarks in an image without interference. In our first experiment, we insert a secret message, an integer "1987", into the image (Figure 3.3) using the spread spectrum algorithm. In this algorithm, we set $L = 30$, $L + M = 41$ and the scale parameter β is a linear interpolation of 0.4 at t_{L+1} and 2.4 at t_{L+M}. This choice of β is obviously not an optimized one, but is better than any constant due to the fact that for higher frequencies more energy can be put in without causing visual distortions. The pseudo-random sequence $(s(0), s(1), \ldots, s(N-1))$ is generated according to the normal distribution $\mathbf{N}(0,1)$. For this configuration, we notice that the watermarked image which is shown in Figure 4.4 is indistinguishable from the original image. The watermarked image can resist the JPEG compression attack with compression factor Q_{jpeg} greater than 15, which is a much better result than in Chapter 4 (See Table 3.1). In addition,

this algorithm is also robust to cropping, even when the most of the watermarked image is cropped. In Figure 4.5 we can find that the re-JPEGed image with $Q_{jpeg} = 16$ is very distorted and the cropped image is totally unusable, but we can still recover the embedded message without any bit errors.

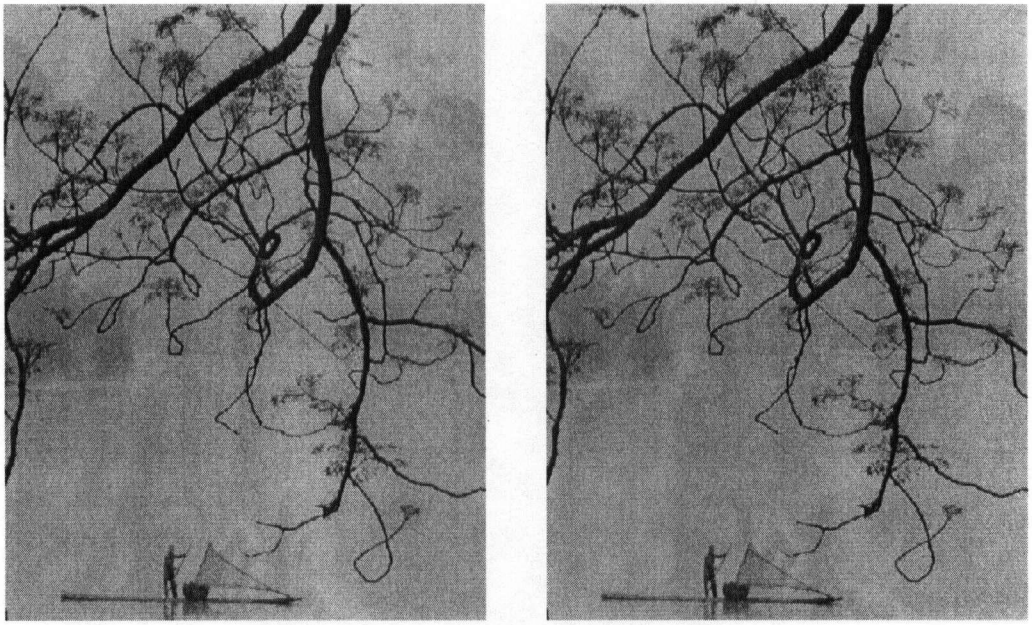


Figure 4.4: Plot of the original image (left) and the watermarked image with spread spectrum algorithm (4.14) (right). No visual distortion is observed.

Another advantage of this spread spectrum scheme is that it can embed several watermarks into an image and can retrieve them without interference. To illustrate this, the original image is first signed with a watermark bit corresponding to a seed that equals to 300 and the watermarked image is then signed again with another watermark bit corresponding to a seed that equals to 600. The final image is shown in Figure 4.6. To detect the watermarks, we calculate the correlation (4.15) with various watermarks. The responses of the watermark detector to the marks is shown in Figure 4.7. From this figure,

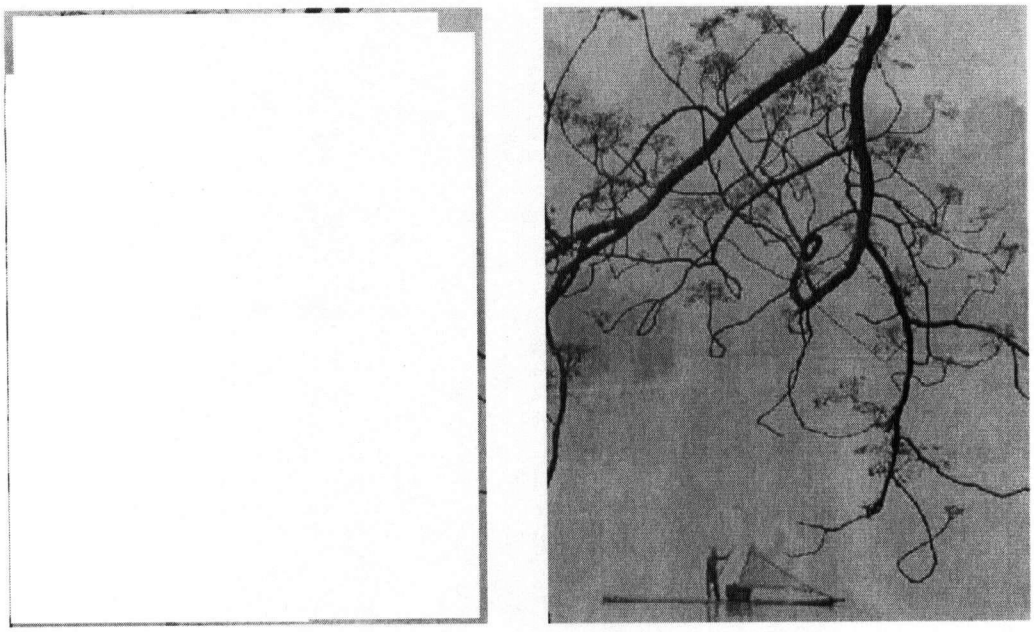


Figure 4.5: Plot of the cropped image (left) and the JPEG compressed watermarked image with compression rate 16 (right). The embedded message can still be recovered in these severely distorted images.

we find that the responses to the correct watermarks are much larger than the responses to the others, suggesting that the possibility of achieving very low false positive and false negative rates.

The more energy we put for a watermark, the more robust the watermark will be. So if we choose the scaling parameter β larger, the watermark will be retrieved with a larger probability of correct recovery. This is partially illustrated by Table 4.1, where β is the scaling parameter, *mean* is the mean value of the detector responses to the various watermarks except the correct one, *std* is the corresponding standard deviation, *peak* is the response of the detector to the correct watermark and *ratio* is the ratio between the values of *peak* and *std*. We can notice that the ratio of the responses of the watermark

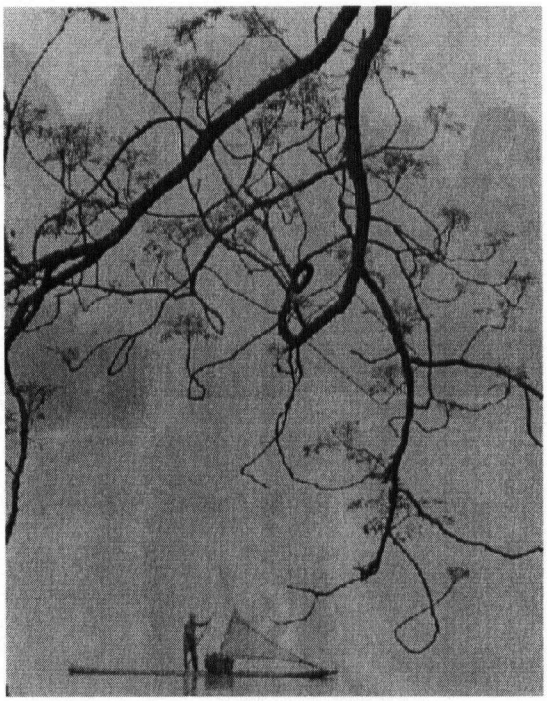


Figure 4.6: Plot of the twicely watermarked image. One watermark corresponds to seed 300, and another one corresponds to seed 600.

detector to the correct watermark and others becomes larger when the value of β increases.

β	0.25	0.5	1	1.5	2	3	4
mean	-36.42	-34.97	-42.28	-43.91	-52.18	-51.72	-56.67
std	605.9	657.2	839.7	1057.1	1318.3	1835.3	2369.4
peak	4343.9	11107.1	25014.4	38736.9	52390.6	79388.4	106267.1
ratio	7.18	16.9	29.8	36.6	39.7	43.3	44.9

Table 4.1: The responses of the watermark detector to the correct watermark and others becomes larger when the value of β increases.

Another performance related factor is the expectation value $E(C)$ of the image in watermarking space Ω. We mentioned that one way to decrease the variance of the correlation r_i is to subtract it out the image. However, we

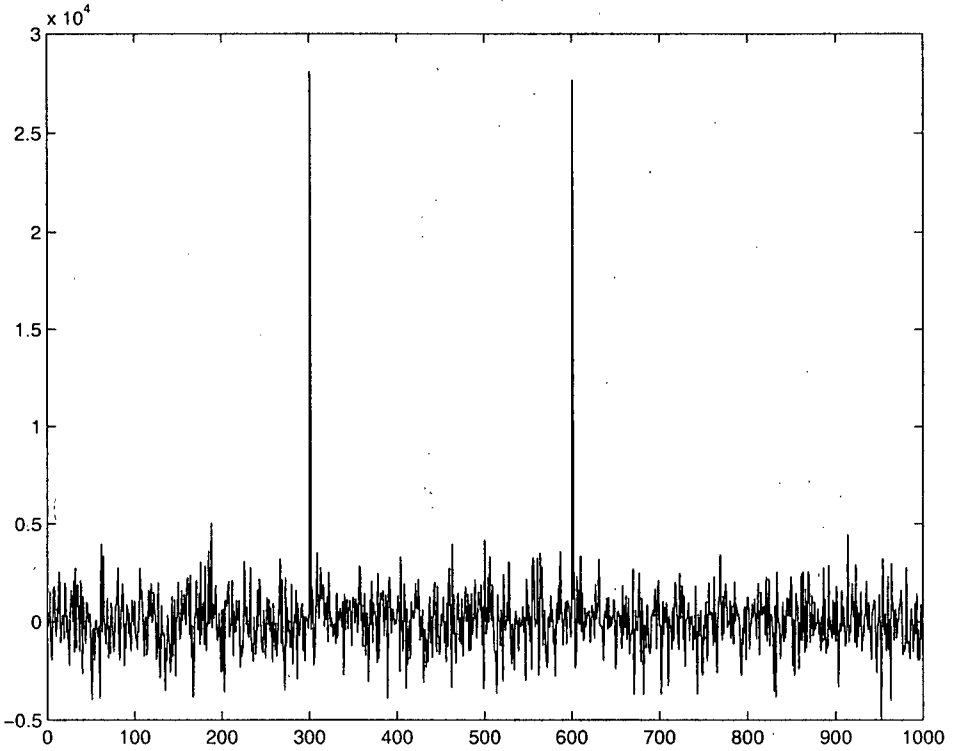


Figure 4.7: Plot of the watermark detector responses to different watermarks. Peaks occur only when the watermarks correspond to seed 300 and 600.

didn't see any big difference if we do so in the experiments. In fact, it is easy to observe that the expectation value $E(C)$ in the DCT domain is zero. This can be illustrated by Table 4.2, where we calculate the average value of the related DCT coefficients of the image in Figure 4.4 for different values of β.

β	0.25	0.5	1	1.5	2	3	4
$\tilde{E}(C)$	0.07998	0.09186	0.08912	0.12591	0.10950	0.12798	0.16099

Table 4.2: The average value $\tilde{E}(C)$of the DCT coefficients for different values of β.

From (4.13), we may expect that bilevel random variable will probably

give the best performance of a watermarking scheme among all random variable because the corresponding variance is minimized in such case. But this might be not true since the second term of (4.13) is relatively small, compared with the first term for most images. So which random generator is used to obtain ϕ_i is not critical to the performance of a watermarking scheme as long as it has zero mean and its variance is one. This is demonstrated by Table 4.3, where we compare the responses of the watermark detector to the correct watermark and others, and the meaning of the parameters *mean*, *std* and *peak* is same as in Table 4.2. No big difference is observed for different random distributions.

	mean	*std*	*peak*
normal	28.23	780.41	24118.1
flat	152.7	870.54	24118.09
bilevel	82.11	804.35	23425.39

Table 4.3: Comparison of the responses of the detector to the correct watermark and the others for different statistic distributions.

Chapter 5

Conclusion

In this thesis, we have given a brief survey on the state of the art for digital watermarking for images and implemented several watermarking schemes based on DCT coefficient manipulation.

From the survey, we notice that there are a large number of publications in image watermarking, but most of them such as patchwork, 2D m-sequence and spread spectrum share similar concepts and consider digital watermarking as communication in non-Gaussian noise. In general, small, pseudo-random alternations are applied to certain coefficients in the spatial or spectral domain, and these alternations can later be recovered by correlation or correlation-like similarity measures. As we have noticed, the choice of the watermarking embedding domain is critical to the watermark robustness. Spatial domain schemes are usually less robust towards noise-like attacks, such as lossy JPEG compression, but its big advantage is that the watermark may be easily retrieved if the image is cropped or translated. On the other hand, the spectral

domain watermarking schemes are in general very robust to the noise-like attacks, but can't survive cropping and translation attacks. In fact, cropping or translation in the spatial domain results in a substantially large distortion in the frequency domain which usually destroys the embedded watermark.

In Chapter 3, we presented several digital watermarking schemes not based on spread spectrum technique, but on spectral features. The first scheme is a variant of the watermarking scheme proposed by Langelaar, et al. for JPEG/MPEG streams, based on selectively discarding high frequency DCT coefficients. Our scheme can survive JPEG compression with medium quantization factor and other noise-like attacks such as blur motion, sharpening, supernova, etc. In addition, unlike the original scheme, this modified scheme always works independent on the local image properties. Another watermarking scheme embeds the watermark bits into an image by modifying the rounding rule for the quantized coefficients. We found that this simple scheme can also survive JPEG compression and its robustness can be improved by introducing a self-reference pattern. The pairing scheme tries to remember some charastic of an image depending on the watermark in a separate file and tries to keep the original image intact. So this scheme provides a good visual imperceptibility, but is sometime inconvenient since a secret separate file is needed during the decoding process. A common disadvantage of these three schemes is as we discussed before that the watermarks are not robust to the cropping and translation attacks and for JPEG attack with a very low quantization factor the confidence measure is very unreliable.

Our watermarking algorithm using the spread spectrum technique modulates each watermark bit into certain coefficient of each 8×8 DCT block in the whole image. Since the discrete cosine transform is performed locally to each block, our scheme can survive cropping or translation attack (as long as we know the block positions) as well as the JPEG compression. In addition, we can embed several watermarks into an image and later on retrieve them without interference. Although less information may be embedded into an image, this scheme can survive more severe lossy JPEG compression attack, e.g., with quantization factor as low as 16 without visual distortion.

The work of this thesis is certainly not complete. Some of the future problems I plan to work on are:

- **Human Visual System**. For maximal robustness it is important to put as much energy into the watermark as possible under the constraint that the watermark remains invisible. So in order to enhance the robustness of the watermark, the characteristic of the Human Visual System is planed to be exploited to adapt the watermark to the image being signed. For example, we may design the weight function α in (4.1) such that its energy is maximized subject to a required maximal acceptable distortion.

- **Reed-Solomon code**. It is inevitable that there will be some degradation to the embedded watermark when the host image is attacked. In order to compensate for errors due to the channel noise and host image modifications, we believe that it is helpful to apply forward error correction codes such as Reed-Solomon code [36, 2] to the watermark being

embedded. By embedding a relatively large amount of data into the host image, the data (watermark) integrity may be ensured.

- **Video and Audio**. Finally, we wish to apply our watermarking algorithms to other forms of multimedia such as audio and video with a minimum amount of perceivable degradation.

Bibliography

[1] W. Bender, D. Gruhl, N. Morimoto and A. Lu, *Techniques for data hiding.* IBM Systems Journal, vol. 35, no. 3, (1996), pp. 313–336.

[2] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1984.

[3] F. Boland, J. O'Ruanaidh and D. Dautzenberg, *Watermarking digital images for copyright protection*, In Proceedings of the International Conference on Image Processing and its Applications, Edinburgh, Scotland, July 1995, pp. 321–326.

[4] A. Bors and I. Pitas, *Image watermarking using DCT domain constraints.* In Proceedings of the International Conference on Image Processing , Lausanne, September 1996, pp. 232–234.

[5] S. Burgett, E. Koch and J. Zhao, *A novel method for copyright labeling digitized image data*, Technical report, Frauenhofer Institue for Computer Graphics, Darmstadt, Germany, September, 1994.

[6] G. Caronni, *Ermitteln unauthorisierter Verteiler Von maschinenlesbaren Daten*, Technical report, ETH Zürich, Switzerland, August, 1993.

[7] G. Caronni, *Assuring ownership rights for digital images*, Proc. Reliable IT Systems, VIS'95, 1995, pp. 251–263.

[8] I. Cox, J. Kilian, T. Leighton and T. Shamoon, *Secure spread spectrum watermarking for images, audio, and video.* In Proceedings of the International Conference on Image Processing 1996, pp. 243–246.

[9] P. Davern and M. Scott, *Fractal based image steganography*, In Lecture notes in computer science: Information Hiding, vol. 1174, Springer, May/June 1996, pp. 279–294.

[10] R. Dixon, *Spread spectrum system with commercial applications*, John Wiley and Sons, New York, 1994.

[11] J. Hernandez, F. Perez-Gonzalez, J. Rodriguez and G. Nieto, *Performance analysis of a 2-D multi-pulse amplitude modulation scheme for data hiding and watermarking still image*, IEEE Journal on Selected Areas of Communications, 16(4), 1998, pp. 510–524.

[12] D. Kahn, *The Codebreakers: The Story of Secret Writing*, New York, New York, U.S.A.: Scribner, 1996, ISBN 0-684-83130-9.

[13] K. Knox and S. Wang, *Digital watermarks using stochastic screens — a halftoning watermark*, In Proceedings of the SPIE International Conference on Storage and Retrieval for Image and Video Databases V, San Jose, CA, 1997, vol. 3022, pp. 310–316.

[14] E. Koch, J. Rindfrey and J. Zhao, *Copyright protection for multimedia data*, Digital Media and Electronic Publishing, 1996

[15] E. Koch and J. Zhao, *Towards robust and hidden image copyright labeling*, In Proceedings of the Workshop on Nonlinear Signal and Image Processing, Marmaros, Greece, June, 1995.

[16] D. Kundar and D. Hatzinakos, *A robust digital image watermarking method using wavelet fusion.* In Proceedings of the International Conference on Image Processing , 1997, pp. 544–547.

[17] D. Kundar and D. Hatzinakos, *Digital watermarking using multiresolution wavelet decomposition*, International Conference on Acoustic, Speech and Signal Processing (ICASSP), vol. 5, Seattle, WA, USA, 1998, IEEE, pp. 2969–2972.

[18] D. Kundar and D. Hatzinakos, *Digital watermarking for telltale tamper-proofing and authentication*, Proceedings of the IEEE — Special Issue on Identification and Protection of Multimedia Information, vol. 87, no. 7, July 1999, pp. 1167–1180.

[19] M. Kutter, *Digital image watermarking: hiding information in images.* Doctoral thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1999.

[20] M. Kutter, F. Jordan and F. Bossen *Digital watermarking of color images using amplitude modulation*, J. of Electronic Imaging, 7(2), 1998, pp. 326–332.

[21] G. Langelaar, R. Lagendijk, J. Biemond, *Watermarking by DCT Coefficient Removal: A statistical approach to optimal parameter settings*, In Proceedings of the SPIE Security and Watermarking of Multimedia Contents, San Jose, California, January 25-27, 1999, pp. 1-13, ISBN: 0-8194-3128-1, SPIE, Bellingham, Washington, Eds: P.W. Wong, E.J. Delp.

[22] E. L. Lehmann, *Testing statistic hypotheses*, Wiley, 1987.

[23] H.D. Lüke, *Korrelationssignale*, Springer, 1992.

[24] M. Maes and C. Overveld,*Digital watermarking by geometric warping*, In Proceedings of the International Conference on Image Processing (ICIP), vol. 1, 1998.

[25] N. Nikolaidis and I. Pitas, *Copyright protection of images using robust digital signatures*, In Proceedings ICASSP 96, May 1996

[26] A. Papoulis, *Probability and Statistics*, Prentice Hall, 1991.

[27] F. Petitcolas, R. Anderson and M. Kuhn, *Information hiding — survey.* In Proceedings of the IEEE, special issues on protection of multimedia content, 87(7), (1999), pp. 1062–1078.

[28] F. Petitcolas, R. Anderson and M. Kuhn, *Attacks on copyright marking systems.* Lecture notes in computer science, vol. 1525, (1998), pp. 218–238.

[29] B. Pfitzmann, *Information hiding terminology*, In First International Workshop on Information Hiding, R. Anderson (ed), 1996, pp. 347–350.

[30] I. Pitas, *A method for signature casting on digital images.* In Proceedings of the International Conference on Image Processing , 1996, pp. 215–218.

[31] I. Pitas and T. Kaskalis, *Applying signatures on digital images*, In IEEE Workshop on Nonlinear Image and Signal Processing, pp. 460–463, Neos Marmaras, Greece, June 1995.

[32] A. Piva, M . Barni, F. Bartolini and V. Capperllini, *DCT-based watermark recovering without resorting to the uncorrupted original image.* In Proceedings of the International Conference on Image Processing, 1997, pp. 520–523.

[33] C. Podilchuk and W. Zeng, *Watermarking of the JPEG bitstream*, In Proceedings of the International Conference on Imaging Science, Systems and Technology, Las Vegas, Nevada, 1997, pp. 253–260.

[34] C. Podilchuk and W. Zeng, *Perceptual watermarking of still images*, In Proceedings of the workshop on multimedia signal processing, Princeton, New Jersey, USA, June 1997.

[35] J. Puate and F. Jordan, *Using fractal compression scheme to embed a digital signature into an image*, In Proceedings of the SPIE, Video Techniques and Software for Full-Service Networks, vol. 2915, Boston, USA, 1996, pp. 108–118.

[36] I. Reed and G. Solomon, *Polynomial codes over certain finite fields*, J. Soc. Indust. Appl. Math., Chapter 8, 1960.

[37] S. Roche, J.-L Dugelay and r. Molva, *Multi-resolution access control algorithm based on fractal coding*, In Proceedings of the International Conference on Image Processing , 1996, pp. 235–238.

[38] J. Ruanaidh, F. Boland and O. Sinnen, *Watermarking digital images for copyright protection*, In Proceedings of the Electronic Imaging and the Visual Arts, Florence, Italy, February 1996.

[39] J. Ruanaidh, W. Dowling and F. Boland, *Phase watermarking of digital images*, In Proceedings of the International Conference on Image Processing, vol. 3, 1996, pp .239–242.

[40] J. Ruanaidh and T. Pun, *Rotation, scale and translation invariant digital image watermarking.* In Proceedings of the International Conference on Image Processing , 1997, pp. 536–539.

[41] G. Schott, *Schola steganographica: in classes octo distributa...* Jobus Hertz, printer, 1680, bound with: Casparis Schotti... Technical curiosa ... Herbipoli, 1665.

[42] R. Schyndel, A. Tirkel and C. Osborne, *A digital watermark*, In Proceedings of the International Conference on Image Processing (ICIP), vol. 2, IEEE, 1994, pp. 86–89.

[43] M. Simon, J. Omura, R. Scholtz and B. Levitt, *The spread spectrum communications Handbook*, McGraw-Hill, New York, 1994.

[44] J. Smith and B. Corniskey, *Modulation and information hiding in images.* Lecture notes in computer science, 1174, 1996, pp. 207–226.

[45] M Swanson, B. Zhu and A. Tewfik, *Robust data hiding for images.* IEEE DSP workshop, 1996, pp. 37–40.

[46] Y.N.K. Tanaka and K. Matsui, *Embedding secret information into a dithered multi-level image*, Proc. IEEE military communication conference, 1990, pp. 216–220.

[47] A. Tirkel, G. Rankin, R. van Schyndel, W. Ho, N. Mee and C. Osborne, *Electronic water mark*, In Proceedings DICTA 1993, December 1993, pp. 666-672.

[48] A. Tirkel, R. Schyndel and C. Osborne, *A two dimensional watermark*, In Proceedings DICTA 1995, 1995.

[49] S. Walton, *Information authentication for a slippery new age*, Dr. Dobbs Journal, vol. 20, no. 4, 1995, pp. 18–26.

[50] H.-J. Wang, P.-C Su and C.-C.J. Kuo, *Wavelet-based blind watermark retrieval technique*, In Symposium on Voice, Video, and Data Communications, Boston, Massachusetts, November 2–5, 1998.

[51] R.B. Wolfgang and E.J. Delp, *A watermark for digital images*, Proceedings of the International Conference on Image Processing, pp. 219–222, Lausanne, Switzerland, 1996, IEEE.

[52] R.B. Wolfgang and E.J. Delp, *A watermark techniques for digital imagery: Further studies*, In Proceedings of the Imaging Science, Systems and Technology, Las Vegas, 1997, pp. 279–287.

[53] X. Xia, C. Boncelet and G. Arce, *A multi-resolution watermark for digital images.* In Proceedings of the International Conference on Image Processing, 1997, pp. 548–551.

[54] M.M. Yeung and F.C. Mintzer, *Invisible watermarking for image verification*, Journal of Electronic Imaging, 7(3), 1998, pp. 578–591.

[55] W. Zeng and B. Liu, *On resolving rightfull ownerships of digital images by invisible watermarks*, In Proceedings of the IEEE International Conference on Image Processing, 1997, Santa Barbara, CA, vol. 1, pp. 552–555.

[56] The Gnu Image Manipulation Program (GIMP) Home Page: *http://www.gimp.org*