WATERMARKING CAPACITY IMPROVEMENT BY LOW DENSITY PARITY CHECK CODES

by

Ahmet BAŞTUĞ

B.S. in E.E., Middle East Technical University, 1999

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronics Engineering

Boğaziçi University

2002

WATERMARKING CAPACITY IMPROVEMENT
BY LOW DENSITY PARITY CHECK CODES

APPROVED BY:

Prof. Bülent SANKUR (Thesis Supervisor)	.....
Assoc. Prof. Levent ARSLAN	.....
Prof. Ümit AYGÖLÜ	.....
Assoc. Prof. Ayşin ERTÜZÜN	.....
Assist. Prof. Mehmet KESKİNÖZ	.....

DATE OF APPROVAL:

To my one-year-old son Hüseyin

ACKNOWLEDGEMENTS

I would like to thank Prof. Bülent SANKUR for his support, advice and encouragement during my thesis study.

I would like to express my appreciation to my wife Elif. Without her support and patience, I wouldn’t have completed this thesis.

ABSTRACT

WATERMARKING CAPACITY IMPROVEMENT BY LOW DENSITY PARITY CHECK CODES

Digital image watermark is an imperceptible, robust, secure message embedded into the image, which identifies one or more of the owner, distributor or recipient of the image, origin or status of the data or transaction dates. Watermarking is also used for data hiding, content labeling, broadcast monitoring and integrity control applications. Digital image watermarking resembles communication systems. Watermark is the sent message. Image is the watermark channel or carrier. Image pixels and possible attacks on the marked image constitute the noise. Only the authorized parties extract the watermark message from the marked image by using detectors.

Digital image watermarking has three major requirements. Watermark should be robust against noise and attacks, imperceptible and carry the required number of bits. These three requirements conflict with each other. To illustrate, increasing the watermark strength makes the system more robust but unfortunately decreases the perceptual quality. As a second example, increasing the number of embedded bits increases the capacity but decreases the robustness.

In this thesis, the goal was to investigate the contribution of the error correcting codes. More specifically, we studied the error correcting codes as a means to increase the watermarking capacity of an image or conversely to decrease the embedding strength, hence to decrease the visual impact of the watermark. We had two watermark channel models, namely the DFT domain and the 8x8 block DCT domain. First, we compared the performance of maximum likelihood (ML) detector vis-à-vis correlation and covariance detector. Second, we compared the performance of LDPC codes vis-à-vis BCH codes and pure repetition codes. We showed that ML detectors are slightly better than covariance detectors and LDPC codes outperform BCH codes and repetition codes by a large margin.

ÖZET

DAMGALAMA KAPASİTESİNİN DÜŞÜK YOĞUNLUKLU HATA DENETİM KODLARI KULLANIMIYLA ARTIRILMASI

Sayısal imge damgası, imgeye eklenen, imgenin sahibi, dağıtıcısı, müşteri kimliği, kaynağı, statüsü ve geçirdiği değişimlerden bir veya birden fazlası hakkında bilgi taşıyan, fark edilmez, saldırılara karşı dayanıklı, güvenli mesajdır. Damgalama, ayrıca, bilgi saklama, içerik etiketleme, yayın denetimi ve bütünlük kontrolü uygulamalarında da kullanılır. Sayısal damgalama iletişim sistemlerine benzer. Damga iletilen mesajdır. İmge damgalama kanalıdır. İmge pikselleri ve olası saldırılar gürültüyü oluşturur. Sadece yetkili şahıslar, kestiriciler kullanımıyla, damga mesajını damgalanmış imgeden çıkartabilirler.

Sayısal imge damgalamada üç önemli gereksinim vardır. Damga gürültü ve saldırılara karşı dayanıklı olmalı, fark edilmez olmalı ve gereksinim duyulan sayıda ikili taşımalıdır. Bu üç gereksinim birbiriyle çelişir. Örneğin, damgalama gücünü artırmak sistemi daha dayanıklı yapar; ama görüntü kalitesini düşürür. İkinci bir örnek olarak, eklenen ikili sayısını artırmak kapasiteyi artırır ama dayanıklılığı azaltır.

Bu yüksek lisans tezinde, seçip uyguladığımız iki damgalama yönteminin kapasitesini artırmayı, bir başka deyişle sezim sırasındaki bit hata oranını düşürmeyi amaçladık. Bu yöntemlerden biri DFT ortamında diğeri de 8x8 blok DCT ortamında çalışıyor. Kapasiteyi artırmak için korelasyon sezicileri yerine en iyi olasılıklı (**M**aximum **L**ikelihood) sezicileri ve damgalama öncesinde Düşük Yoğunluklu Hata Denetim (**L**ow **D**ensity **P**arity **C**heck) kodlarını uyguladık. ML sezicilerinin performanslarını korelasyon sezicilerininkilerle ve LDPC kodlarının performanslarını Bose-Chaudhuri-Hochquenghem (BCH) kodları ve salt yineleme kodlarınınkilerle kıyasladık. ML sezicilerinin korelasyon sezicilerinden ve LDPC kodlarının da BCH kodlarından ve salt yineleme kodlarından daha iyi olduklarını gösterdik. ML sezicilerini ve LDPC kodlarını birlikte kullanarak, hata miktarını oldukça düşürdük.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

A	amplitude parameter in the transform distribution
b	message bit sequence of length T_k
b(i)	i^{th} bit in **b**
c	code bit sequence of length T_n
$\mathbf{c_c}$	concatenated code bit sequence of length T_c
ch	chip sequence length for one code bit
c(i)	i^{th} bit in **c**
d	length of code word in LDPC coding
G	code generator matrix
H	code parity check matrix
I	transformed (DFT, 8x8 block DCT) image
$\tilde{\mathbf{I}}\mathbf{(i)}$	image transform coefficients corresponding to c(i)
$\mathbf{I_w}$	transform coefficients of the watermarked image
$\tilde{\mathbf{I}}_\mathbf{w}\mathbf{(i)}$	watermarked transform coefficients hosting c(i)
k	length of message word length for (n,k) block coding
K_1	key used to generate the spreading code
K_2	key used to distribute watermark to transform coefficients
L	# of skipped low DFT coefficients in each quadrant
M	# of marked medium DFT coefficients in each quadrant
n	length of code word for (n,k) block coding
N	image size for an NxN image
N_b	# of 8x8 blocks in an NxN image = $N^2/64$
p	channel raw error rate
p_b	bit error rate after error correction coding
$\mathbf{p_c}$	pseudo-random sequence generated by key K_1
$\tilde{\mathbf{p}}_\mathbf{c}\mathbf{(i)}$	spread spectrum sequence of coded bit c(i)
pdf	probability density function
P	parity check matrix
t	error correction capability of BCH codes

T_c	# of marked coefficients =length of concatenated coding
T_k	length of source message
T_n	length of coded source message
$\mathbf{X}$	image matrix in spatial domain
$\mathbf{X_w}$	watermarked image pixel matrix in spatial domain
$\mathbf{W}$	watermark image matrix
$\mathbf{w_c}$	spread modulated, concatenated code bit sequence = $\mathbf{c_c}.\mathbf{p_c}$
$\tilde{\mathbf{w}}_\mathbf{c}\mathbf{(i)}$	spread modulated, concatenated code bit sequence of c(i)
$w_c(i)_j$	j^{th} element of $\tilde{\mathbf{w}}_\mathbf{c}\mathbf{(i)}$
$\mathbf{W_m}$	watermark transform domain mask matrix
$\mathbf{z}$	syndrome vector
α	scale parameter in the transform distribution
β	shape parameter in the transform distribution
γ	watermark strength
Γ	gamma function
Λ	likelihood ratio
η	decision threshold
A/D	analog to digital
BCH	Bose Chaudhuri Hochquenghem
BER	bit error rate
CD	compact disc
D/A	digital to analog
DCT	discrete cosine transform
DFT	discrete Fourier transform
DS-CDMA	direct sequence code division multiple access
DVD	digital vieratile disc
DWT	discrete wavelet transform
FDMA	frequency division multiple access
HVS	human visual system
IDCT	inverse discrete cosine transform
IDFT	inverse discrete Fourier transform

ISRC	international standard recording code
LDPC	low density parity check
ML	maximum likelihood
OSI	open systems interface
PSNR	peak signal to noise ratio
PDMA	pixel division multiple access
TDMA	time division multiple access
VCD	video compact disc
WDR	watermark to document ratio

1. INTRODUCTION

1.1. Digital Media: Its Advantages and Disadvantages

Digital storage and transmission is the major trend of handling information. The audio, image and video industries are distributing their products in digital form. Broadcast televisions, big corporations and photo archives are converting their content from analog and other forms to digital. With the increasing availability of a lot of advanced multimedia broadcasting services such as pay-per-view, video-on-demand, tele-marketing, tele-teaching, electronic newspapers, tele-gaming, electronic commerce, advertising, interactive TV, digital libraries and web magazines, this trend will further increase [1].

Digital technology has many superior properties as compared to the analog technology. First of all, the quality of digital audio, image and video is superior to that of analog form due to noise free transmission. Secondly, it is easier to process and distribute digital media. Therefore, most of the multimedia applications exploit digital technology. On the other hand, digital media has the disadvantage that it yet lacks good copyright protection mechanism. Since the unauthorized reproduction, distribution and manipulation of digital media are very easy, the authorized service providers are reluctant to offer commercial services in digital form [2].

1.2. Copyright Protection of Digital Media

In order to provide copy protection and copyright protection for digital multimedia, two complementary techniques are proposed as shown in Figure 1.1 in the context of a communication system. **Watermarking** which can be put into the physical layer of well-known layered data communications models such as OSI serves the **encryption,** which can be put at the application layer.

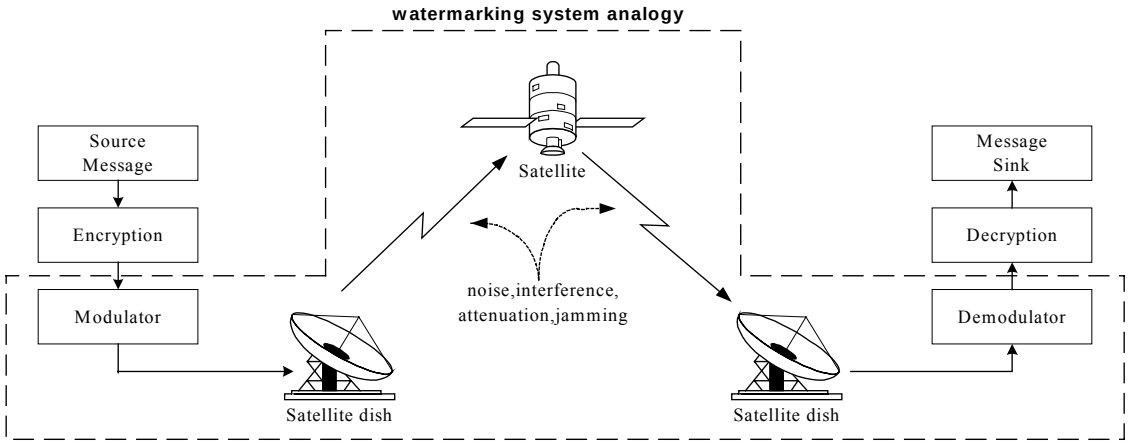

Figure 1.1. Encryption for message protection and watermarking for media protection

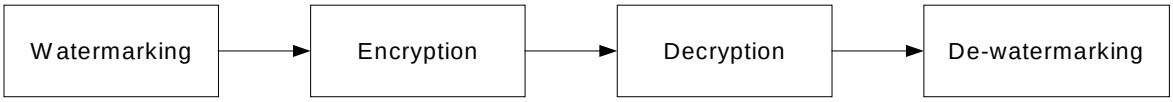

Figure 1.2. Encryption and watermarking for media protection

Encryption is applied as a first security measure. In this technique, the data is scrambled at the transmitter using a secret key. The authorized receiver is the only other party, which is intended to know this secret key in order to descramble the received data. Since, on the average, it takes years for a spy computer to find out the key, encryption looks like a safe technique. However, since the computers are getting more and more speedy and it is possible to use multi-processor, multi-computer, multi-human systems, it is getting easier to break the encryption protection and do illegal reproduction and distribution of valuable digital media. Moreover, once the receiver has received and decrypted the data, it is no longer protected. For these reasons, encryption is not a perfect solution for copyright protection. In fact, it is mainly concerned with secure communications but not copyright protection.

Encryption can be used in two ways: to protect the message or to protect the media. The encryption scheme shown in Figure 1.1 protects the message but not the media and hence it is useless. The encryption scheme shown in Figure 1.2, however, protects the media and hence it is useful for secure communications of the media.

The second security measure, which is newly emerging and known as watermarking, is applied as a last line of defense. Watermarking can not by itself prevent copying, modification and redistribution of digital media; but if encryption fails to do so during transmission, watermarking allows the document to be traced back to its rightful owner and to the point of unauthorized user after the delivery of the data [3].

The enclosed region in Figure 1.1 is an analogy of the watermarking system. It consists of the watermark embedding module (modulator), the marked media (noise, interference), attacks (attenuation, noise, jamming), the distribution system (satellite system for this example) and the watermark detector (demodulator).

1.2.1. Watermarking as a Last Means of Solution for Copyright Protection

Watermarking is a new technology based on the combination of many different fields such as cryptography, communication theory, information theory and signal processing. A digital watermark is an imperceptible, robust, secure message embedded into the document. Watermark message identifies at least one of the media owner, the distributor of the media, the recipient of the media, the origin or status of the data or the transaction dates. It is hidden in the media in such a way that it is not noticed and it always exists in the media no matter what type of processing the media experiences both intentionally and unintentionally.

To completely span the copyright requirements, three sets of watermark information are necessary [1]:

Stamping number: It is an identification number like ISBN. An independent authority is responsible of keeping record of it. The aim of using it is to identify a work for proof of ownership in case illegal copies are distributed. In other words, it is used for copyright protection.

Terminal identification number: It is the identity of the customer and is used to detect the originator of illegal copies. This application is also known as fingerprinting.

Right to copy flag: It is a flag indicating if copying is permitted or not. It is also called copy protection and is used in recorder devices to prevent illegal mass copying.

There is a serious problem though with the copy protection [1]. Let's build a scenario about the encryption, playing control and recording control of CD/VCD/DVD content. Today, watermarking technology is not mandatory in the player and recorder devices and most probably it will not be either in the future. Therefore, there will be both watermark *compliant* and *noncompliant* devices. A legal encrypted copy of a work can be played on a compliant player but not on a noncompliant player, for the noncompliant player can not decrypt it. The output of the compliant player can not be recorded on a compliant recorder since the recorder would detect the watermark. However, such output can be recorded on a noncompliant recorder. This will result in a decrypted illegal copy of the work. This copy can be played on a noncompliant player but can not be played on a compliant player since the latter would detect the watermark and prohibit playback. In brief, the customer has a choice of either buying a compliant device to play legal content but not pirated ones or a noncompliant one to play pirated content but not purchased one. Unfortunately, there is nothing to do if one has both compliant and noncompliant devices.

Combining the mentioned three watermarking messages, "right to copy flag" is used to possibly prevent copying, "terminal identification number" to detect copyright violators and "stamping number" to prosecute them.

1.3. Applications of Watermarking

Watermarking is not limited to copyright protection and copy protection. It is also used in the following fields [1]:

1.3.1. Content Labeling And Hidden Annotations

Watermarking can be used in content labeling, multimedia indexing and transaction tracking, usage control, access level control and medical applications. For example, a digital camera can hide the date and place of the taken photo which is in the category of content labeling. As another example, in a medical application, the watermarking system

can embed patient records directly into radiography images in such a way to speed up the access to records and to prevent errors of mismatching between patient records and images.

1.3.2. Broadcast Monitoring

Advertisements, TV programs and news items have high commercial value. The existence of embedded watermarks in them enables an automated broadcast surveillance monitoring system to check whether they are broadcasted as contracted and by the authorized source.

1.3.3. Data Hiding

Watermarks can also be used to hide secret private messages. This is also known as steganography. In this type of application, robustness is not of much concern. Because, the assumption is, third parties are not aware of the existence of the watermark in the media. Therefore, capacity of this application can be up to the limit of creating awareness of its existence.

1.3.4. Integrity Control

In some applications such as news pictures, it is important to be sure that the content of the media has not changed since its distribution. As a verification mechanism, the detector compares the extracted watermark with the embedded one. If they do not match, it means that the content has been modified. This application is different than all the others since in this case, the watermarking system should be non-robust. Such a watermark is called fragile watermark and it should disappear if the media experiences any intentional attack. However, this does not mean that it shouldn't be robust to common signal processing such as A/D-D/A conversions and JPEG compression.

1.3.5. Device Control

Devices can be controlled by watermarked content included in the same device or in other devices. Previously mentioned copy control falls in this category. There are some

other applications. In 1989, a technique was devised that allows action toys to interact with television programs [1]. In this technique, a simple video watermark modulates the intensities of horizontal scan lines within each frame of the video. A light-sensitive device placed near the television detects this modulation signal and transmits a high-frequency infrared signal, which synchronizes the actions of the interactive toys nearby. In a more recent application, a unique identifier is embedded into printed and distributed images such as magazine advertisements, tickets and so on. After the image is captured by a digital camera, the watermark is read by a software on a PC and the identifier is used to direct a web browser to an associated website.

1.4. Problem Statement

Low SNR is a phenomenon of watermarking channels, which severely limits the capacity. For steganographic message or meta-data applications, many bits like hundreds or even thousands may be needed [4]. In this thesis, we investigated increasing the watermark hosting capacity of still images by the application of error correction codes, particularly low density parity check (LDPC) codes. Though introduced long before by Gallager in 1962, the use of LDPC codes was not understood before MacKay rediscovered them [5]. LDPC codes were not practical until lately because of their complexity and computational difficulties. They are known to be the best codes performing very near to the Shannon limits.

1.5. Thesis Outline

The rest of the thesis is organized as follows: The second chapter covers digital image watermarking in general. It consists of requirements, system model, methodology and classification of digital image watermarking. The third chapter explains the implemented watermarking techniques. The fourth chapter is devoted to coding, in particular LDPC codes. Both the theory and the practical implementation of LDPC codes are explained. Simulation results are given in chapter 5. The sixth chapter is devoted to conclusions. The images used throughout the simulations and the tabular simulation results are given in Appendix A and Appendix B, respectively.

2. DIGITAL IMAGE WATERMARKING

2.1. Image Watermarking Requirements

Digital watermarking, particularly digital image watermarking, has several conflicting requirements. The three most important requirements are robustness, imperceptibility and capacity [1].

The most demanding requirement is the perceptual transparency. Watermark has to be embedded in such a way that the quality of the underlying host image should not be degraded much. As an acceptable measure, the modifications should be unnoticed as long as the marked image is not compared with the original one. The second important requirement is the robustness against intentional and unintentional degradations. In other words, it is desired that the watermark always remain in the host data. The unintentional effects can be compression, filtering, A/D & D/A conversion, re-sampling, etc. The intentional attacks can be the attempts to remove, alter or jam the watermark information. In brief, a robust watermark is impossible or very difficult to defeat without degrading the quality of the marked document to such an extent that the document is no more useful or has no commercial value. The third requirement is capacity. The watermarking technique should be capable of hiding the necessary amount of watermark data in the host media without degrading the visual quality too much. There are several other requirements such as simple and fast embedding and recovery of the watermark operation without original document, etc.

The requirements listed above almost always conflict with each other. For example, a very robust watermark can be obtained by highly modifying the host data for each bit of the watermark by increasing the watermark strength. However, this large modification will be perceptible. As a second example, increasing the number of embedded bits increases the capacity but decreases the robustness. Therefore, the maximum amount of modification that can be acceptable for the quality of the media and robustness are the two determining factors for the maximum amount of watermark bits that can be stored in a data object.

2.2. Watermarking System Model

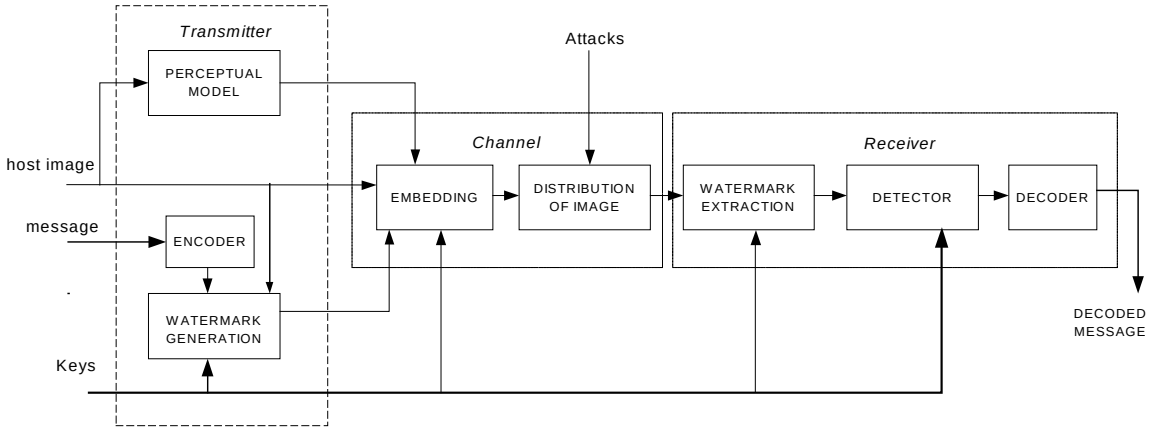

Figure 2.1. Digital image watermarking system model

A general digital image watermarking system model is shown in the Figure 2.1. It is very similar to communication systems and hence image watermarking should be seen as a data communications problem. Message encoding and watermark generation stages resemble source processing. Watermark embedding can be associated with modulator. Distribution of the media and the concomitant attacks are channel phenomenons. Watermark extraction is like demodulation and finally detection and message decoding are like post-reception operations. The similarities of watermarking and communications systems are demonstrated in Figure 2.2 and Figure 2.3.

Image watermark generation systems have an advantage over common communication systems. In communication systems, the characteristics of the channel are not known in advance. One can only assume that the used channel fits a known channel model. The channel characteristics change all the time and the transmitter needs to estimate these periodically. For the watermarking case, however, the cover signal, that is the image, is the channel and the transmitter knows it. One can exploit the image characteristics to adaptively embed the watermark as demonstrated with an additional connection from the image to the embedding block in Figure 2.3.

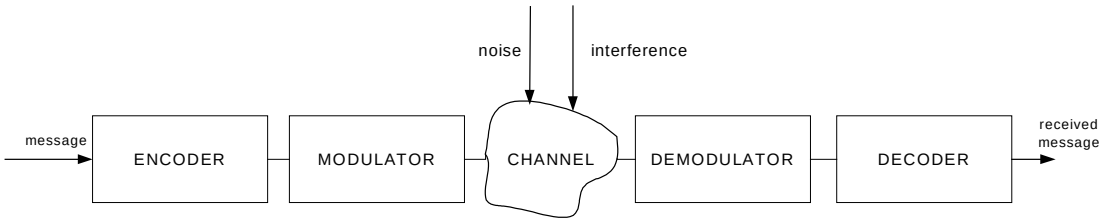

Figure 2.2. Classical communications system model

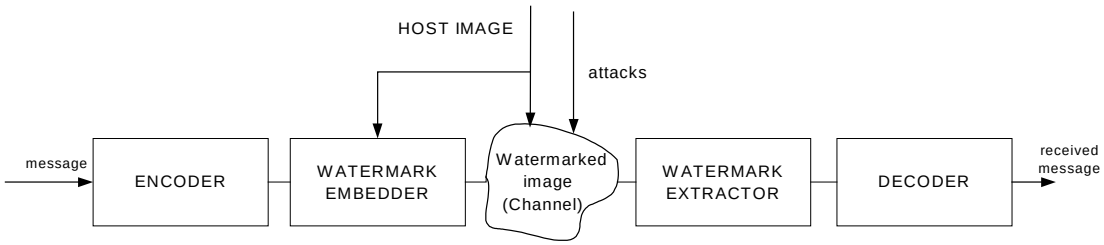

Figure 2.3. Watermarking system

2.2.1. Watermark Generation

The watermark message is a logo or in most cases a series of binary data of a given length. For image copyright applications, the common practice is that it should be around 60-80 bits. This is necessary to embed the International Standard Recording Code (ISRC), year of copyright and the granted permissions on the work. For access control applications, the required number of bits is much less like one or a few bits. For meta-data applications, though, much more bits like 300–400 bits or even thousands are required.

The generated message is optionally passed through error correction coding.

2.2.2. Watermark Embedding

The coded sequence is spread spectrum modulated using a spreading sequence, which is a continuous or discrete random noise sequence generated by using a **key**. The spread spectrum watermark signal is embedded into the image by modifying the image characteristics such as luminance values or transform domain coefficients. Selection of the coefficients depends on perceptual criteria as well as on a **key** instrumented permutation to

increase the security and robustness of the system. Embedding can be done in an image dependent/independent additive manner or by some substitution mechanisms.

It is often necessary to utilize Human Visual System (HVS) models for adaptively embedding the watermark. This can reduce the impacts of the modifications on image quality or for the same visual quality a much stronger watermark can be embedded.

The human eye is sensitive to the following characteristics of the image [9,10,11]:

- **Contrast:** This is the sensitivity of the eye to a signal in the presence of another signal.
- **Frequency:** Eye is more sensitive to some frequencies, particularly low frequencies, than others.
- **Luminance Sensitivity:** This is a measure of detection threshold on a constant background. Alterations to low and high luminance values are more noticeable.
- **Edges and Textured Areas:** Around edges and textured areas, the HVS is less sensitive to distortions than smooth areas.

One can combine the above four properties to construct a perceptual mask which determines the amount of modification permitted on each image cover data (pixels, transform coefficients) value. Using perceptual masks, energy can be increased locally in places where the human eye can't notice it. This increases robustness and hence capacity.

2.2.3. Watermark Extraction

After the watermarked image is subjected to the possible intentional and unintentional attacks, the watermarked coefficients are extracted using the same key used in the embedding stage.

2.2.4. Watermark Detection and Message Decoding

Once the marked coefficients are extracted, coded message is detected by using the same spreading sequence based on the same key used in the embedding stage. The detector

can be a correlator type, its variants or a maximum likelihood one. The despread coded bit sequence is then given to the decoder and the message bits are obtained. Watermarking channels can differ widely in their bit error performance, ranging from 10^{-10} and lower for access control applications to 30-40 % for data hiding applications. Therefore, it is not immediately obvious that channel coding would be beneficial in watermarking channels.

2.3. Classification of Watermarking Techniques

Watermarking techniques can be classified in terms of [2]:

- Processing domain as,
  - Spatial domain
  - Transform domains (DFT, DCT, DWT)
- Modification type as,
  - Additive
  - Additive-multiplicative
  - Substitution
- The availability of the original data as
  - Blind (oblivious)
  - Non-blind (non-oblivious)
- Privacy as
  - Private or asymmetric
  - Public or symmetric

2.3.1. Processing Domains

Spatial Domain Marking: In this technique, the pixel luminance and chrominance values are modified to embed the watermark. There are two embedding methods, which are also applicable to transform domain watermarking. As a first alternative, the image pixels can be divided into distinct sets. Each watermark bit modifies the set allocated for it. Then, the number of watermark bits embedded in the image will be equal to the number of sets. Defining this method as pixel division multiple access (PDMA), we can see it as time division multiple access (TDMA) or frequency division multiple access (FDMA) schemes

in common communication systems. An example implementation is shown in Figure 2.4. The grouping needn't be structured. The elements of sets can be randomly selected from the whole image by random permutation. As a second method, shown in Figure 2.5, all the watermark bits can modify the whole image, which can be seen as direct sequence code division multiple access (DS-CDMA). With the first method, there is no interference between the embedded bits and the detection is easier; but, if the image is cropped, the affected watermark bits located at the border areas are lost. In contrast, if DS-CDMA is used, the probability of recovering all bits after cropping the image is much higher; however the watermark bits may interfere with each other, and the detection is computationally more difficult.

Transform Domain Marking: In these techniques, the image undergoes a mathematical transformation before watermark casting is done [7,8]. After watermark is embedded using techniques similar to the ones used for spatial domain watermarking, the inverse transformation is applied to turn back to the spatial domain. The pixel-domain and the transform domain techniques are shown in Figure 2.6.

In this thesis, we are interested with the two well-known transform domains, namely DFT (Discrete Fourier Transform) and DCT (Discrete Cosine Transform).

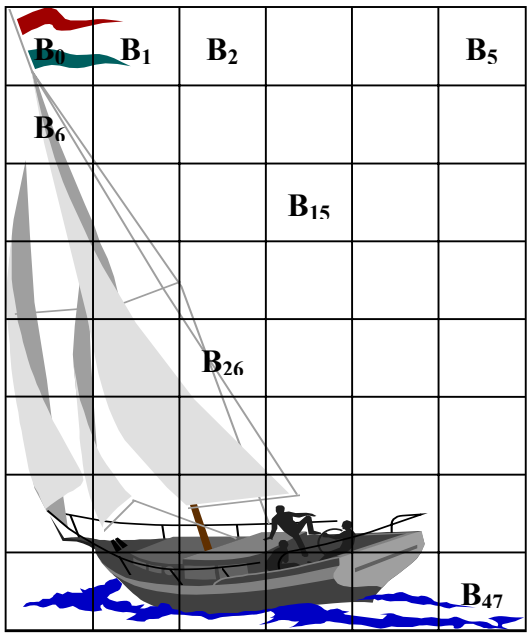

Figure 2.1. Structured block marking

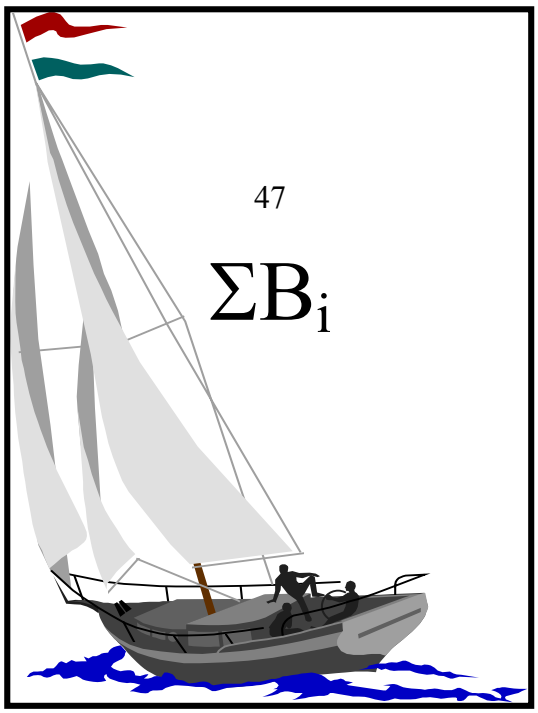

Figure 2.2. CDMA based marking

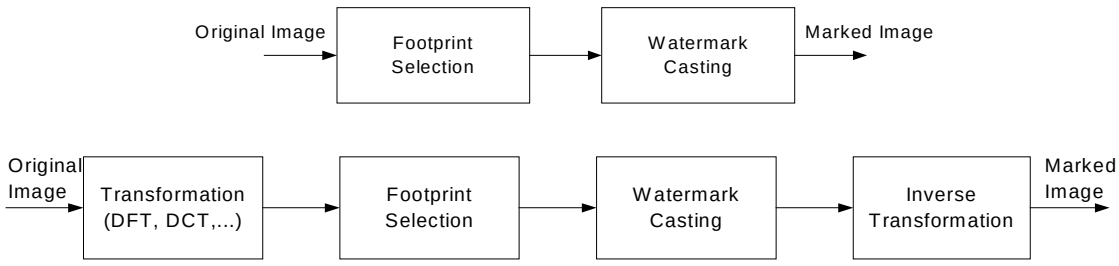

Figure 2.3. Pixel domain and transform domain marking techniques

DFT Domain Marking: DFT coefficients are complex numbers. Therefore there're two possible domains for DFT marking: phase and amplitude.

Advantages of using DFT phase are given as:

- It is advantageous to put the watermark in the most visible components of image for obtaining a more robust system. Because, once such a marked image is attacked, the quality of the image deteriorates to such an extent that, the image loses its commercial

value. DFT phase is more significant than DFT amplitude for the intelligibility of images, and hence it might be a good choice as watermark cover data.

- Phase modulation has “better noise immunity” than amplitude modulation.

There is however one serious disadvantage of using DFT phase as:

- DFT phase is not robust against translation attacks since translation in the spatial domain changes the phase of DFT coefficients.

The advantage of using DFT amplitude is that:

- Total image DFT amplitude is robust to **translation** or **shift** attacks in the spatial domain since cyclic translation of image in the spatial domain does not affect DFT amplitude.

The disadvantage of using DFT amplitude is:

- DFT amplitude does not have good noise immunity as compared to DFT phase.

DCT Domain Marking: DCT coefficients are real-valued. The advantage of using DCT is:

- An image is split up in pseudo frequency bands by DCT. Therefore watermark can easily be embedded in the middle band frequencies.

For the block DCT, the image is first partitioned into a number of blocks and DCT transform is applied on each block before watermark embedding.

The advantages of using block DCT are:

- Block DCT has perceptual models.

- Block DCT is widely used in compression schemes such as JPG. By anticipating which coefficients will experience loss during compression, one can mark in the surviving coefficients. This increases the robustness against compression.

The disadvantage of using DCT schemes is:

- DCT is sensitive to **translation** or **shift** attacks in the spatial domain. Before detection procedure, one needs to use synchronization techniques against these kinds of attacks to take back the image to its original orientation.

2.3.2. Watermark Embedding Styles

Let x be the cover image, x_w be the resulting modified coefficient, w be the added watermark and γ be the watermark strength. γ doesn't need to be constant over all coefficients. It can be varying to utilize the local properties of host image.

Additive Marking: In this method, the watermark sequence is added to a selected set of pixel or transform domain coefficients. The added quantities are white sequences such as Gaussian or uniform noise or ±1 sequence. The embedding formula is

$$x_w = x + \gamma w \tag{2.1}$$

Additive-Multiplicative Marking: In this method, the watermark data are added to a selected set of pixel or transform domain coefficients proportional to their magnitude. The embedding formula is

$$x_w = x + \gamma x w \quad \text{or} \quad x_w = x\, e^{\gamma w} \quad \text{or} \quad x_w = x + \gamma |x| w. \tag{2.2}$$

With this method, small coefficients are not modified much whereas large coefficients are severely affected. This is nice from the perceptual transparency point of view because HVS sensitivity is nearly constant with respect to the relative changes in coefficient magnitudes.

Substitution Marking: There are various substitution algorithms. We will briefly explain the one developed by Koch and Zhao [6].

The algorithm works on randomly selected 8x8 DCT coefficient blocks. Two coefficients from the mid-frequency range are randomly selected. Say f_b denotes the 8x8 DCT block and $f_b(m_1,n_1)$, $f_b(m_2,n_2)$ denote the selected coefficients. The absolute difference between the selected coefficients is given by

$$\Delta_b = | f_b(m_1,n_1) | - | f_b(m_2,n_2) | \tag{2.3}$$

To embed one bit w_i in the selected block, the coefficient pair $f_b(m_1,n_1)$, $f_b(m_2,n_2)$ is modified such that the distance becomes

$$\Delta_b = \begin{cases} \geq q & \text{if } w_i = 1 \\ \leq -q & \text{if } w_i = 0 \end{cases} \tag{2.4}$$

where q is a parameter controlling the embedding strength

2.3.3. Blind vs. Non-blind Watermarking

If the original image is available for the recovery of watermark, this is called non-blind (non-oblivious) watermarking. If it is not, the watermarking system is blind (oblivious). The non-oblivious case is surely more robust; however in real life, the original image may not be available for all applications.

2.3.4. Private vs. Public Watermarking

A technique is private, if the document owner is the only person who can extract the watermark for he is the only person who can either access the original image or who knows the code to look for. In other words, techniques, which allow the watermark to be detected knowing the content in advance, are called private techniques. The data owner decides whether a given watermark is present in the data or not.

Techniques, which permit anyone to read the watermark, are called public. In other words, the keys to detect or to read the watermark data are publicly available.

Private mechanisms are more robust than public ones. Because, once the keys are known, it is easy for an attacker to remove the watermark or make it unreadable with the public mechanism. Most of the commercial, so far non-robust, watermarks depend on public schemes whereas research focuses on private marking.

3. IMPLEMENTED WATERMARKING TECHNIQUES

3.1. Watermarking in the DFT Magnitude Domain

The first marking technique that we implemented works in a selected "medium frequency" region of the full frame DFT [7]. Two-dimensional Discrete Fourier Transform (DFT) of an image with size NxN is defined as

$$I(u,v) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} X(k,l) e^{-j\frac{2\pi}{N}(uk+vl)} \qquad u,v \varepsilon 0,1\ldots(N-1) \tag{3.1}$$

and inverse discrete Fourier transform (IDFT) for an image with size NxN is defined as

$$X(k,l) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} I(u,v) e^{j\frac{2\pi}{N}(uk+vl)} \qquad k,l \varepsilon 0,1\ldots(N-1) \tag{3.2}$$

Just like in any other transform domain marking, modification on any DFT coefficient affects the whole image. That is, the watermark is spread over all the pixel luminance values and this makes the watermark robust against cropping attacks. We embed the watermark on top of DFT magnitudes using the additive-multiplicative method explained in Chapter 2. Choosing DFT magnitudes adds extra robustness because DFT amplitudes are invariant to translation attacks. Since the DFT of an image translated by (k_0,l_0) is given by $DFT\{X(k-k_0,l-l_0)\} = \tilde{I}(u,v) = I(u,v)\, e^{-j2\pi uk_0} e^{-j2\pi vl_0}$, it follows that its magnitude is invariant:

$$|\tilde{I}(u,v)| = |I(u,v)| \tag{3.3}$$

Watermarking techniques in the DFT domain for the whole image have the disadvantage that the insertion strength can not be controlled locally since the watermark is spread over all the pixels of the image. However, this shortcoming can be overcome by using masking in the spatial domain after having marked the image in the DFT domain.

3.1.1. Selection of DFT Coefficients

All the DFT coefficients are not modified. The low frequency components of an image are perceptually the more significant ones and any modification on them deteriorates the image fidelity. Therefore, watermarking shouldn't be applied on low frequency components. On the other hand, the high frequency components are the ones, which are usually less significant in terms of fidelity. As a consequence, compression techniques utilize this property and suppress the high frequency components first to reduce the size of images. Therefore, the watermarking techniques that modify high frequency coefficients can not be robust carriers of watermark. This leaves us with the choice of bandpass coefficients.

If the horizontal and vertical frequency range is normalized to the interval (-1,1), a reasonable bandpass absolute frequency region for watermarking is between 0.25 and 0.55. Extending these limits towards lower frequencies increases the capacity but decreases the perceptual quality. In an NxN image, there are N^2 DFT coefficients. We designate the selected DFT coefficients skipping the lower L frequency coefficients and marking the next M coefficients at each of the four vertices of the 2-dimensional DFT matrix. This makes up totally 4*M coefficients. $L \approx (N/2)^2*0.25^2$ and $M \approx (N/2)^2*0.55^2$ - L. The selected DFT region for watermarking is shown in Figure 3.1. The watermark patterns in the 3rd and 4th quadrants are the mirrors of 1st and 2nd quadrant watermark patterns to preserve the symmetry of the Fourier coefficients in order to obtain real-valued pixels in the spatial domain; namely |I(N-u,N-v)| is equal to |I(u,v)|. In other words, DFT coefficients with Hermitian symmetry must be modified identically by the embedded watermark.

We designate the 2-D DFT frequencies in $(-\pi, \pi]$ range in Figure 3.1. The term at the upper-left corner is the DC term. The highest frequency term is located in the center. In

order to visualize the DFT spectrum better, we draw the diamond-shaped watermark region in Figure 3.2 by moving the origin of the spectrum in Figure 3.1 to the center. It is also possible to select a region that has an annular shape as shown in Figure 3.3. This selection is more balanced and represents middle bands better; but it is computationally more expensive. Besides, there are minor performance differences between two selections. We used the diamond-shaped region in our simulations.

3.2. Embedding in the Block DCT Domain

The second watermarking technique we implemented [8], works in a selected "medium frequency" region of the 8x8 block DCT domain. We first divide the NxN image into (N/8)*(N/8) = $N^2/64$ nonoverlapping 8x8 blocks; then take DCT on each block and embed the watermark using the additive-multiplicative method explained in Chapter 2 on top of a selected set of medium coefficients.

8x8 Discrete Cosine Transform (DCT) is defined as:

$$I(u,v) = \frac{\zeta(u)}{2}.\frac{\zeta(v)}{2}\sum_{k=0}^{7}\sum_{l=0}^{7} X(k,l).\cos(\frac{(2k+1)u\pi}{16})\cos(\frac{(2l+1)v\pi}{16}) \qquad (3.1)$$

and 8x8 Inverse Discrete Cosine Transform (IDCT) is defined as:

$$X(k,l) = \sum_{u=0}^{7}\sum_{v=0}^{7} \frac{\zeta(u)}{2}.\frac{\zeta(v)}{2} I(u,v).\cos(\frac{(2k+1)u\pi}{16})\cos(\frac{(2l+1)v\pi}{16}) \qquad (3.2)$$

where k,l,u,v ε {0,1,2,3,4,5,6,7} and $\zeta(u) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{for } u > 0 \end{array} \right\}$, $\zeta(v) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{for } v = 0 \\ 1 & \text{for } v > 0 \end{array} \right\}$.

As observed from equations 3.4 and 3.5, DCT and IDCT are linear transformations and all DCT coefficients are real. Any image block can be represented as a superposition

of scaled DCT basis images scaled with DCT coefficients. These basis functions for 8x8 image blocks are illustrated in Figure 3.4.

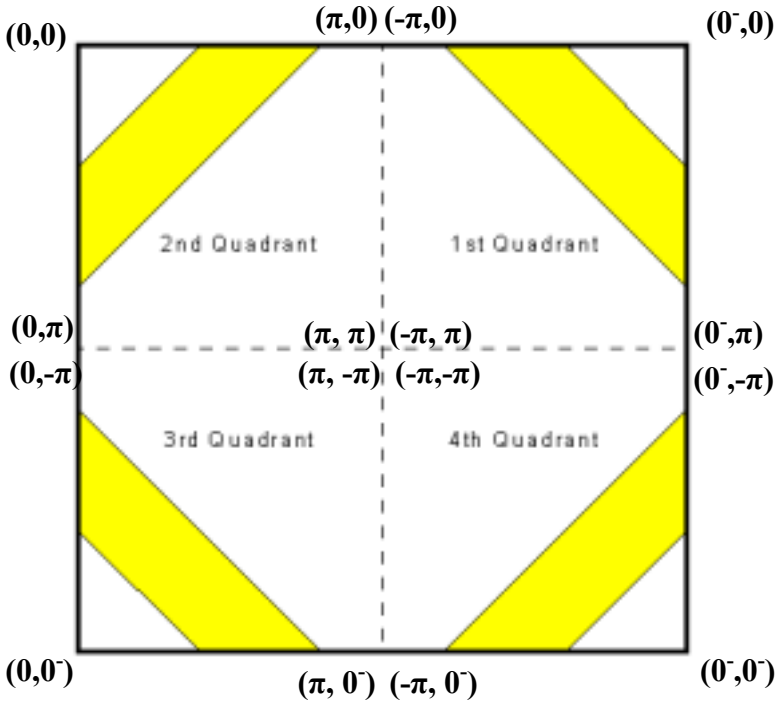

Figure 3.1. Marked DFT spectrum

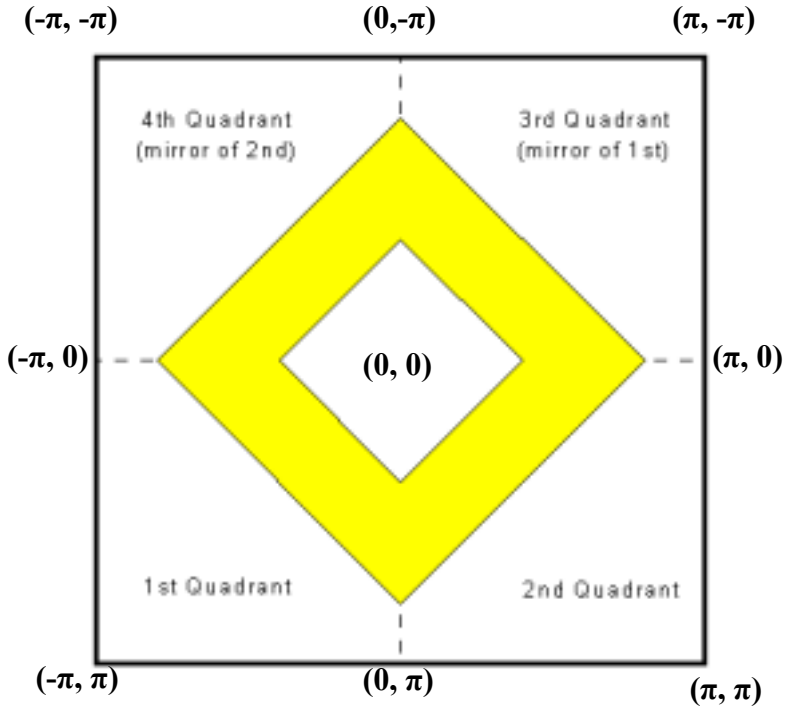

Figure 3.2. Transformed DFT spectrum

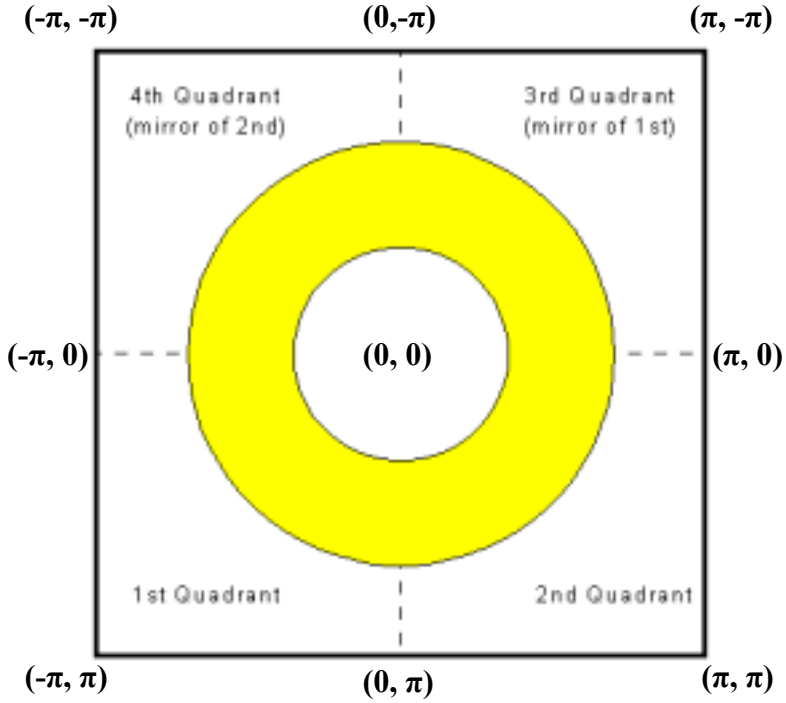

Figure 3.3. Annular transformed DFT spectrum

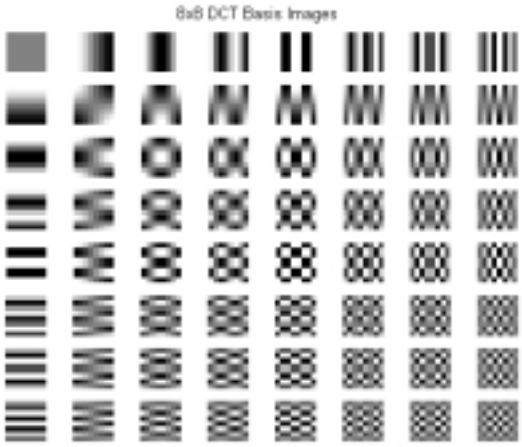

Figure 3.4. 8x8 DCT basis images

3.2.1. Selection of Block DCT Coefficients

The common practice for block DCT marking is to skip a few, say 6, coefficients in the zigzag scan pattern, and mark the following bandpass coefficients in each 8x8 DCT block. The reason for selecting these coefficients is, similar to the DFT case, the trade-off between perceptual transparency and robustness. As demonstrated in Figure 3.5, we mark 16 bandpass coefficients in each block in order to alter approximately an equal number of

transform domain coefficients as in the full frame DFT technique for comparison purpose. In this figure, the upper left coefficients correspond to low-pass frequencies. The increasing block numbers show the zigzag path followed in the block.

1	3	4	10	11	21	22	
2	5	9	12	20			
6	8	13	19				
7	14	18					
15	17						
16							

Figure 3.1. Marked region in 8x8 DCT block.

The deployment of the marked DCT coefficients over the whole image is illustrated in the 6x5 block image in Figure 3.6. The black regions are the marked DCT coefficients. In each block, 16 coefficients are watermarked, which are shown with black patterns. Any element of $\mathbf{w}_c$ can mark any of the 16 coefficients of any block by a random permutation operation.

Figure 3.2. 6x5 =30 blocks of size 8x8 in the DCT domain

3.3. Watermarking Algorithm

Watermarking methods are similar for the DFT and the block-DCT techniques with some minor differences. The flow diagram of the applied algorithm with numbered steps explained in the text is given in Figure 3.7.

3.3.1. Watermark Insertion

Step 1 is the generation of binary antipodal message data sequence **b** of length T_k.

Step 2 is the error correction encoding stage. Partition the message sequence **b** of length T_k into a number of k-length blocks and encode them with an (n,k) block code such as BCH or LDPC. The coded sequence **c** is of length T_n.

Step 3 is the repetition coding stage. Every code bit is repeated ch times. That is, repetition-coded vector $\mathbf{c_c}$ of length T_n*ch = T_c is generated.

Step 4 is spread spectrum modulation stage. Generate a ± 1 pseudo-random sequence $\mathbf{p_c}$ of length T_c using key K_1 and multiply $\mathbf{b_c}$ with $\mathbf{p_c}$ to obtain $\mathbf{w_c}$.

Step 5 is the generation of the watermark mask in the transform domain as follows:

- **For the DFT technique:** Distribute $\mathbf{w_c}$ randomly in the 1st and 2nd quadrants of watermark region shown in Figure 3.1 using key K_2. This is in fact a random permutation process. Rotate the distributed watermark sequence by 180 degrees to generate the mirror regions in the 3rd and the 4th quadrants. The regions left over, are filled with zeros, and consequently the watermark mask $\mathbf{W_m}$ of size NxN in the DFT domain is generated.
- **For the DCT technique:** Distribute $\mathbf{w_c}$ randomly to the block DCT domain watermark bands, as illustrated in Figure 3.6, using key K_2. The regions left over, are filled with zeros, and consequently the watermark mask $\mathbf{W_m}$ of size NxN in DCT domain is generated.

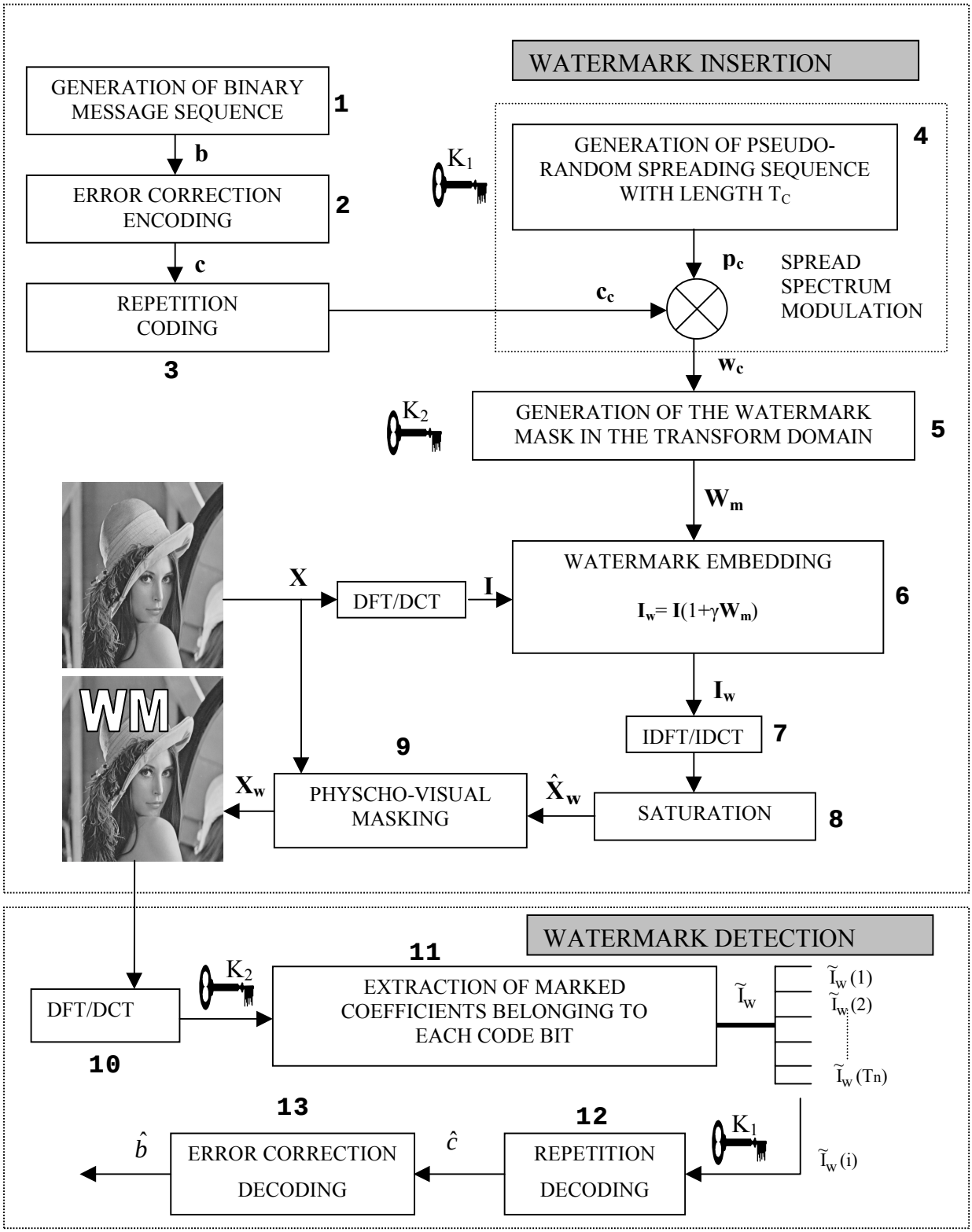

Figure 3.1. Flow diagram of the applied watermarking algorithm

Step 6 is embedding watermark in the transform domain. Embed the watermark $\mathbf{W_m}$ to **I** in the additive-multiplicative manner as:

$$\mathbf{I_w = I}(1+\gamma\mathbf{W_m}) \ , |\gamma| < 1 \qquad (3.1)$$

The γ value is chosen to tradeoff between robustness and perceptibility. The higher the γ, the more robust the watermark but at the same time the more perceptible.

Step 7 is inverse transformation to spatial domain. Take IDFT of $\mathbf{I_w}$ in the case of DFT embedding and block IDCT in the case of block DCT embedding.

Step 8 is saturation. Saturate the pixels bigger than 255 and lower than 0 to 255 and 0, respectively, to obtain $\hat{\mathbf{X}}_\mathbf{w}$.

Step 9 is applying physcho-visual masking in the spatial domain and measuring the perceptual quality. Now we have the preliminary watermarked image $\hat{\mathbf{X}}_\mathbf{w}$ in the spatial domain, we can optionally pass $\hat{\mathbf{X}}_\mathbf{w}$ through visual masking using "Human Visual System" (HVS) models, in order to adaptively embed the watermark, strongly in perceptually less significant regions and weakly in perceptually significant regions. These HVS measures of interest are contrast, frequency, luminance sensitivity, edges and textured areas [9,10,11]. Based on these measures, a NxN physcho-visual mask matrix **M** is generated, the elements of which are in the range [0,1]. The indices of **M**, where the values are higher, indicate the pixel locations where watermark can be embedded with higher strength. The marked image is generated from $\hat{\mathbf{X}}_\mathbf{w}$ as:

$$\mathbf{X_w = M}\,\hat{\mathbf{X}}_\mathbf{w} + (1 - \mathbf{M})\mathbf{X}. \qquad (3.2)$$

A straightforward first-hand perceptual quality evaluation method is to watch for watermarking artifacts. A rough measure of image fidelity is given by the peak signal to noise ratio (PSNR) and defined as:

$$\mathrm{PSNR} = 10 \log \frac{X_{peak}^2}{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[X(i,j) - X_w(i,j)\right]^2} \tag{3.3}$$

where X_{peak} is the maximum pixel luminance value in the image. PSNR values higher than 38 dB are required for acceptable image quality. Another criterion is the watermark to document ratio (WDR) and is defined as:

$$\mathrm{WDR} = 10 \log \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[X(i,j) - X_w(i,j)\right]^2}{\sum_{i=1}^{N}\sum_{j=1}^{N} X(i,j)^2}. \tag{3.4}$$

It is a measure of the ratio of watermark energy to the energy of host image. There are also more sophisticated measures such as "visual dB", a signal to noise ratio measure that takes into account the noise visibility function.

After the perceptual quality is measured, if the requirements are not satisfied γ should be reduced and watermark-embedding stage should be repeated.

3.3.2. Watermark Detection

Step 10 is taking respectively the DFT and the block DCT transform of the marked image.

Step 11 is the extraction of the marked coefficients. Extract the marked coefficients corresponding to each bit, using key K_2. For the DFT technique, the coefficients coming from 3rd and 4th quadrant are the complex conjugates of those coming from 1st and 2nd quadrants. After adding the two coefficient sets, T_n seperate **sets** of watermarked coefficient sequences, each of size ch, are obtained. These sets are the data independently handled at the bit decoding level. In other words, each bit is seperately repetition decoded.

We denote this set of ch-length vectors as $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})$ where i is between 1 and T_n. $\tilde{\mathbf{I}}_{\mathbf{w}} = [\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{1})\, \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{2}) \ldots \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{T_n})]$ where $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) = [\tilde{I}_w{}^1(i)\, \tilde{I}_w{}^2(i) \ldots \tilde{I}_w{}^{ch}(i)]$.

Step 12 is repetition decoding. Key K_1 is used to regenerate the same pseudo-random sequence $\mathbf{p_c}$ used for spectrum spreading. This $\mathbf{p_c}$ sequence, which is of length T_c, is divided into T_n consecutive ch length sequences $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ corresponding to each coded bit. Using $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})$ and $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ each coded bit is detected with a selected detector, as detailed in Section 3.4, which can be of correlator variety or maximum likelihood detector.

Step 13 is error correction decoding. The message bits are decoded from the detected code bits.

3.4. Detection Mechanisms

There are in general two types of statistical inference for the detection of random variables: "Parametric statistical inference" and "Nonparametric statistical inference".

Parametric methods require assumptions regarding the underlying distribution (e.g., Gaussian, exponential, Weibull, Gompertz distributions). That is, statistical procedures and decisions are based on distributions of the actual data values. These are computed using distributions, which are exactly specified using a finite-dimensional parameter.

“Nonparametric” methods require no or weaker distributional assumptions. Instead of fully specifying the family of distributions, we select a parameter, which exists for any distribution, and use that as the basis for inference. In most cases, it is advantegous to use nonparametric methods because the required assumptions are less restrictive than those for fully parametric models, they are easier to grasp and less costly to implement.

The performance of parametric methods as compared to non-parametric ones depends on how good the data fits the assumed underlying distribution. The data, which are of interest, are the image DFT magnitudes and the image block DCT coefficients in the DFT and the block DCT watermarking techniques, respectively. We applied two families

of parametric detectors. One contains the correlation/covariance detectors with the underlying zero-mean white Gaussian distribution and the other contains the optimum ML detectors based on Bayes tests with certain underlying distributions.

Practically, correlation is the easiest and the most straightforward detecting mechanism. It is the optimum detector when the embedding mechanism is additive and the host image features on top of which watermark is embedded follow a Gaussian pdf. This is not the case though for the discussed watermarking techniques since neither the embedding techniques are additive, nor the watermarked coefficients have Gaussian pdfs. Therefore, correlation is theoretically not the global optimum detection mechanism but the optimum linear detector. The optimum detection mechanism is the optimum detector in the Bayes sense, which takes the statistical characteristics of the transform coefficients into consideration. It is of interest, however, to compare the performances of the correlation / covariance detectors with the optimum detection method since they are much more straightforward to implement.

3.4.1. Correlation Detector

$$\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) = \tilde{\mathbf{I}}(\mathbf{i})\,(1+\gamma\,\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i})) \tag{3.1}$$

for the i^{th} bit where the watermarked coefficients vector is $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})$; the embedded sequence vector is $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ and $\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i}) = c(i)\,\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$.

The correlation detector has the form

$$\left\{\begin{array}{l} c(i) = \;\;1,\ \text{if} < |\,\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})\,|\,.\,\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) > \;\; \geq 0 \\ c(i) = -1,\ \text{if} < |\,\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})\,|\,.\,\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) > \;\; < 0 \end{array}\right\} \tag{3.2}$$

where "< . >" is the scalar product operation.

$$|\,\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})\,| = |\,\tilde{\mathbf{I}}(\mathbf{i})\,(1+\gamma\,\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i}))\,)\,| \;= |\,\tilde{\mathbf{I}}(\mathbf{i})\,|(1+\gamma\,\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i})) \text{ since } 0<\gamma<1.$$

$$|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) = |\tilde{\mathbf{I}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) + \gamma |\tilde{\mathbf{I}}(\mathbf{i})| \;(\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) \,.\, \tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i}))$$, where $\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i}) = c(i)\,\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$.

$$|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| \bullet \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) = |\tilde{\mathbf{I}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) + (\gamma |\tilde{\mathbf{I}}(\mathbf{i})| \bullet |\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})|^2)\, c(i)$$

$$|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) = |\tilde{\mathbf{I}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) + (\gamma |\tilde{\mathbf{I}}(\mathbf{i})| \,.\, \mathbf{U})\, c(i) \qquad (3.3)$$

where **U** is the unity vector.

The first term on the right of Equation 3.12 is a random variable with zero mean based on the assumption that the pseudo-random sequence $\mathbf{p_c}$ and the host image DFT, **I,** are uncorrelated. Therefore,

$$E\{|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| \,.\, \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})\} = (\gamma |\tilde{\mathbf{I}}(\mathbf{i})| \,.\, \mathbf{U})\, c(i) = \zeta_i\, c(i) \qquad (3.4)$$

where ζ_i is a positive power strength value for the coded bit c(i).

3.4.2. Covariance Detector

The correlator detector would perform better if the pseudo-random sequence $\mathbf{p_c}$ and the host image DFT, **I**, were orthogonal. Since this is not the case, a better solution is to subtract the means of $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})$ and $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ before correlation. The covariance detection rule for each code bit c(i), i =1,2,...,T_n, has the form

$$\begin{Bmatrix} c(i) = \;\; 1 \;\; , \text{if} < (|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| - \mu_{|\tilde{I}wi|}).(\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) - \mu_{\tilde{p}ci}) > \geq 0 \\ c(i) = -1 \;\; , \text{if} < (|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| - \mu_{|\tilde{I}wi|}).(\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) - \mu_{\tilde{p}ci}) > < 0 \end{Bmatrix} \qquad (3.1)$$

where $\mu_{|\tilde{I}_{wi}|}$ is the mean of $|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})|$, $\mu_{\tilde{p}_{ci}}$ is the mean of $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ and "< . >" is the scalar product operation.

To understand why covariance should perform better, we consider Equation 3.12. In this inner product expression, the interference term due to the host image itself is

$$E\{\sum_{j=1}^{ch} |\tilde{I}(i)_j| . \tilde{p}_c(i)_j\} \tag{3.2}$$

Our goal should be to minimize this. One **deterministic** solution might be to choose the marked image coefficients and the pseudo-random sequence before watermark insertion in such a way that the above cumulative-sum is minimized as much as possible. If this is too constraining, then one can at least ensure that

$$\sum_{j=1}^{ch} \tilde{p}_c(i)_j = 0. \tag{3.3}$$

This is image-independent, hence **stochastic**. This will result in an equal number of positive and negative correlation-output noise terms. Assuming that the image coefficients corresponding to positive and negative pseudo-terms are distributed similarly, the expected sum given in Equation 3.15 will not be high.

The deterministic solution is obviously not practical. The stochastic solution is in fact based on filtering out the interference or enhancing the correlation structure. Since most of the image components are in the low-pass band, high pass filtering the watermarked image will tend to suppress the interference term. In fact, subtracting the mean is a simplistic high-pass filtering from $\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})$. A more principled way of interference suppression would be to implement a Wiener filter. One can thus use the Wiener estimate of the watermark in the image and obtain the inner product with $\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i})$ as

$$< \text{Wiener}\{\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})\}.\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) >$$

3.4.3. Normalized Correlation Variants

There are two other possible detectors, which are variants of correlation and covariance detectors: One is the normalized correlation detector and the other is the correlation coefficient detector.

Normalized correlation detector likelihood ratio is defined as:

$$\Lambda = \frac{< | \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | . \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) >}{\sqrt{< | \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | . | \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | > . < \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) . \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) >}}$$

Correlation coefficient detector likelihood ratio is defined as:

$$\Lambda = \frac{< (| \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | - \mu_{|\tilde{I}wi|}) . (\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) - \mu_{\tilde{p}ci}) >}{\sqrt{< (| \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | - \mu_{|\tilde{I}wi|}) . (| \tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i}) | - \mu_{|\tilde{I}wi|}) > . < (\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) - \mu_{\tilde{p}ci}) . (\tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) - \mu_{\tilde{p}ci}) >}}$$

Decision rule for both detectors is as:

- Decide for code bit 1, if $\Lambda > \eta$
- Decide for code bit 0, if $\Lambda < -\eta$

In our case, the threshold η is again 0. Therefore, normalized correlation detector gives exactly the same result as correlation detector. Similarly, the correlation coefficient detector gives the same result as covariance detector. In fact, if we know that watermark exists in the document, then using normalized detectors has no advantage. We can use either the normalized or non-normalized detector types for **reading** the watermark. But, in some applications, there may be both marked and unmarked media and it is also necessary to test for the existence of the watermark. For these kinds of applications, normalized inner product should be preferred. For example, one can apply

- Decide for erasure if $-\eta < \Lambda < \eta$
- Decide for code bit 1 if $\Lambda > \eta$

- Decide for code bit 0 if $\Lambda < -\eta$

3.4.4. Optimum Detector for the DFT Technique

ML approach necessitates finding a distribution model, which well represents the DFT amplitudes. One possible choice is Weibull distribution defined in [12] as

$$f_x(x) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^{\beta}\right], (x > 0) \qquad (3.1)$$

The two specific cases $\beta=1$ and $\beta=2$ correspond to the well-known exponential and Rayleigh distributions, respectively. The distribution is defined only for positive axis since the amplitudes of DFT coefficients are all positive.

Estimation of α and β parameters is done on 16 non-overlapping radial DFT bands given in Figure 3.8. Hence, we estimate 16 different (α, β) sets. We distribute the DFT coefficients in these bands considering only the radial correlation of spectral components. We build the sets in this way with the belief that the coefficients that have similar radial frequencies have similar distributions. In fact, spectral components depend both on the radius and angle. An example representation utilizing both radial and angular correlation of spectral components is given in Figure 3.9 [12]. Though includes the angular correlation, this representation is far from being accurate in terms of radial correlation among the elements in any set.

Since we do not have access to the original image coefficients, we do the estimation on the marked image coefficients. We assume that, the presence of the watermark does not change the parameters significantly, provided that γ is sufficiently small. The validity of the assumption will be demonstrated by experiments.

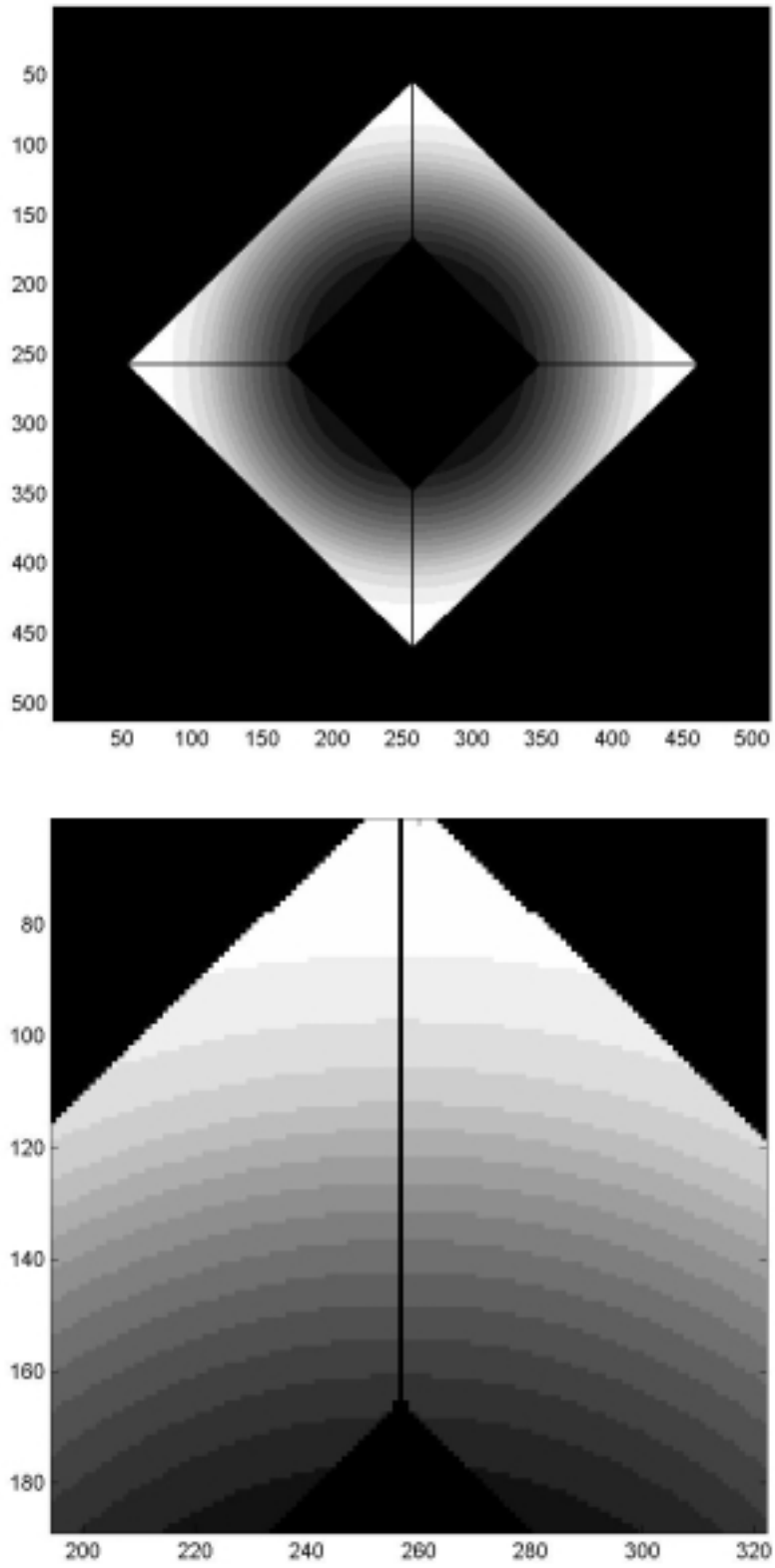

Figure 3.1. 16 radial bands in DFT for the estimation of the (α, β) parameters

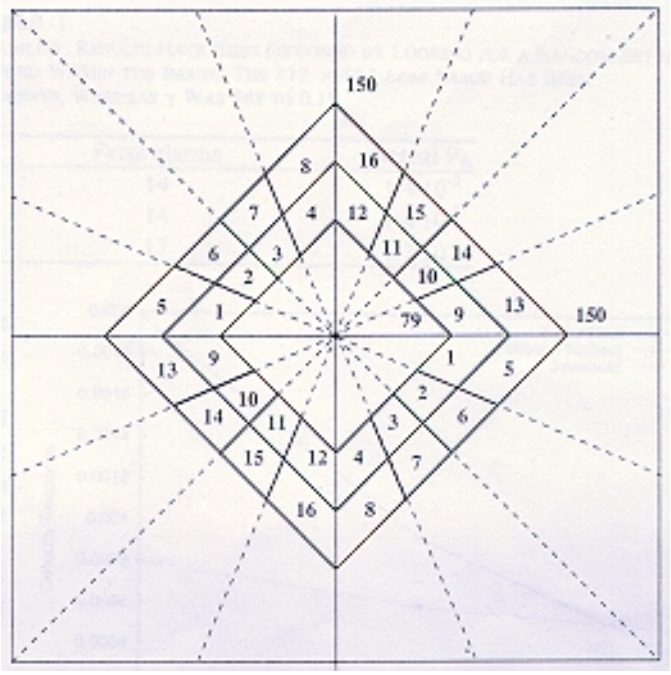

Figure 3.2. Alternative DFT region partitioning for estimation of the (α, β) parameters

When we calculate the mean and variance of a sequence with underlying Weibull distribution, we get

$$\mu_x = \alpha\Gamma\left(1+\frac{1}{\beta}\right) \tag{3.2}$$

$$\sigma_x^2 = \alpha^2\{\Gamma\left(1+\frac{2}{\beta}\right) - \left[\Gamma\left(1+\frac{1}{\beta}\right)\right]^2\} = \alpha^2\Gamma\left(1+\frac{2}{\beta}\right) - \mu_x^2 \tag{3.3}$$

where Γ is the Gamma function.

(α, β) parameter set can be found from the mean and variance of the sequence. As a solution strategy for (α, β), we use the ML estimates for these parameters given in [13] as

$$\tilde{\alpha} = \left(\frac{1}{M/8}\sum_{i=1}^{M/8} x_i^{\tilde{\beta}}\right)^{\frac{1}{\tilde{\beta}}} \tag{3.4}$$

and

$$\tilde{\beta} = \left[\left(\sum_{i=1}^{M/8} x_i^{\tilde{\beta}} \log x_i \right) \left(\sum_{i=1}^{M/8} x_i^{\tilde{\beta}} \right)^{-1} - \frac{1}{M/8} \sum_{i=1}^{M/8} \log x_i \right]^{-1} \qquad (3.5)$$

First, we need to find β from equation 3.21. This equation is a x = f(x) type equation and necessitates a "fixed point" solution. For its solution, MATLAB Symbolic Toolbox can be used. After β is found, it is put in equation 3.20 and α is found.

Once the model parameters (α, β) have been found, decision is done separately on "footprint" of each code bit. Take the i^{th} bit, embedded as $|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| = |\tilde{\mathbf{I}}(\mathbf{i})|\,(1+\gamma\,\tilde{\mathbf{w}}_{\mathbf{c}}(\mathbf{i}))$. For the sake of easier formulation, define $\mathbf{y} = \left|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})\right|$, such that y_j represents the j^{th} modulated DFT amplitude in this sequence and m_j is the j^{th} pseudorandom value in the sequence
Then, (3.6)

$$f_y(\mathbf{y}) = \prod_{j=1}^{ch} \frac{\tilde{\beta}_j}{\tilde{\alpha}_j(1+\gamma\, w_c(i)_j)} \left(\frac{y_j}{\tilde{\alpha}_j(1+\gamma\, w_c(i)_j)}\right)^{\tilde{\beta}_j - 1} \exp\left[-\left(\frac{y_j}{\tilde{\alpha}_j(1+\gamma\, w_c(i)_j)}\right)^{\tilde{\beta}_j} \right], (y > 0)$$

where $(\tilde{\alpha}_j, \tilde{\beta}_j)$ is one of the 16 parameter sets depending on which set y_j belongs to, and where we have used the independence assumption of the set $\{y_j\}$.

Since the a-priori probabilities and false detection costs of 1 and –1 are equal, Bayes test becomes:

$$\Lambda(y) = \frac{f(y\,|\,1)}{f(y\,|\,-1)} \underset{-1}{\overset{1}{\gtrless}} 1 \qquad (3.7)$$

Putting Equation 3.22 in the Λ(y) and omitting the dependence upon the bit index i, we obtain:

$$\Lambda(\mathbf{y}) = \frac{\prod_{j=1}^{ch} \frac{\beta_j}{\alpha_j(1+\gamma m_j)} \left(\frac{y_j}{\alpha_j(1+\gamma m_j)}\right)^{\beta_j - 1} \exp\left[-\left(\frac{y_j}{\alpha_j(1+\gamma m_j)}\right)^{\beta_j}\right]}{\prod_{j=1}^{ch} \frac{\beta_j}{\alpha_j(1-\gamma m_j)} \left(\frac{y_j}{\alpha_j(1-\gamma m_j)}\right)^{\beta_j - 1} \exp\left[-\left(\frac{y_j}{\alpha_j(1-\gamma m_j)}\right)^{\beta_j}\right]} \underset{-1}{\overset{1}{\gtrless}} 1 \qquad (3.8)$$

In order to make the computation easier, we take the natural logarithm of the likelihood ratio and obtain the log-likelihood ratio sufficient statistics as

(3.9)

$$S(\mathbf{y}) = \left(\sum_{j=1}^{ch} \beta_j \ln(1-\gamma m_j) + \sum_{j=1}^{ch} \left(\frac{y_j}{\alpha_j(1-\gamma m_j)}\right)^{\beta_j}\right) - \left(\sum_{j=1}^{ch} \beta_j \ln(1+\gamma m_j) + \sum_{j=1}^{ch} \left(\frac{y_j}{\alpha_j(1+\gamma m_j)}\right)^{\beta_j}\right) .$$

$S(\mathbf{y}) \underset{-1}{\overset{1}{\gtrless}} 0$. In other words, decide for code bit = 1 if,

$$\left(\sum_{j=1}^{ch} \beta_j \ln(1-\gamma m_j) + \sum_{j=1}^{ch} \left(\frac{y_j}{\alpha_j(1-\gamma m_j)}\right)^{\beta_j}\right) > \left(\sum_{j=1}^{ch} \beta_j \ln(1+\gamma m_j) + \sum_{j=1}^{ch} \left(\frac{y_j}{\alpha_j(1+\gamma m_j)}\right)^{\beta_j}\right) \qquad (3.10)$$

3.4.5. Optimum Detector for the Block-DCT Technique

One possible good choice to model the distribution of the DCT coefficients is the zero-mean generalized Gaussian distribution defined in [8] as

$$f_x(x) = Ae^{-|\alpha x|^{\beta}} \qquad (3.1)$$

The two specific cases β=1 and β=2 correspond to the well-known Laplacian and Gaussian distributions, respectively. A and α are functions of β and the standard deviation σ as

$$\alpha = \frac{1}{\sigma}\left(\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\right)^{1/2} \text{ and } A = \frac{\alpha\beta/2}{\Gamma(1/\beta)} \qquad (3.2)$$

Thus, the two parameters β and σ are sufficient to specify the distribution. That is, we need to estimate these two parameters to accurately construct the ML detector.

Estimation of β and σ parameters is done on each of the marked 16 coefficients in the 8x8 blocks, separately. That is, there are $N_b = N^2/64$ DCT samples, the number of 8x8 blocks for NxN image, for each block coefficient and 16 different (β, σ) values are obtained after estimation. From (β, σ), one can calculate α and A. In fact, there is no need to calculate A because it is irrelevant to the likelihood ratio.

The variance can be found as

$$\sigma^2_{i,j} = \frac{1}{Nb(1+\gamma^2)} \sum_{k=1}^{Nb} I^2_{i,j}(k) - \left(\frac{1}{Nb} \sum_{k=1}^{Nb} I_{i,j}(k) \right)^2 \qquad (3.3)$$

where $\sigma_{i,j}$ is the standard deviation corresponding to the ensemble of DCT coefficients in the i^{th} row and the j^{th} column of the 8x8 block DCT. k stands for the 2-D block index corresponding to one of N_b blocks.

Estimates of $\beta_{i,j}$ can be obtained by matching the sample mean absolute value and the sample variance of the DCT coefficients to those of the generalized Gaussian distribution as proposed in [14] for solving the ML equation

$$\frac{E(|I_{i,j}|)}{\sigma_{i,j}} = \frac{\Gamma(2/\beta_{i,j})}{\sqrt{\Gamma(1/\beta_{i,j})\Gamma(3/\beta_{i,j})}}. \qquad (3.4)$$

An open solution for $\beta_{i,j}$ does not exist. Instead, one can sample the solution space by computing the right-hand side of Equation 3.32 for several values of β and find the best match. We constructed an ensemble between $0.3 < \beta < 2$ in steps of 0.01 to estimate the best approximate $\beta_{i,j}$ parameters.

Once ($\alpha_{i,j}$, $\beta_{i,j}$) parameters are estimated for each of the 16 coefficient sets, we can start decoding each bit independently. Say, **y** is the DCT coefficient sequence for a specific bit, y_j representing the j^{th} DCT coefficient and m_j representing the j^{th} pseudorandom value in this sequence.

Again under independence assumption, the pdf of the DCT coefficient sequence for any one bit is the product of the pdfs of the DCT coefficients in the chip sequence. Putting this into likelihood ratio, we obtain the Bayes test as

$$\Lambda(\mathbf{y}) = \frac{\prod_{j=1}^{ch} \frac{1}{1+\gamma mj} \exp(-|\frac{\alpha jyj}{1+\gamma mj}|^{\beta j})}{\prod_{j=1}^{ch} \frac{1}{1-\gamma mj} \exp(-|\frac{\alpha jyj}{1-\gamma mj}|^{\beta j})} \qquad (3.5)$$

where (β_j, α_j) is one of the 16 parameter sets depending on which set y_j belongs to.

In order to make the computation easier, we take the natural logarithm of the likelihood ratio and obtain the log-likelihood ratio sufficient statistics as

$$S(\mathbf{y}) = (\sum_{j=1}^{ch} \ln(1-\gamma mj) + \sum_{j=1}^{ch} |\frac{\alpha jyj}{1-\gamma mj}|^{\beta j}) - (\sum_{j=1}^{ch} \ln(1+\gamma mj) + \sum_{j=1}^{ch} |\frac{\alpha_j yj}{1+\gamma mj}|^{\beta j}) \qquad (3.6)$$

$S(\mathbf{y}) \gtrless_{-1}^{1} 0$. In other words, decide for code bit = 1 if,

$$(\sum_{j=1}^{ch} \ln(1-\gamma mj) + \sum_{j=1}^{ch} |\frac{\alpha jyj}{1-\gamma mj}|^{\beta j}) > (\sum_{j=1}^{ch} \ln(1+\gamma mj) + \sum_{j=1}^{ch} |\frac{\alpha_j yj}{1+\gamma mj}|^{\beta j}) \qquad (3.7)$$

We proposed three alternative detectors for the watermark detector: correlation detector, covariance detector and ML detector. Though ML detector should give the highest performance, we are interested in finding out whether suboptimal but much simpler detectors perform sufficiently close. We will test this with simulations in Chapter5.

4. CODING STRATEGIES FOR WATERMARKING

4.1. Introduction

In any communication system, "error coding" the information bit sequence may help to achieve more efficient and more reliable transmission. Since watermarking is also a communications system, we consider using two families of error correction codes, BCH codes and LDPC codes, to increase the watermark message carrying capacity of images. In the following discussion, we elaborate coding, specifically BCH and LDPC coding.

Error coding is adding redundancy to the payload bit stream in such a way that there is memory, influence of some or all of the bits on the others' value, in the expanded sequence. In a communication system with error correction facility, two new blocks are introduced. One is the encoder block at the transmitter side. This block takes the message sequence as input, adds controlled redundancy to it and sends out a longer coded sequence. The second one is the decoder block at the receiver side. The decoder uses the redundancy introduced by the encoder to detect and correct errors that occurred during transmission. Figure 4.1 shows the block diagram of a communication system with error correction blocks.

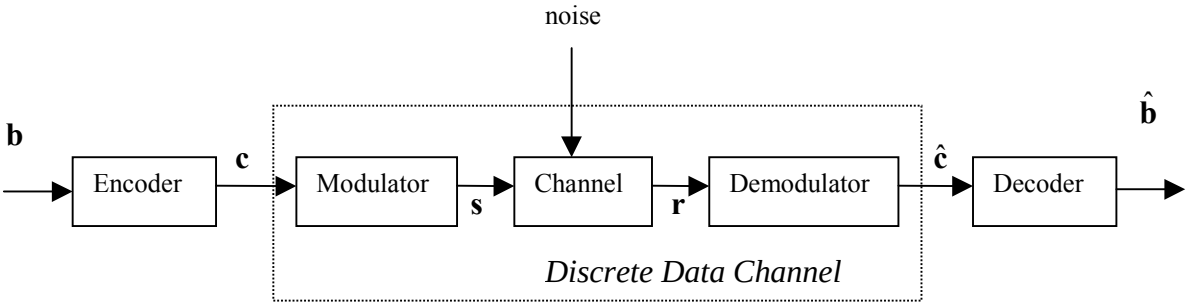

Figure 4.1. Communication system with forward error correction facility

Since watermarking systems are also communication systems, we will base our discussion on watermarking as applied to the model in Figure 4.1. The binary sequence **b** is the watermark message. It passes through encoder at the transmission side, the output of

which is the coded binary **c** sequence. The **c** vector is mapped via the chosen watermarking technique to the modulated signal **s,** which is transmitted through the watermarking channel. In this thesis work, this channel is the image itself with all its concomitant distortions. The watermarked image may suffer from various intentional and unintentional attacks. The extracted noisy watermark **analog** sequence **r** is fed to the demodulator block, which tries to recover the original code sequence from the possibly distorted channel output sequence **r**. The demodulator outputs hard or soft decisions $\hat{\mathbf{c}}$ for the received code bits. For binary signaling, a hard decision device decides for 1 or 0, so that the value of the received code **bit** is quantized to binary values before sent to the decoder. A soft decision device, in contrast, gives a multivalued indication, which can be interpreted as a measure of how close the received signal is to the 0 or 1 decision. In other words, it is a **confidence measure**. This is realized by quantizing the demodulated signal range to more than two, typically 8, levels. These soft decision outputs are used by the ML decoder for deciding for the values of the transmitted **bit sequence**.

In this thesis, we applied BCH coding with hard decision decoding and LDPC coding with soft decision decoding. The applied watermark embedding techniques are additive-multiplicative and the DFT and the block-DCT amplitudes are not Gaussian. Therefore, our watermarking channels are not AWGN channels. Nevertheless, in the following text, we discuss the advantage of soft decision decoding on AWGN channels. Though this does not exactly apply to our watermarking channels it gives an idea. An example of hard decision and soft decision with 3-bit quantizer is shown in Figure 4.2. In this figure, f(r|1) and f(r|0) are the Gaussianly distributed conditional probability density functions.

The verbal interpretation of the quantized soft decision outputs is given in Table 4.1. Soft decision is known to bring about 3 dB coding gain with respect to hard decision in AWGN channels. The more the number of quantization levels is, the more the coding gain is. However, not much is gained by increasing the number of levels any further. In fact, theoretically, increasing the number of levels up to infinity brings about only 0.5 dB gain with respect to 3 bit quantization. Hence, any complexity, beyond 8-level quantization, is not warranted.

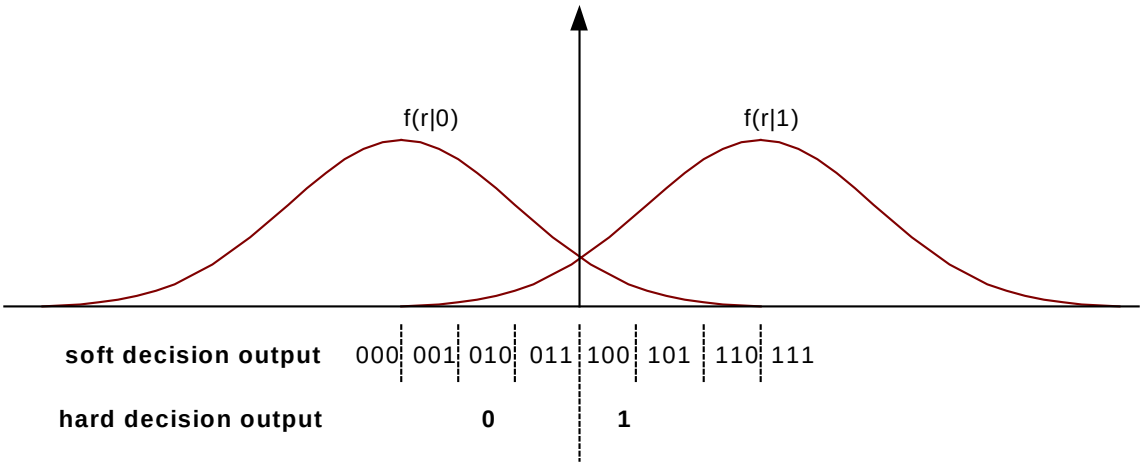

Figure 4.2. Hard and soft decision demodulation of a BPSK signal

Table 4.1. Interpretation of soft decision outputs in Figure 4.2

Soft Decision Value	**Confidence Measure**
000	VERY STRONG 0
001	STRONG 0
010	WEAK 0
011	VERY WEAK 0
100	VERY WEAK 1
101	WEAK 1
110	STRONG 1
111	VERY STRONG 1

Let's explain, why soft decoding achieves coding gain in AWGN channel. Let us assume that infinite number of quantization levels exists and on-off signaling is used. For an (n, k) block code, there are 2^k valid codewords $\mathbf{c_i}$. When hard decision decoding is used, there are 2^n decision elements in decision sample space. In contrast, there are infinitely many decision elements in decision sample space when soft decision is used. ML decoding rule common to hard and soft decoding is given in Equation 4.1.

$$\max_i \{P(\underline{r} \mid \underline{c}_i)\} = \max_i \{\prod_{j=1}^{n} [\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(r_j - c_{ij})^2 / 2\sigma^2]\} \qquad (4.1)$$

For hard decision decoding, since $r_j \in \{0,1\}$, then $|r_j - c_{ij}|$ and $(r_j - c_{ij})^2$ are also $\in \{0,1\}$. Then, after some manipulations on Equation 4.1, hard decision ML decoding rule becomes

$$\min_i \sum_{j=1}^{n} | r_j - c_{ij} | \qquad (4.2)$$

This is nothing but minimizing the **Hamming distance** of the received hard decision vector on codeword ensemble. For soft decision decoding, ML decision rule becomes

$$\min_i \sum_{j=1}^{n} (r_j - c_{ij})^2 \qquad (4.3)$$

This is nothing but minimizing the **Euclidean distance** of the received soft decision vector to the codeword ensemble. Euclidean distance is the exact and accurate measure for ML sequence decoding, whereas Hamming distance is an approximation. Therefore, ML decoding using Euclidean distance measure as a consequence of soft decision, performs better than ML decoding using Hamming distance measure as a consequence of hard decision.

4.2. Coding Alternatives on Watermarking Channels

In Figure 4.1, we showed the general block diagram of a watermarking system as a communication system. The scheme is typically based on concatenation coding. Typically, the outer code is a block code like BCH, while the inner code is a repetition code. The repetition coder-decoder can also be interpreted as a diversity transmission-reception. We repeat each single code bit "ch" times and multiply this $\mathbf{c_c}$ sequence with the binary spreading vector $\mathbf{p_c}$. Thus, the encoder block can be expanded as given in Figure 4.3.

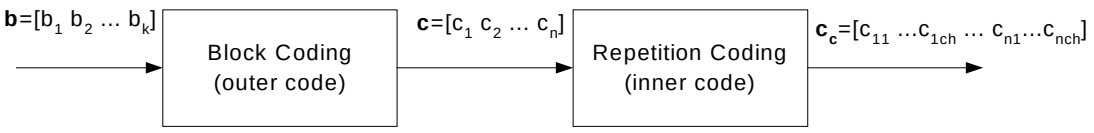

Figure 4.1. Block diagram of concatenated coding

Concatenation can also be done in the reverse order: Block coding used as inner code and repetition coding as outer code. However, this latter proposal is not appropriate for watermarking channels that operate at very high probability of error. In this case, the inner (n,k,t) block code which can correct up to t errors in one block, can not cope with the high probability error rate, and in fact may deteriorate the reception. On the other hand, if the error rate per block were never to exceed t errors, then inner block code would correct all of them and there would be no need for outer repetition code.

Repetition coding outperforms block codes at high channel error rate conditions (more than 10%), whereas block coding works much better than repetition coding at lower channel error rate conditions [15,16]. Hence, the rationale of the proposal is that, the inner repetition code improves the channel conditions so sufficiently that the outer block code works on this improved sequence to make it even better.

We want to investigate the advantage of "Soft decision decoding" over "Hard decision decoding", and the various combinations of the "Inner repetition-outer block coding" scheme. Therefore, we will apply

- Repetition codes only with hard decison decoding,
- Repetition + BCH codes with hard decison decoding
- Repetition + LDPC codes with soft decision decoding

and compare their performances.

We select BCH block codes; because at block lengths of a few hundred, they outperform all other block codes with the same block length and code rate [17]. We select LDPC codes; because they are believed to be the best error correction codes performing very near to the Shannon limits [5,18].

In the rest of this chapter, we will give basic notions about BCH codes and LDPC codes.

4.3. Cyclic Codes

Bose-Chaudri-Hocquenghmem (BCH) codes are a special class of cyclic linear block codes. An (n,k) linear block code is called a cyclic code if a cyclicly shifted n-tuple codeword is also a codeword. That is, if $\mathbf{U}=(u_0,u_1,...,u_{n-1})$ is a codeword, then $\mathbf{U^{(i)}} = (u_{n-i},u_{n-i+1},...,u_{n-1}, u_0,u_1,...,u_{n-i-1})$ obtained by *i* right-shift operation should also be a codeword.

The codeword **U** can be expressed in a polynomial form as

$$\mathbf{U(x)} = u_0 + u_1x + u_2x^2 + ... + u_{n-1}x^{n-1}. \qquad (4.1)$$

From this, we will show with an example that, $\mathbf{U^{(i)}(x)} = x^i\,\mathbf{U(x)}$ *mod* (x^n+1).

Say i =1. Then,

$$\mathbf{U(x)} = u_0 + u_1x + u_2x^2 + ... + u_{n-1}x^{n-1}$$

$$x\,\mathbf{U(x)} = u_0x + u_1x^2 + u_2x^3 + ... + u_{n-1}x^n. \qquad (4.2)$$

Adding u_{n-1} to the right hand side of equation 4.5 twice, which is equivalent to adding and subtracting it once in modulo-2 arithmetic, we get:

$$x\,\mathbf{U(x)} = u_{n-1} + u_0x + u_1x^2 + u_2x^3 + ... + u_{n-2}x^{n-1} + u_{n-1}x^n + u_{n-1}$$

$$= \mathbf{U^{(1)}(x)} + u_{n-1}x^n + u_{n-1} = \mathbf{U^{(1)}(x)} + u_{n-1}(x^n + 1)$$

$$\Rightarrow \mathbf{U^{(1)}(x)} = x\mathbf{U(x)} \bmod (x^n+1).$$

One can generate a cyclic code using a generator polynomial as

$\mathbf{g(x)} = g_0 + g_1x + g_2x^2 + ... + g_{n-k}x^{n-k}$, which is unique for an (n,k) cyclic code, where g_0 and g_{n-k} are 1. It is also necessary that **g(x)** is a factor of $x^n + 1$. That is $(x^n + 1) = \mathbf{g(x)h(x)}$. Here **h(x)** can also be a generator matrix. As an example for n=7,
$(x^7+1) = (1+x+x^3)\ (1+x+x^2+x^4)$. If we use $\mathbf{g(x)} = (1+x+x^3)$ as the generator polynomial, then it is (7,4) cyclic code. If we use $\mathbf{g(x)} = (1+x+x^2+x^4)$, then it is (7,3) cyclic code. If the message polynomial **m(x)** is written as $\mathbf{m(x)} = m_0 + m_1x + m_2x^2 + ... + m_{k-1}x^{k-1}$, then, the codeword, **U(x)**, is generated as **U(x) = m(x) g(x)** *mod2*.

In some cases, systematic coding may be desirable. A systematic (n,k) linear block code maps k-tuple message into n-tuple code in such a way that part of the code coincides with the message. The remaining (n-k) bits constitute the parity check sequence. To generate a systematic code, we first multiply the message sequence with x^{n-k} and obtain $x^{n-k}\ \mathbf{m(x)} = m_0\ x^{n-k} + m_1\ x^{n-k+1} + m_2\ x^{n-k+2} + ... + m_{k-1}x^{n-1}$. If we divide this equation by **g(x)**, then we get:

$$x^{n-k}\ \mathbf{m(x)} = \mathbf{q(x)\ g(x)} + \mathbf{p(x)}, \qquad (4.3)$$

where the remainder $\mathbf{p(x)} = p_0 + p_1x + p_2x^2 + ... + p_{n-k-1}x^{n-k-1}$. That is, $\mathbf{p(x)} = x^{n-k}\ \mathbf{m(x)}$ *mod g(x)*. Adding **p(x)** to both sides of Equation 4.6 in modulo-2 arithmetic, we obtain $\mathbf{p(x)} + x^{n-k}\ \mathbf{m(x)} = \mathbf{q(x)\ g(x)} = \mathbf{U(x)}$. This is a valid codeword polynomial, since it is a polynomial of degree less than or equal to (n-1) and divisible by **g(x)**. Then, the code sequence becomes $\mathbf{U} = [\ p_0\ p_1\ p_2\ ...\ p_{n-k-1}\ m_0\ m_1\ m_2\ ...\ m_{k-1}\ ]$.

4.4. BCH Codes

BCH codes are binary cyclic codes, which have the capability of correcting multiple errors. They provide a large selection of block lengths, code rates and error correction capability. Table 4.2 shows the block length (n), message length (k) and error correction capability (t) of some block codes.

BCH codes are important because at block lengths of a few hundreds, they outperform all other block codes with the same block length and code rate [17]. For a

given code rate, the decoded error probability is known to improve with increasing block length n [17]. As block length increases, the performance curves show an all-or-nothing behavior. This can be attributed to the law of large numbers. A binary block code can correct t out of n code bits. As n becomes large, it is with high probability that the number of errors will almost always be higher or lower than t [17]. Therefore for BERs lower than $\sim 10^{-2}$ a high block length is advantageous.

Table 4.1. Some BCH codes

n	k	*t*	**n**	k	*t*	**n**	k	*t*	**n**	k	*t*	**n**	k	*t*	**n**	k	*t*
7	4	1	127	113	2	255	207	6	255	47	42	511	376	15	511	193	43
15	11	1	127	106	3	255	199	7	255	45	43	511	367	16	511	184	45
15	7	2	127	99	4	255	191	8	255	37	45	511	358	18	511	175	46
15	5	3	127	92	5	255	187	9	255	29	47	511	349	19	511	166	47
31	26	1	127	85	6	255	179	10	255	21	55	511	340	20	511	157	51
31	21	2	127	78	7	255	171	11	255	13	59	511	331	21	511	148	53
31	16	3	127	71	9	255	163	12	255	9	63	511	322	22	511	139	54
31	11	5	127	64	10	255	155	13	511	502	1	511	313	23	511	130	55
31	6	7	127	57	11	255	147	14	511	493	2	511	304	25	511	121	58
63	57	1	127	50	13	255	139	15	511	484	3	511	295	26	511	112	59
63	51	2	127	43	14	255	131	18	511	475	4	511	286	27	511	103	61
63	45	3	127	36	15	255	123	19	511	466	5	511	277	28	511	94	62
63	39	4	127	29	21	255	115	21	511	457	6	511	268	29	511	85	63
63	36	5	127	22	23	255	107	22	511	448	7	511	259	30	511	76	85
63	30	6	127	15	27	255	99	23	511	439	8	511	250	31	511	67	87
63	24	7	127	8	31	255	91	25	511	430	9	511	241	36	511	58	91
63	18	10	255	247	1	255	87	26	511	421	10	511	238	37	511	49	93
63	16	11	255	239	2	255	79	27	511	412	11	511	229	38	511	40	95
63	10	13	255	231	3	255	71	29	511	403	12	511	220	39	511	31	109
63	7	15	255	223	4	255	63	30	511	394	13	511	211	41	511	28	111
127	120	1	255	215	5	255	55	31	511	385	14	511	202	42	511	19	119

It is also possible to analytically calculate the BER performances of different BCH codes. The BER of an (n,k,t) code is approximated as

$$P_b \cong \frac{1}{n} \sum_{m=t+1}^{n} m \binom{n}{m} p^m (1-p)^{n-m} \qquad (4.1)$$

where p is the channel error probability. Say the channel error probability p= 0.02. Then the theoretical BER results are $P_b(15,7,2)$ = 6.2064e-004, $P_b(31,16,3)$ = 4.3436e-004, $P_b(63,30,6)$ = 3.0190e-005, $P_b(127,64,10)$ = 4.7960e-006. Hence, for p= 0.02, there is theoretically two orders of magnitude improvement as we switch from (15,7,2) to (127,64,10).

4.5. Linear Parity Check Codes

LDPC coding is a special class of linear parity check coding. A linear (d,k) parity check code is represented by a kxd binary generator matrix **G** such that $\mathbf{c} = \mathbf{G}^T\mathbf{b}$ *mod 2*, where **b** is the payload data sequence and **c** is the codeword. Though not strictly necessary, we will use generator matrices in their systematic form as $\mathbf{G}^T = \begin{bmatrix} \mathbf{I_K} \\ \mathbf{P} \end{bmatrix}$ where $\mathbf{I_K}$ is the k-sized identity matrix and **P** is the mxk binary parity matrix. m is equal to (d-k). We can use this matrix to define another matrix as $\mathbf{H} = \begin{bmatrix} -\mathbf{P} & \mathbf{I_M} \end{bmatrix}$ which is the parity check matrix in systematic form. It is also possible to go in the reverse direction: First define **H** in systematic form, then find **G**. One can easily show that $\mathbf{HG}^T = \mathbf{0}$. $\hat{\mathbf{c}} = \mathbf{c} + \mathbf{n} = \mathbf{G}^T\mathbf{b} + \mathbf{n}$ *mod 2* where **n** is the additive noise. Then, $\mathbf{H}\hat{\mathbf{c}}$ *mod 2* $= \mathbf{H}(\mathbf{G}^T\mathbf{b} + \mathbf{n})$ *mod 2* $= \mathbf{0} + \mathbf{Hn}$ *mod 2* $= \mathbf{z}$ where the syndrome vector $\mathbf{z} = \mathbf{H}\hat{\mathbf{c}}$ *mod 2*.

Our goal is to find **c** sequence with length d. If we can find it, we can also find the message sequence **b**, since **b** is the first k bits of codeword **c**. How can we find **c**? By subtracting **n** from $\hat{\mathbf{c}}$. Since subtraction is equivalent to addition in modulo-2 arithmetic, it is also possible to find **c** by adding **n** to $\hat{\mathbf{c}}$. Then, the problem is to find **n**. To find **n**, we need to solve the equation $\mathbf{z} = \mathbf{H}\hat{\mathbf{c}}$ *mod 2* which is equivalent to

$$\mathbf{z} = \mathbf{Hn} \text{ mod } 2 \qquad (4.1)$$

H and $\hat{\mathbf{c}}$ are known. One can also easily calculate **z** from these. Then, the sole problem is to find **n** from **z** and **H** in Equation 4.8. Though looks easy, the solution of finding **n** is very difficult, since it contains d unknowns and (d-k) = m equations.

4.6. LDPC Codes

Low Density Parity Check (LDPC) codes are linear parity check codes with parity check matrices **H** that have a small number of ones and all the rest zeros. This property decreases the complexity of the practical decoder as will be explained later. A (d,p,y) LDPC code's parity check matrix **H** has p ones in each column and y ones in each row, where d is the row size. Therefore, the column length is (dp/y). The code rate is (y-p)/y. An example LDPC matrix **H** is shown in Table 4.3.

Introduced by Gallager in 1962 [18] and therefore also known as Gallager codes, LDPC codes were overlooked due to their high computational complexity. Mackay re-introduced these codes in his work "Good Error-Correcting Codes based on Very Sparse Matrices"[5]. He showed how to implement a practical decoder for LDPC codes. We used that decoder in our work.

Table 4.1. (20,3,4) parity check matrix

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

4.7. Practical Decoding of the LDPC Codes

In this section, we briefly explain the practical decoding of LDPC codes. As explained before, our goal is to find **n** in

$$\mathbf{z} = \mathbf{Hn} \ mod \ 2$$

Redefining the problem, we need to find the **n** noise vector with length d from the received syndrome vector **z** with length m using mxd parity check matrix **H**.

We build a bipartite graph having m check nodes and d bit nodes as in Figure 4.4. **n** sequence represents the binary noise sequence and **z** sequence represents the syndrome sequence. Bidirectional rows indicate the locations of the ones in the parity check matrix. Say there is a bidirectional row between n_i and z_j. Then **H**(j,i)=1.

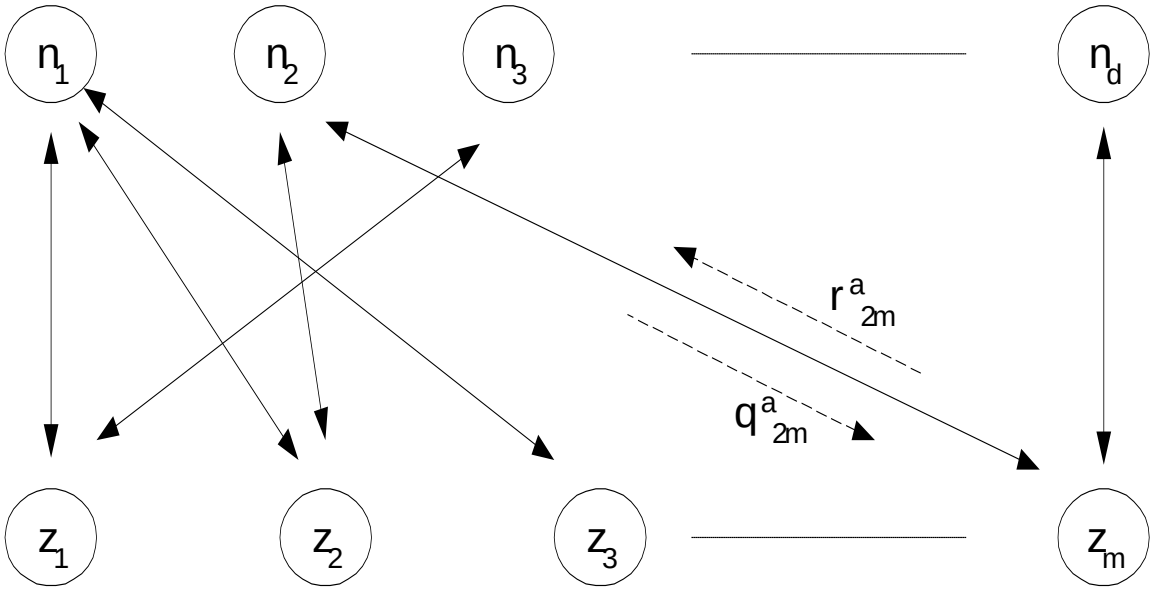

Figure 4.1. Message passing on bipartite graph

4.7.1. Algorithm

The algorithm is an iterative message passing algorithm. It is also known as "belief propagation" algorithm. For a (d,p,y) code, p messages originate from each bit node and y messages originate from each check node at each iteration. In other words, each message bit is checked by p check bits and each check bit has in common y message bits.

At each iteration, each bit node n_i sends messages q^a_{ij} to p check nodes which are the targets of the bidirectional rows originating from node n_i in Figure 4.4. These messages indicate the bit node's belief that it has value **a** given the messages received from (p-1) check nodes other than node j. Similarly, each check node z_j sends messages r^a_{ij} to y bit nodes which are the targets of the bidirectional rows originating from node z_j in Figure 4.3. These messages indicate the check node's belief that it is satisfied with bit node i value, **a**, given the messages received from (y-1) bit nodes other than node i.

After each iteration, a tentative decoding result indicating the value of noise sequence is obtained. This result is improved each time. The more the number of iterations is, the better the result is. The algorithm ends when the decoding achieves the observed syndrome vector or the permitted number of iterations exceeds which is failure. The necessary number of iterations for the algorithm to reach to the correct solution is proportional to d and p. This is why p is kept small. Figure 4.5 shows an example of error correction using LDPC codes in thirteen steps.

4.7.2. Algorithm Initialization

The algorithm starts by assigning q^a_{ij} its value from the likelihood ratio $\Lambda(n_i)=f(n_i|1)/f(n_i|-1)$ as $p_l^1 = q^1_{ij} = \frac{\Lambda(ni)}{\Lambda(ni)+1}$ and $p_l^0 = q^0_{ij} = \frac{1}{\Lambda(ni)+1}$. These are the soft detected values. Therefore, we see that LDPC codes utilize **soft decoding**, which was mentioned before and hence achieve coding gain.

4.7.3. Horizontal Step

This step is the check step when we compute the probabilities of observed value of z_m when n_l=0 and n_l=1 as:

$$r0_{ml} = \sum_{\{n_{l'} : l' \in L(m/l)\}} P(z_m \mid n_l = 0, \{n_{l'} : l' \in L(m/l)\}) \prod_{l' \in L(m)/l} q_{m l'}^{n_{l'}} \quad (4.1)$$

$$r^1_{ml} = \sum_{\{n_{l'} : l' \in L(m/l)\}} P(z_m \mid n_l = 1, \{n_{l'} : l' \in L(m/l)\}) \prod_{l' \in L(m)/l} q_{m l'}^{n_{l'}} \quad (4.2)$$

Bits other than bit l have a seperable distribution given by the probabilities $\{q^0_{ml'}, q^1_{ml'}\}$.

4.7.4. Vertical Step

This step takes the computed r^0_{ml} and r^1_{ml} and updates the bit confidence values q^0_{ml} and q^1_{ml} as:

$$q^0_{ml} = p_l{}^0 \frac{\prod_{m' \in M(l)/m} r^0_{m'l}}{\prod_{m' \in M(l)/m} r^0_{m'l} + \prod_{m' \in M(l)/m} r^1_{m'l}} \quad (4.1)$$

and

$$q^1_{ml} = p_l{}^1 \frac{\prod_{m' \in M(l)/m} r^1_{m'l}}{\prod_{m' \in M(l)/m} r^0_{m'l} + \prod_{m' \in M(l)/m} r^1_{m'l}} \quad (4.2)$$

4.7.5. Decoding

After each iteration, decoding procedure sets n_l to 1 if $q_1^l > 0.5$ or in other words sets n_l to 0 if $q_1^l < 0.5$ and decides if the checks **Hn** = **z** *mod 2* are all satisifed. The algorithm halts when the checks are satisfied.

For complete reference of decoding algorithm refer to [5].

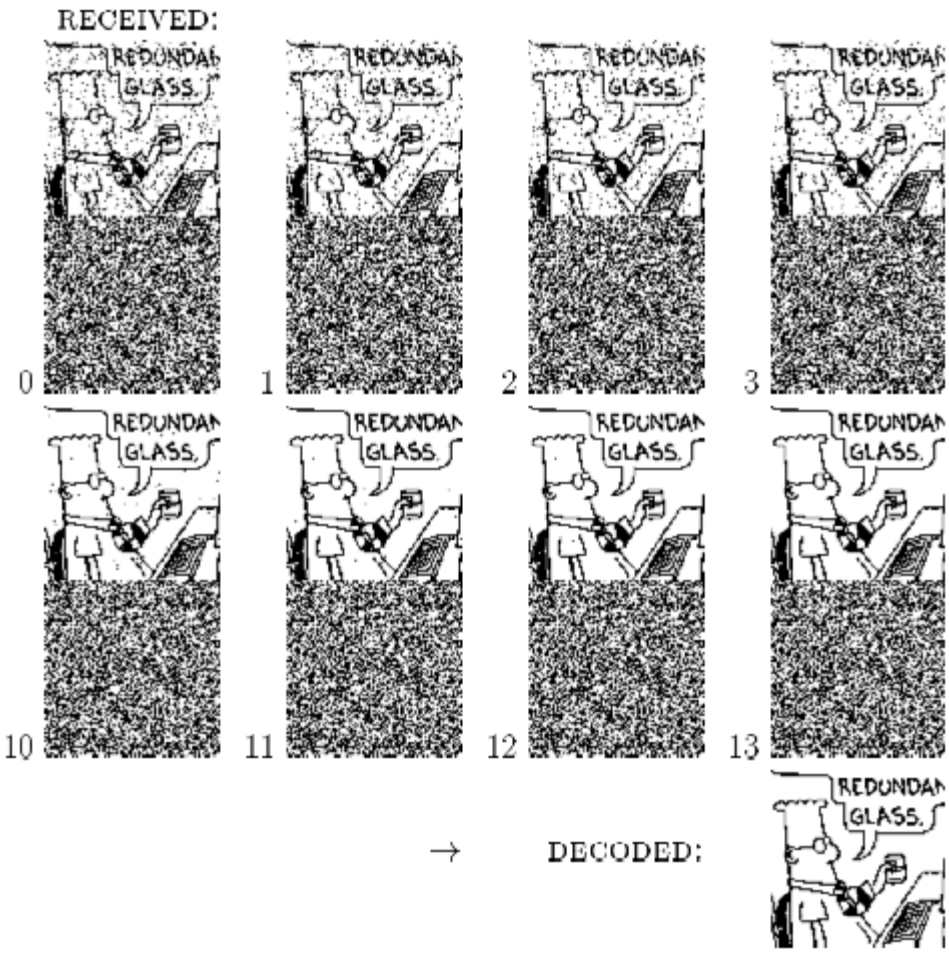

Figure 4.1. A cartoon with 7.5 percent error corrected by LDPC codes

5. SIMULATIONS AND RESULTS

5.1. Introduction

In our simulations, we did tests on nine different images given in Appendix A. To obtain the final results, we discarded the best and worst two results and took the average of the remaining results.

We measured the performance vis-a-vis:

- Detector structure: correlation detector variants, ML detector
- Coding strategy: repetition coding, BCH coding, LDPC coding
- Insertion strength (γ)
- Insertion domain: DFT, block-DCT

The results in tabular form are given in Appendix B. In Sections 5.2-5.5, we report the results with the DFT technique and in Section 5.6, we compare the relevant results of the DFT domain with those of the DCT domain.

5.2. Performance with Different Detectors

In this part, our goal is to investigate if it pays off using maximum likelihood detectors instead of correlation and covariance detectors. We also compare the BER performance of the covariance detector with the BER performance of the correlation detector.

The BER results of the repetition-only-coded DFT technique for $\gamma = 0.2$ are plotted in Figure 5.1. We embed 256, 512 and 1024 bits and hence 256, 128 and 64 pixels/bit, respectively.

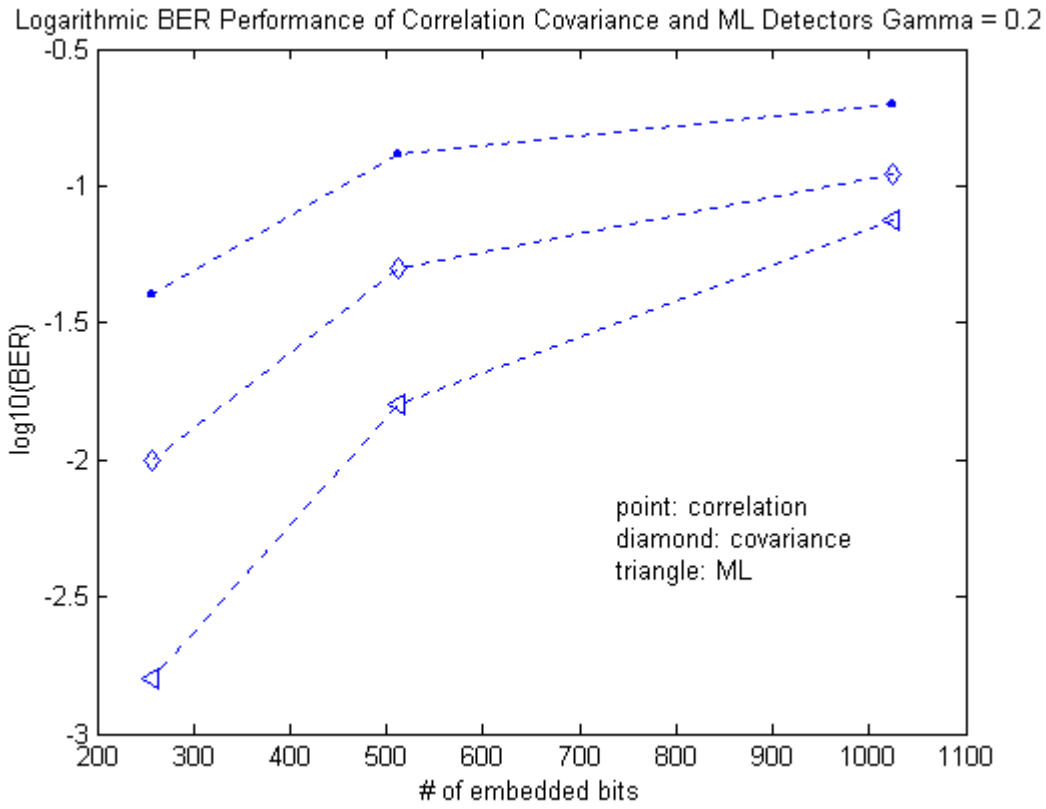

Figure 5.1. Logarithmic BERs of the repetition-only coded DFT method

BER obviously deteriorates with increasing number of bits since the number of pixels per bit decreases inversely proportionally to the number of embedded bits. This decreases the energy per bit and hence results in higher BERs.

The covariance detector performs better than the correlation detector; because subtracting the mean of the watermarked coefficients and the random watermark sequence, cancels the interference effect of the DC component of the host image. As expected, the optimum ML detector performs the best. For the chip rate of 256, it gives 25 times superior BER performance than the correlation detector and six times superior BER performance than the covariance detector. When we compare the BERs of the ML detector with 512 and 1024 embedded bits with the BERs of the covariance detector with 256 and 512 embedded bits, respectively, we see that they are almost equal. Hence, we can claim that ML detector achieves twice capacity. This performance differential hads us consider that it is worth using the ML detector. With the employment of error correction codes, this performance differential will widen even further.

BER results of the ML detector for three different watermark strengths are given in Figure 5.2.

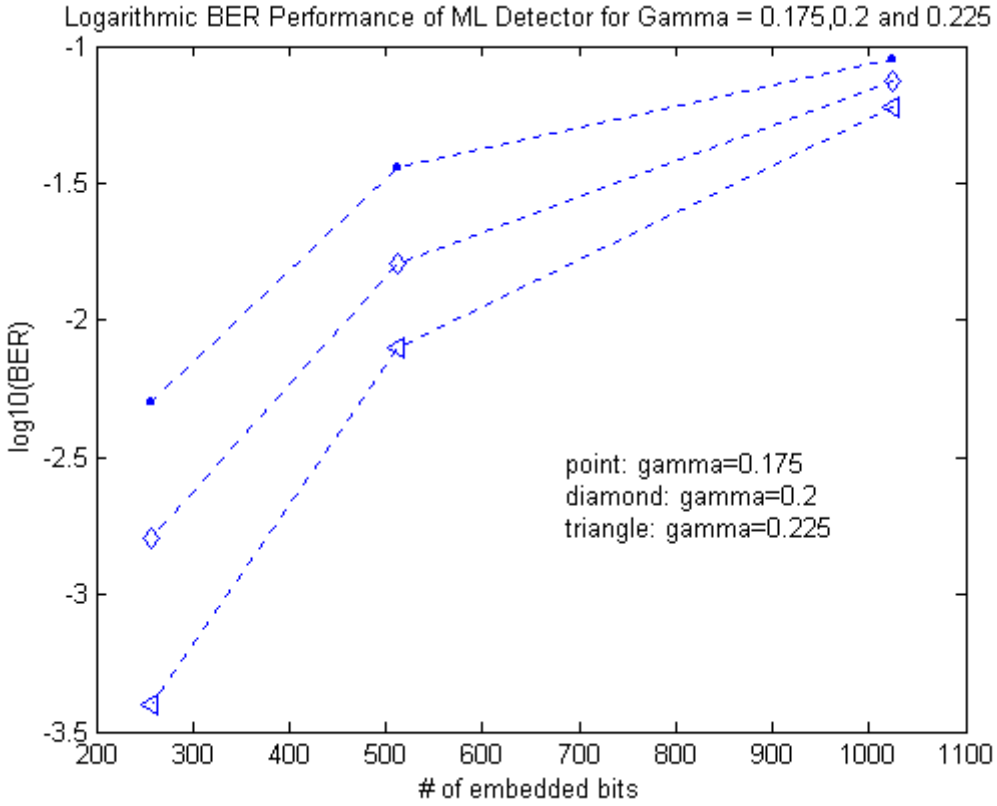

Figure 5.2. Logarithmic BER results of ML detector for the DFT technique

The results of the previous test have shown that ML detector designed to be optimal in the Bayes sense is the best detector, which decreases the BER and hence increases the capacity of the watermarking system. In order to see whether error correction codes can further increase the capacity, we applied BCH and LDPC codes together with ML detector.

5.3. Performance with BCH Codes

As discussed before, BCH codes with large block lengths are known to perform better than BCH codes with small block lengths. In order to see this fact experimentally, we applied BCH codes with varying block lengths by partitioning the watermark message into several different length blocks accordingly. The BER results obtained from (15,7,2),

(31,16,3), (63,30,6), (127,64,10), (255,123,19) and (511,250,31) BCH codes for $\gamma = 0.175$ and $\gamma=0.2$ are given in Figure 5.3 and Figure 5.4, respectively. For $\gamma = 0.2$, getting the simulation results of (255,123,19) and (511,250,31) BCH codes for ~250 bits were impractical due to very long simulation time. Therefore, we supplemented the semianalytical results of these codes for these rates. Experimental results obtained for 256, 512 and 1024 bits are plotted together in Figure 5.5.

To confirm the correctness of this approach, we compare the experimental results with the semi-analytical ones. We do this only for ~250 bit results for illustration purpose in Table 5.1. As the semi-analytical approach is obviously based on the assumption that the raw error rate has been correctly estimated, for channel error rates we took the experimental ML 512 bit results from Table B.1 in Appendix B as $p = 0.04$ for $\gamma=0.175$ and $p = 0.022$ for $\gamma=0.2$.

Table 5.1. Semi-analytical and experimental BCH codes BER for the DFT method

Coding	Gamma = 0.175 p=0.04 (Theoretical)	Gamma = 0.175 (Experimental)	Gamma = 0.2 p=0.022 (Theoretical)	Gamma = 0.2 (Experimental)
15,7,2	0.0044	0.005	0.0008	0.0014
31,16,3	0.0049	0.0047	0.0006	0.0009
63,30,6	0.0016	0.003	0.000055	0.00002
127,64,10	0.0013	0.002	0.000011	-
255,123,19	0.00033	0.0008	0.00000011	-
511,250,31	0.00072	0.001	0.000000015	-

Comparing the semi-analytical and the experimental results in Table 5.1, we see that they are of the same order. Hence, we can say that Equation 4.7 can be used to predict coding performance whenever obtaining simulation results is impractical, provided that an accurate estimate of raw channel error rate is available. Another observation from Table 5.1 is that (255, 123, 19) BCH codes outperform (511, 250, 31) codes for $\gamma = 0.175$, whereas perform worse for $\gamma = 0.2$. In fact, from Equation 4.7 we can conclude that for different ranges of p, different BCH codes perform better. We determined the best one of the six given BCH codes with all possible channel error rates as:

- $p < 0.032$ => best code = BCH (511,250,31)
- $0.032 < p < 0.057$ => best code = BCH (255,123,19)
- $0.057 < p < 0.068$ => best code = BCH (63,30,6)
- $p > 0.068$ => best code = BCH (15,7,2)

For $\gamma = 0.175$, $p = 0.04$. Therefore (255,123,19) is better than (511,250,31). For $\gamma = 0.2$ though, $p = 0.022$ and hence (511,250,31) is better than (255,123,19). From these, we conclude that BCH codes with larger block lengths may perform worse than BCH codes with smaller block lengths when we go to higher channel error rates. Hence, one needs to consider the channel error rate before selecting a particular BCH code from an ensemble of BCH codes.

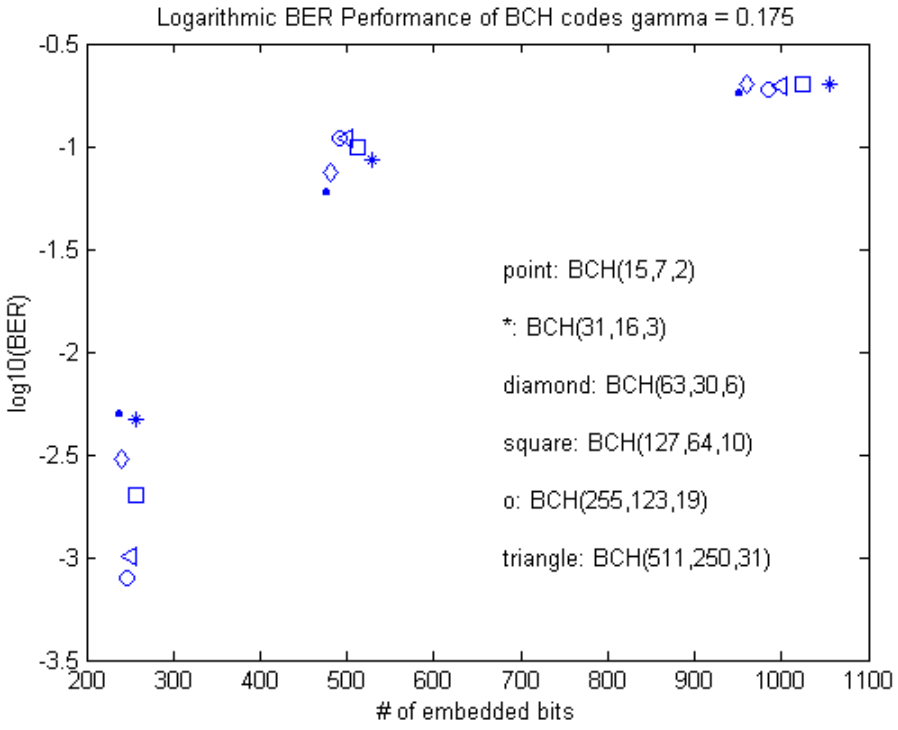

Figure 5.1. Logarithmic BER results of ML detector + BCH codes

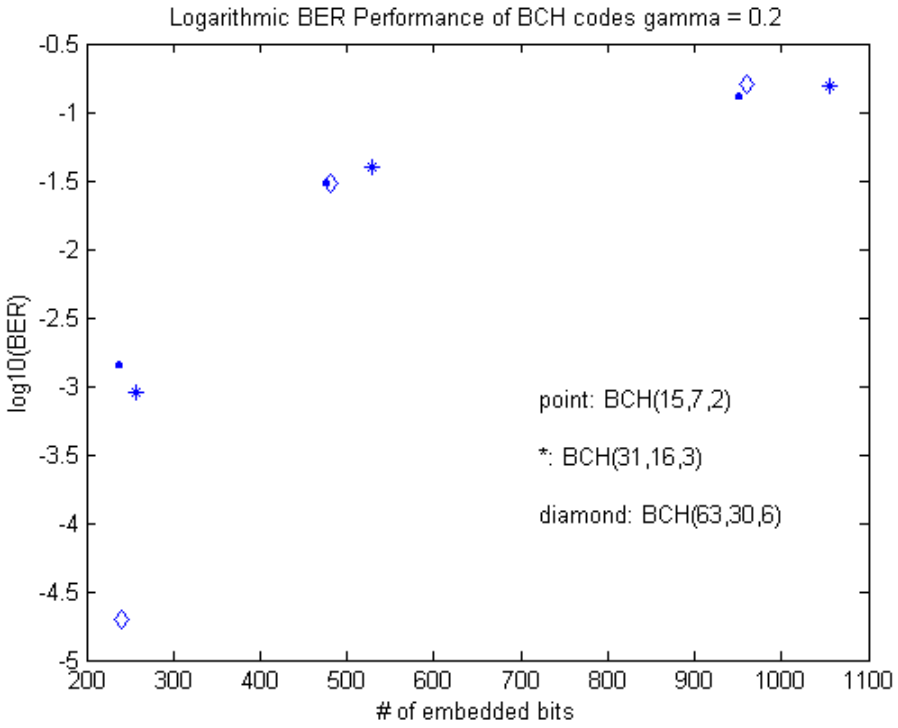

Figure 5.2. Logarithmic BER results of ML detector + BCH codes

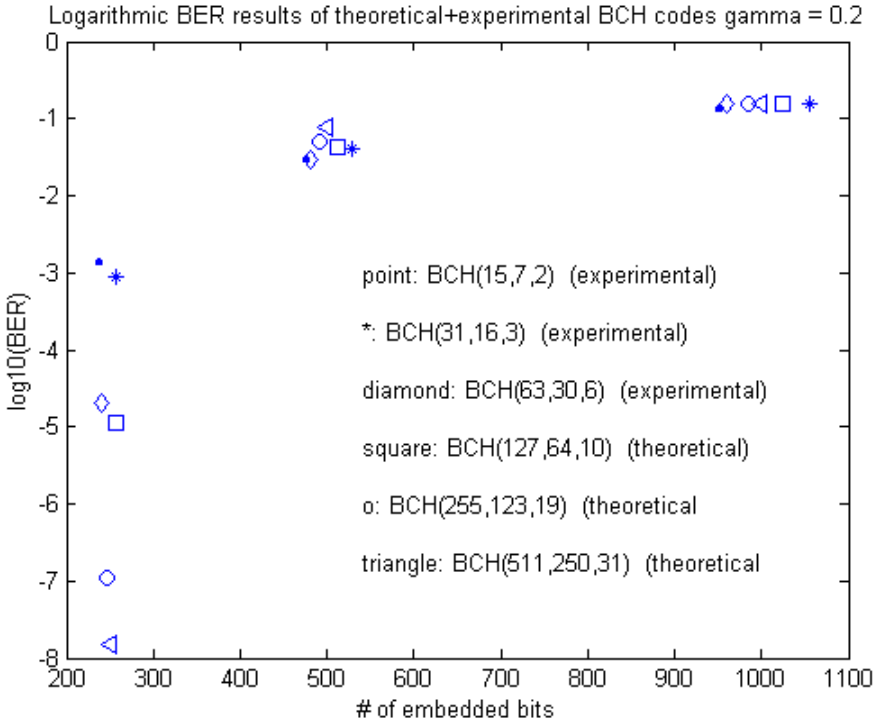

Figure 5.3. Combined theoretical-experimental BERs of ML detector + BCH codes

5.4. Performance with LDPC Codes

In this section, we look at the performance of LDPC codes and compare it with the performance of the repetition-only codes and with the performance of the best BCH-codes. The results for $\gamma = 0.175$ and $\gamma = 0.2$ are plotted in Figure 5.6 and 5.7 respectively from which we observe the following:

BCH coded watermarking systems are better than the repetition-only ones for approximately less than 320 embedded bits (more than 196 pixels/bit) for $\gamma = 0.175$ and 480 bits (more than 136 pixels/bit) for $\gamma = 0.2$. One can explain this as follows: Coding can be seen as an investment. First, you pay by adding redundancy, and then try to earn more than you invest by coding gain. If the coding gain is less than the loss due to the added redundancy, block coded systems perform worse than the repetition-only coded systems.

BCH codes do not perform well at high channel error rate conditions. When the number of embedded bits increases, so does the error rate. When a certain bit rate is reached, BCH coding is not able to compensate for the introduced redundancy.

Fixing the acceptable BER limit to 10^{-3} again, LDPC-coded systems perform significantly better than repetition-only and BCH-coded systems. We see that, it is possible to embed about 550 bits with watermark strength 0.175 and it is possible to embed about 680 bits with watermark strength 0.2 which is less than 250 bits for the repetition-only coded systems. Hence, LDPC-coded systems achieve about three times more capacity than repetition-only coded systems with the discussed DFT watermarking system.

The comments that can be made for the LDPC codes are similar to the ones for BCH codes. LDPC-coded systems also perform worse than the repetition-only ones when the channel error rate of the implemented system is so high that the coding gain turns out to be less than the loss due to added redundancy. However, from Figure 5.6 and Figure 5.7, we deduce that watermarking channels can benefit from LDPC codes at higher error rates as compared to BCH codes.

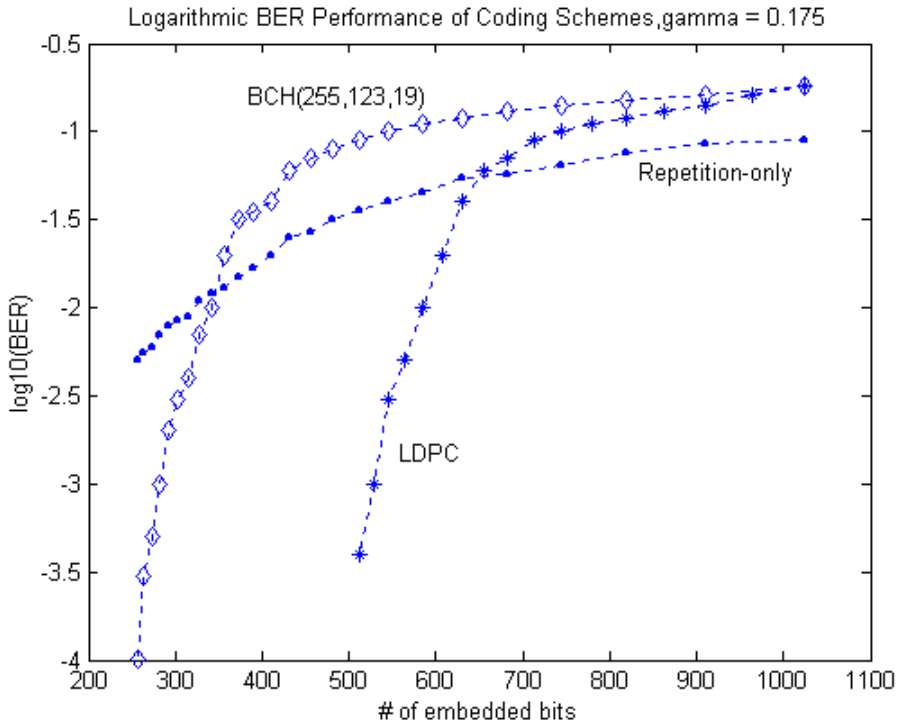

Figure 5.1. Coding results for $\gamma = 0.175$

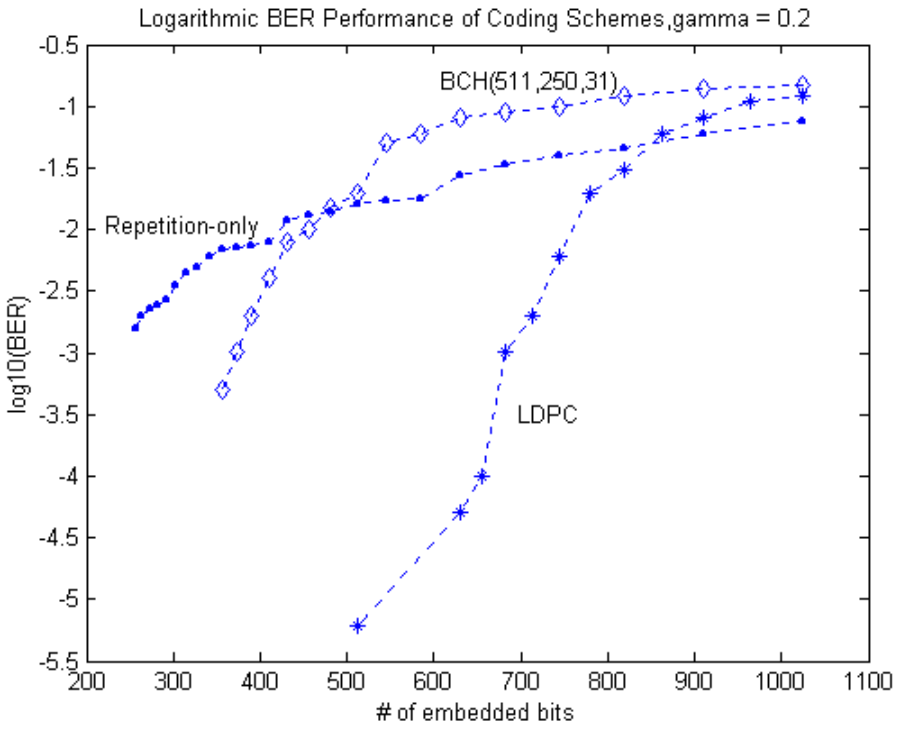

Figure 5.2. Coding results for $\gamma = 0.2$

5.5. Performance with Insertion Strength

Like in every communications system, watermarking systems also depend on the transmitted signal energy since BER decreases when signal energy is increased. Watermark energy increases linearly with γ. In Figure 5.8, BER test results of the repetition-only and LDPC coded DFT techniques as a function of γ are given. It is interesting to see that repetition-only logarithmic BER decreases approximately linearly whereas LDPC logarithmic BER decreases at a much higher rate. As the channel condition is improved by an application of higher γ, LDPC coding brings in further improvement.

As we increase the watermark strength to achieve more reliable system, we decrease the perceptual quality of the marked image, which is the second performance criterion of the watermarking techniques. "Peak signal to noise ratio" (PSNR) and "Watermark to document ratio" (WDR) values, which were explained before, are given in Table 5.2.

Table 5.1. PSNR and WDR Results

γ	PSNR (dB)	WDR (dB)
0.175	42.78	-37.87
0.2	41.65	-36.74
0.225	40.65	-35.74
0.25	39.75	-34.84

The PSNR value for a marked image must be more than 38dB for perceptual fidelity. As observed from Table 5.2, the given four watermark strengths fullfil this requirement and the PSNR value decreases with increasing γ. However, we should remind that the results in Table 5.2 are the average values of the used nine images. The actual results may vary considerably from image to image. Therefore, one should allow some margin and mark with a slightly smaller γ. Considering that for γ=0.25 PSNR is about 40dB, giving some margin, γ=0.2 or at most γ=0.225 are good choices.

An example of an original and marked image with watermark strength 0.275, which is even more than the used strengths, is shown in Figure 5.9. Figure 5.10 shows the enhanced difference image.

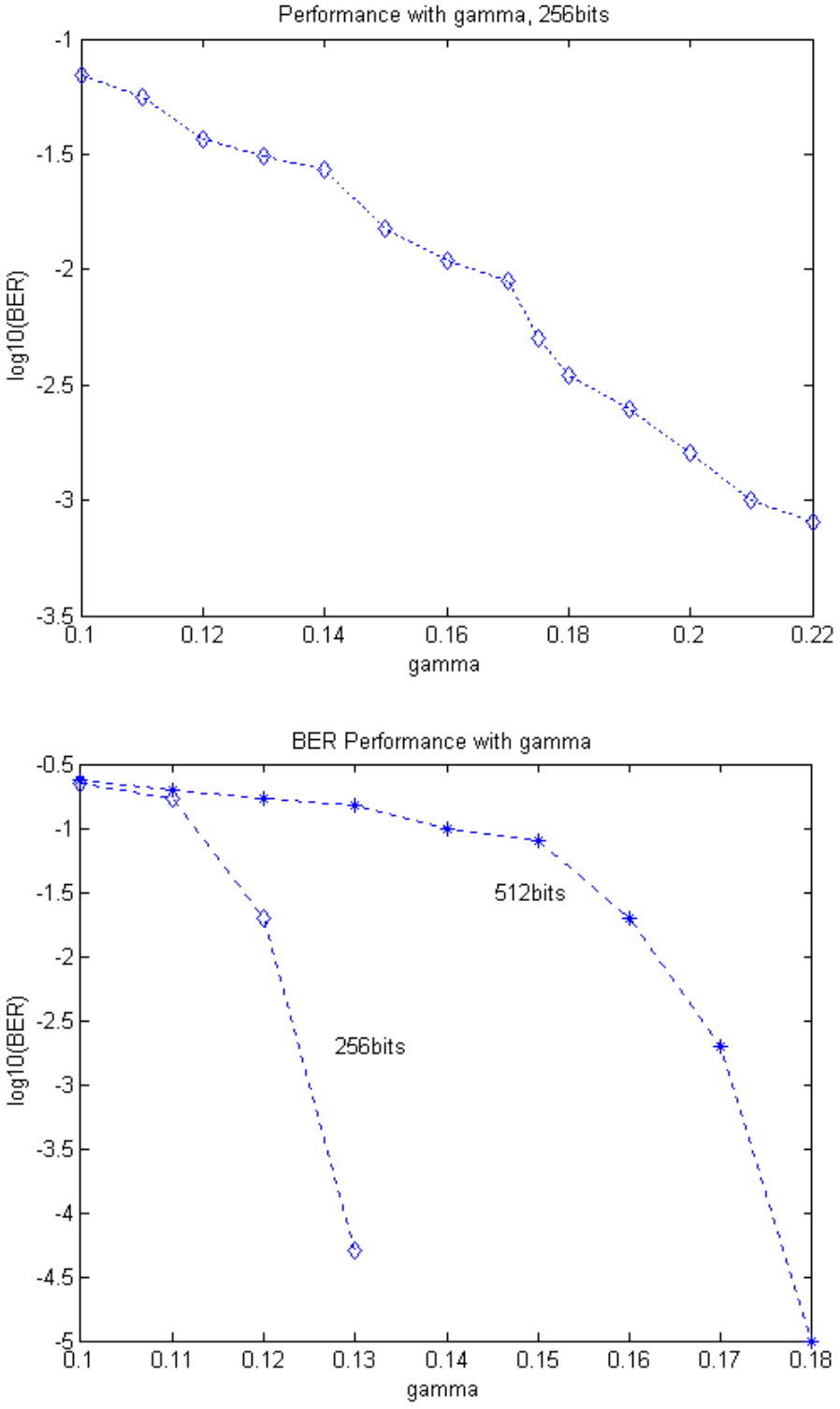

Figure 5.1. BER of repetition-only (upper figure) and LDPC coded (lower figure) DFT techniques as a function of γ

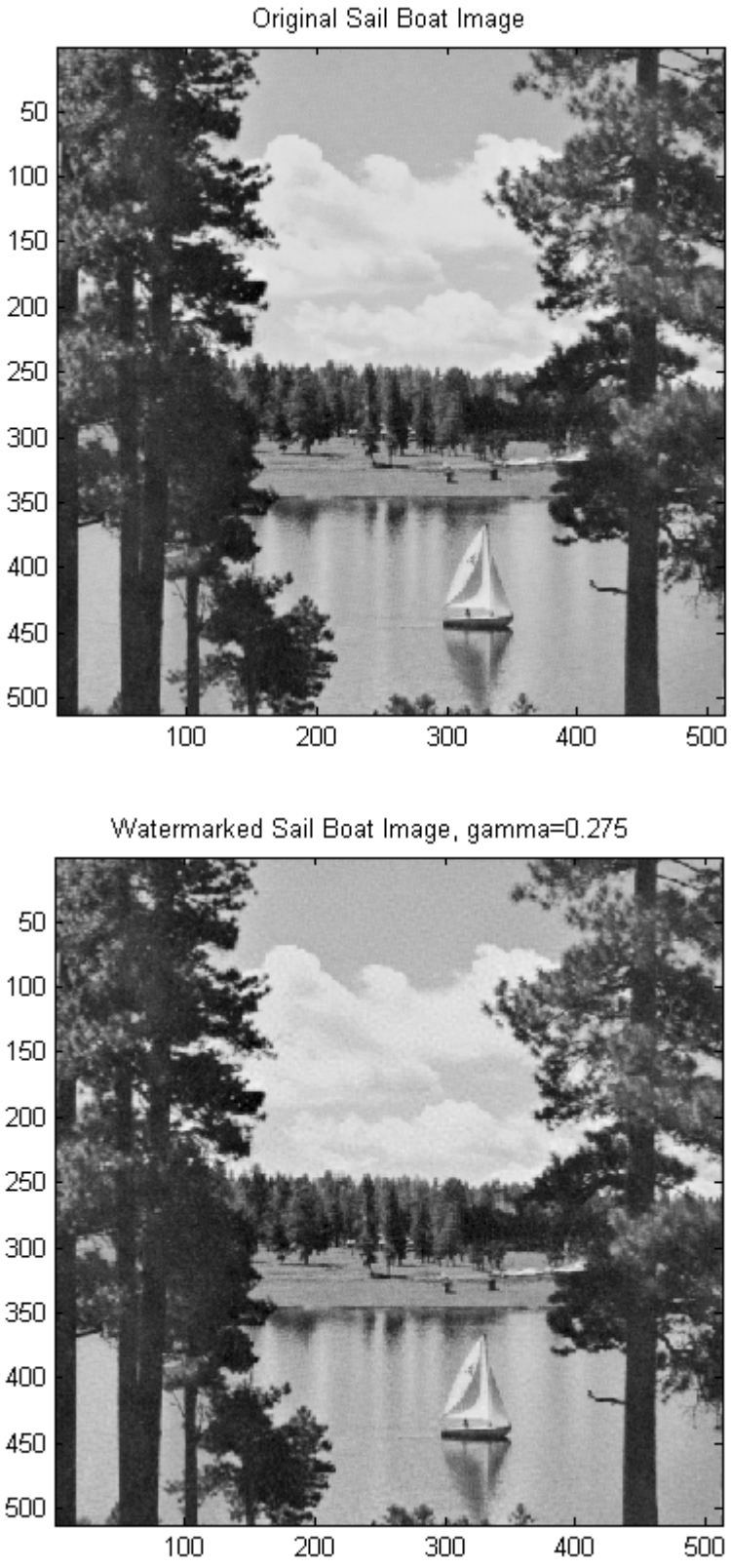

Figure 5.2. Original (upper figure) and DFT watermarked(lower figure) images

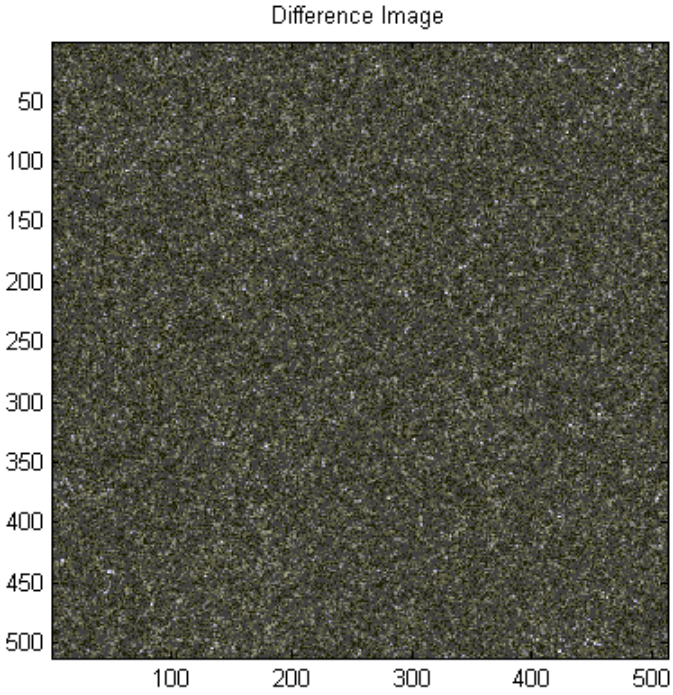

Figure 5.3. Enhanced watermark (difference) image

5.6. Comparison Between Embedding Domains

In this section, we compare the performance with respect to the embedding domain, that is DFT bandpass coefficients versus block DCT bandpass coefficients. BER test results of the DCT technique are given in Figure 5.11 and Figure 5.12. We show that ML detector based on generalized Gaussian statistics outperforms the correlation and covariance detectors. Similarly, as expected, LDPC coding proves also to be the most advantageous coding scheme for also block DCT embedding.

The merged BER results of DFT and DCT techniques are given in Figure 5.13. When we compare the two, we see that DFT method achieves significantly better results than the block DCT method. The reasons why the ML detector applied on DFT coefficients achieves better results than the ML detector applied on block DCT coefficients can be argued as follows:

The Weibull statistics may be a better representation of DFT magnitudes than the generalized Gaussian statistics for the block DCT coefficients.

As will be explained in the text, the additive noise coming from the cover DCT coefficients is higher than that coming from the DFT magnitudes. This is because of the higher variance of DCT coefficients.

Table 5.1. Capacity of the DFT and DCT methods @BER = 10^{-3}

Gamma	DFT Capacity	DCT Capacity
0.175	550	-
0.2	680	320

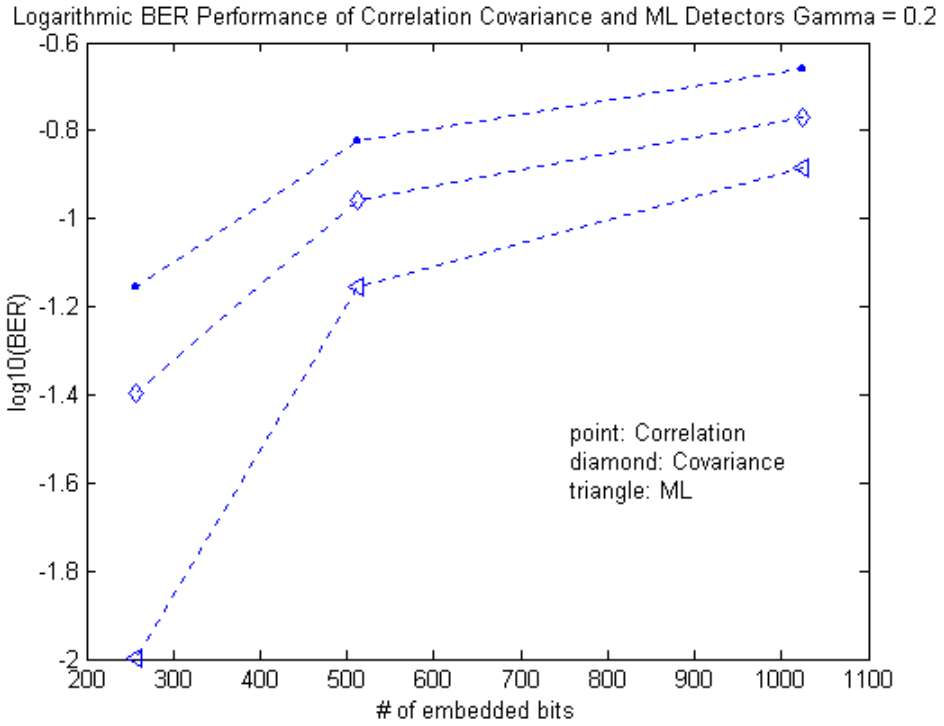

Figure 5.1. Logarithmic BER results of the DCT technique

For completeness, we give the PSNR and WDR results of the block DCT method in Table 5.4.

Table 5.2. PSNR and WDR results

Gamma	PSNR(dB)	WDR(dB)
0.175	42.75	-37.84
0.2	41.61	-36.70
0.225	40.60	-35.69
0.25	39.70	-34.78

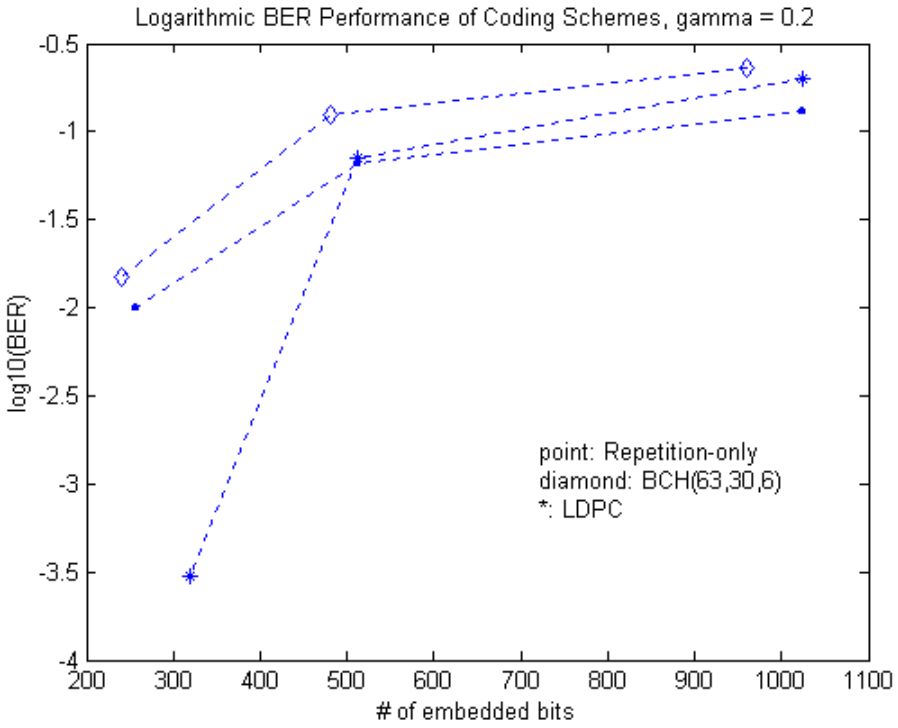

Figure 5.2. Logarithmic BER results of the DCT technique

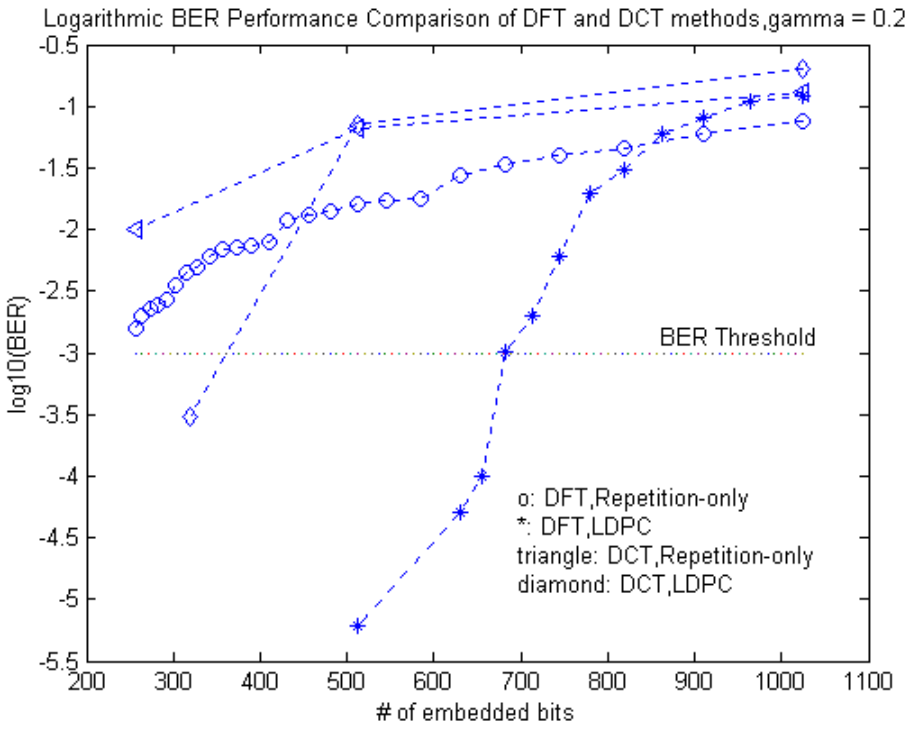

Figure 5.3. BER Comparison of DFT and DCT techniques

An example of an original and marked image with watermark strength 0.275, which is even more than the used strengths, is shown in Figure 5.14 and Figure 5.15. Figure 5.16 shows the exaggerated difference image. This figure shows that block DCT method adapts very well to image characteristics.

5.7. Interpretation of the Results

In this section, we demonstrate the reproduction of simulation results **from only the raw host images**, namely without doing any marking, with the assumption of a perfect correlation detector that should give the same performance as the performance of covariance detector in the simulations. The correlation detector output for any one bit can be obtained as: $|\tilde{\mathbf{I}}_{\mathbf{w}}(\mathbf{i})| \bullet \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) = |\tilde{\mathbf{I}}(\mathbf{i})| \bullet \tilde{\mathbf{p}}_{\mathbf{c}}(\mathbf{i}) + (\gamma |\tilde{\mathbf{I}}(\mathbf{i})| . \mathbf{U})$ c(i). The first term on the right is the random **noise** signal which depends on the deterministic host data and the specific random watermark sequence whereas the second term is the deterministic **watermark** signal that depends on only the host data. Opening this equation, we obtain the expected value of the signal to noise ratio for any one bit as:

$$\mathrm{SNR} = \mathrm{E}\{\frac{(\gamma\sum_{j=1}^{ch}|\mathrm{I(i)_j}|)^2}{(\sum_{j=1}^{ch}|\mathrm{I(i)_j}\,|\,\mathrm{p_c(i)_j})^2}\} = \frac{(\gamma\sum_{j=1}^{ch}|\mathrm{I(i)_j}|)^2}{\mathrm{E}\{(\sum_{j=1}^{ch}|\mathrm{I(i)_j}\,|\,\mathrm{p_c(i)_j})^2\}}.$$

$\mathrm{E}\{(\sum_{j=1}^{ch}|\mathrm{I(i)_j}\,|\,\mathrm{p_c(i)_j})^2\} = \sum_{j=1}^{ch}|\mathrm{I(i)_j}|^2$ since the expected values of the cross terms in the summation are 0. Therefore, the expected SNR ratio for any one bit is obtained as

$\mathrm{SNR} = \dfrac{(\gamma\sum_{j=1}^{ch}|\mathrm{I(i)_j}|)^2}{\sum_{j=1}^{ch}|\mathrm{I(i)_j}|^2}$. With the assumption that the spread DFT/DCT amplitude

coefficients ($|I(i)_j|.p_c(i)_j$) are normally distributed, the theoretical BER for this bit is equal to $Q(\sqrt{SNR}) = ½\ \mathrm{erfc}(\sqrt{SNR/2})$. In order to determine the BER, we take the mean of the obtained values over all bits.

We did tests on 6 different images and obtained the theoretical logarithmic BER results for DFT, block-DCT and, though not implemented, for DCT magnitudes. The results for 256 bits, in other words for 256 chips, for different watermark strengths are given below. Each row of "BERDFT", "BERDCT" and "BERblockDCT" matrices are the BER results corresponding to the image in the same row of "Images". "meanBERDFT", "meanBERDCT" and "meanBERblockDCT" are the averages of the results from 6 images.

Images = [baboon.bmp
airplane.bmp
goldhill.bmp
lena.bmp
peppers.bmp
sailboat.bmp]

WatermarkStrengths = [0.17 0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25]

BERDFT =

[-1.7356 -1.8698 -2.0085 -2.1547 -2.3056 -2.4636 -2.6270 -2.7950 -2.9733
-1.6240 -1.7478 -1.8736 -2.0074 -2.1464 -2.2903 -2.4374 -2.5919 -2.7525
-1.6497 -1.7732 -1.9039 -2.0376 -2.1816 -2.3243 -2.4827 -2.6346 -2.7990
-1.5611 -1.6765 -1.7979 -1.9236 -2.0538 -2.1893 -2.3284 -2.4780 -2.6272
-1.6634 -1.7894 -1.9232 -2.0588 -2.2019 -2.3508 -2.5033 -2.6628 -2.8308
-1.6406 -1.7665 -1.8933 -2.0282 -2.1679 -2.3146 -2.4650 -2.6207 -2.7858]

BERDCT=

[-1.4865 -1.5930 -1.7059 -1.8208 -1.9434 -2.0679 -2.1984 -2.3326 -2.4722
-1.3207 -1.4104 -1.5056 -1.6040 -1.7029 -1.8109 -1.9213 -2.0369 -2.1470

-1.4850 -1.5920 -1.7044 -1.8196 -1.9434 -2.0683 -2.1977 -2.3341 -2.4743
-1.5274 -1.6385 -1.7571 -1.8783 -2.0037 -2.1376 -2.2733 -2.4148 -2.5615
-1.4070 -1.4962 -1.6032 -1.7069 -1.8184 -1.9374 -2.0590 -2.1741 -2.2999
-1.4269 -1.5255 -1.6325 -1.7425 -1.8568 -1.9740 -2.0955 -2.2261 -2.3518]

BERblockDCT =

[-1.3865 -1.4844 -1.5849 -1.6943 -1.8000 -1.9145 -2.0285 -2.1529 -2.2779
-0.9985 -1.0570 -1.1190 -1.1805 -1.2441 -1.3111 -1.3792 -1.4480 -1.5231
-1.3333 -1.4250 -1.5174 -1.6177 -1.7209 -1.8280 -1.9448 -2.0521 -2.1772
-1.0446 -1.1080 -1.1753 -1.2397 -1.3072 -1.3820 -1.4546 -1.5336 -1.6112
-1.0271 -1.0847 -1.1545 -1.2123 -1.2851 -1.3405 -1.4331 -1.5045 -1.5775
-1.2025 -1.2824 -1.3607 -1.4530 -1.5356 -1.6295 -1.7229 -1.8166 -1.9211]

meanBERDFT =

[-1.6458 -1.7705 -1.9000 -2.0351 -2.1762 -2.3221 -2.4740 -2.6305 -2.7948]

meanBERDCT =

[-1.4423 -1.5426 -1.6515 -1.7620 -1.8781 -1.9993 -2.1242 -2.2531 -2.3845]

meanBERblockDCT =

[-1.1654 -1.2403 -1.3186 -1.3996 -1.4821 -1.5676 -1.6605 -1.7513 -1.8480]

As observed, the BER performance of the DFT technique is the best and that of the block DCT is the worst. The mean BER values of the three techniques are demonstrated in Figure 5.17 and Figure 5.18.

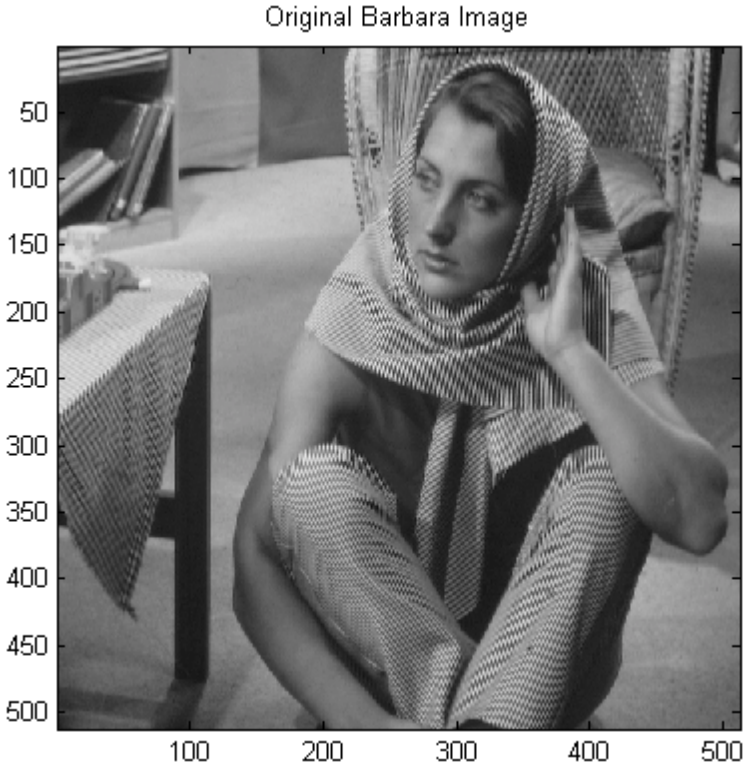

Figure 5.1. Original Barbara image

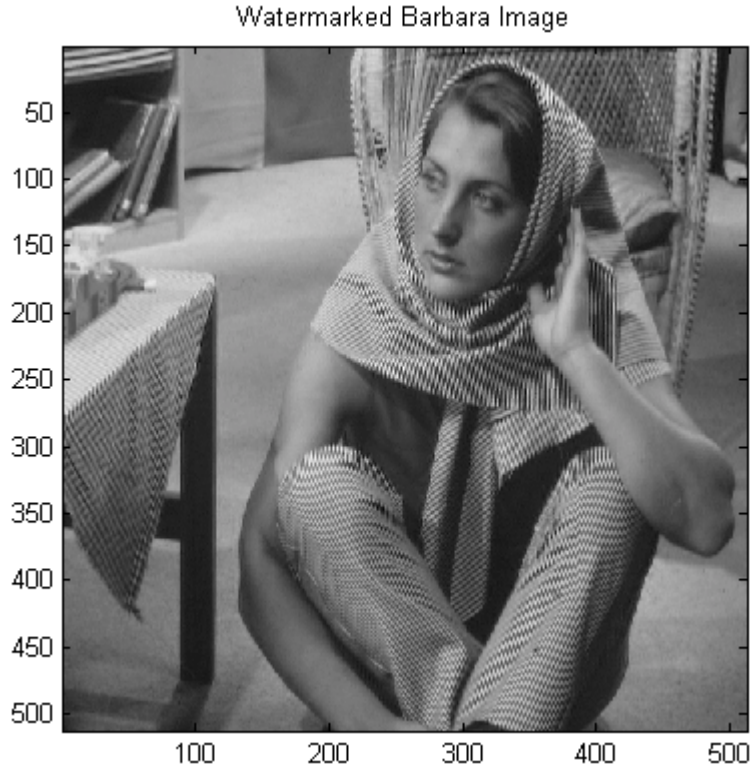

Figure 5.2. Watermarked Barbara image

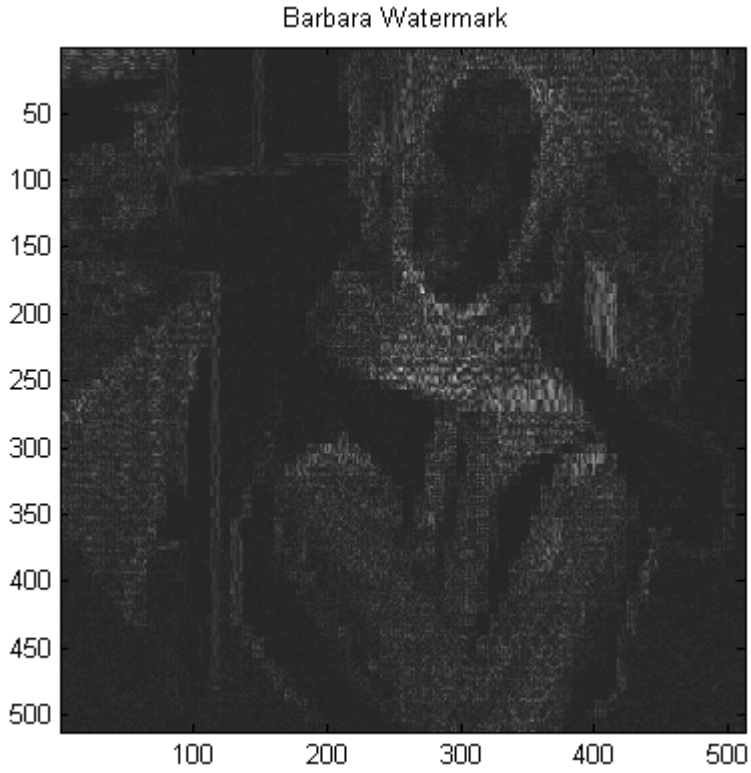

Figure 5.3. Enhanced watermark (difference) image

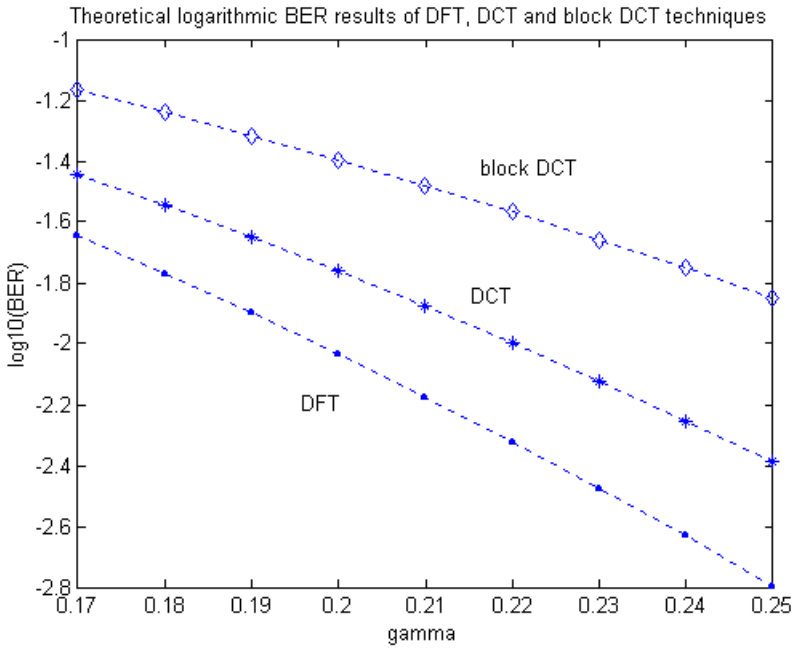

Figure 5.4. The theoretical BERs vs γ for 256 bits.

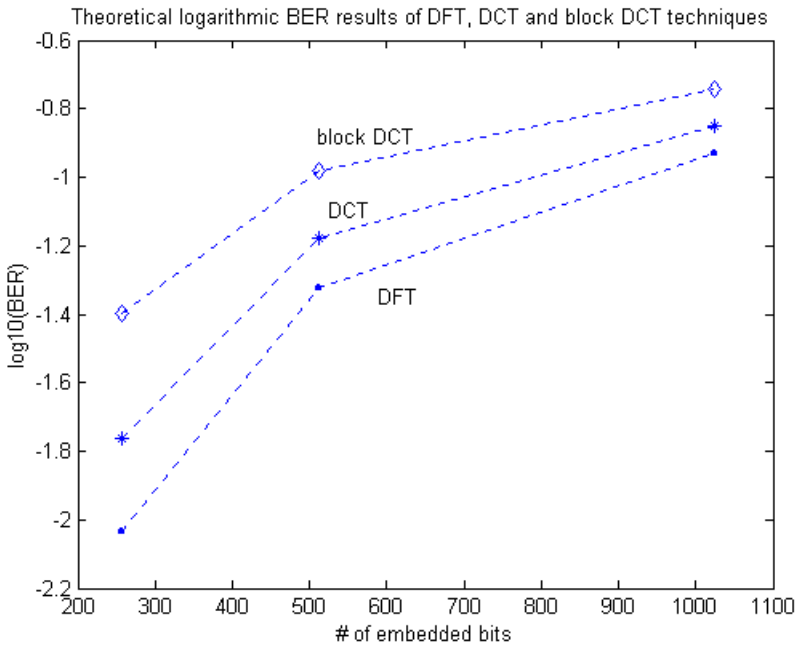

Figure 5.5. The theoretical BERs vs # of embedded bits for $\gamma = 0.2$

In Figure 5.19, and Figure 5.20 we demontrate the calculated BER results merged with the experimental correlation and covariance detector BER results for DFT and block-DCT techniques, respectively. The theoretical results match the covariance detector results perfectly from which we conclude that $|\mathbf{I_w}|.\mathbf{p_c}$ has a Gaussian distribution.

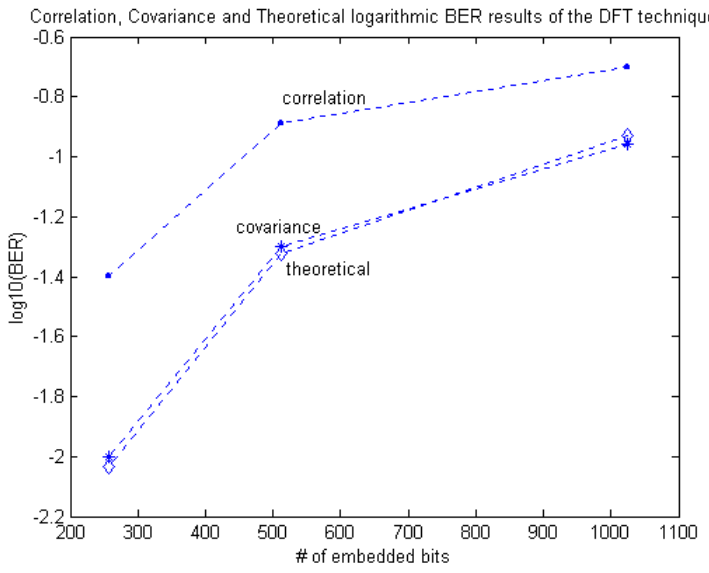

Figure 5.6. Theoretical and experimental covariance BER results of DFT technique

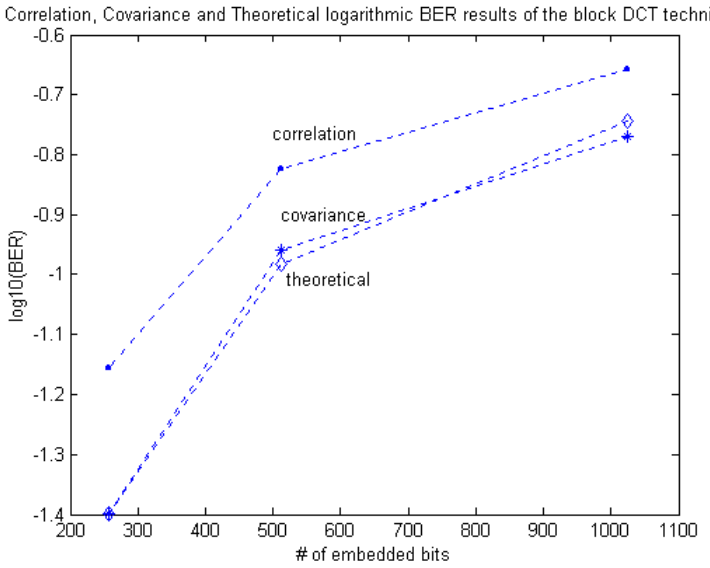

Figure 5.7. Theoretical and experimental covariance BER results of DCT technique

6. CONCLUSIONS

Watermarking channels are low SNR channels, and hence the bit error rate is very high. In this thesis, we proposed using ML detectors and LDPC error control codes to decrease the bit error rate or in other words to improve the meta-data carrying capacity of the "DFT magnitudes" and "8x8 block DCT coefficients" watermarking channels. For the DFT magnitude marking, we concluded that applying ML detectors and LDPC codes together improve the capacity significantly. In fact, using ML detector alone doubles the capacity and applying LDPC coding with ML detector further increases the capacity by three times and as a result six times more capacity is achieved. For the block DCT marking, we observe similar performance improvements when ML detector and LDPC coding are used. Nonetheless, the BER results and the achieved capacity with block DCT marking system are far worse than the relevant results for the DFT marking system since the absolute DCT coefficients have much higher variance than the DFT amplitudes, which increases the carrier noise. In brief, though previously our goal was to evaluate the performance of LDPC codes as compared to BCH codes and repetition-only codes, we also cocluded that the capacity improvement by LDPC codes depends also on the performance of the selected detector and the applied marking method.

Future research for obtaining higher capacity from watermarking channels may focus on

- the investigation and comparison of Turbo codes, regular and irregular Gallager (LDPC) codes and Mackay-Neal codes [5],
- the utilization of HVS models for adaptively embedding the watermark with varying watermark strength on each marked coefficient,
- the application of pre-detection filtering, particularly high pass filtering, on the marked data for the cancellation of noise resulting from the cover data and concomitant attacks.

APPENDIX A: USED IMAGES

Figure A.1. 512x512 grayscale airplane image

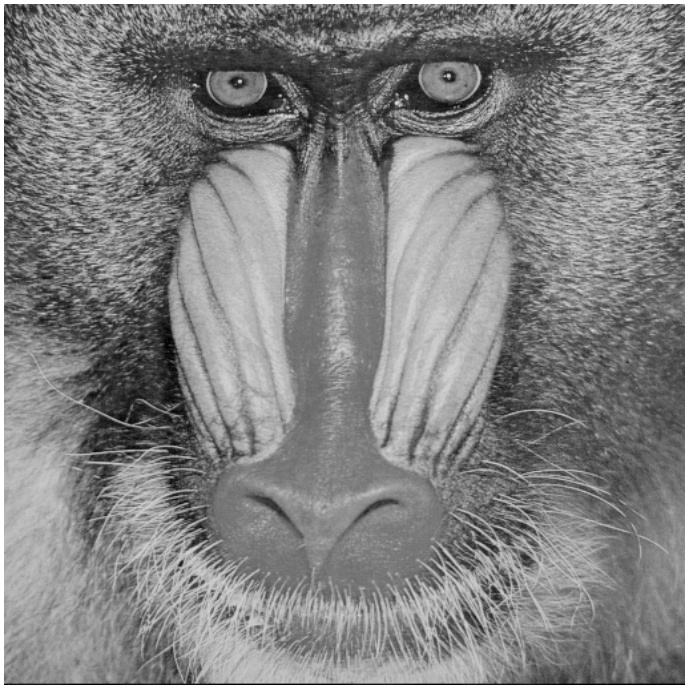

Figure A.2. 512x512 grayscale baboon image

Figure A.3. 512x512 grayscale Barbara image

Figure A.4. 512x512 grayscale boat image

Figure A.5. 512x512 grayscale couple image

Figure A.6. 512x512 grayscale Goldhill image

Figure A.7. 512x512 grayscale Lena image

Figure A.8. 512x512 grayscale peppers image

Figure A.9. 512x512 grayscale sailboat image

APPENDIX B: SIMULATION RESULTS

Table B.1. DFT technique simulation results

METHOD	γ	# OF BITS	Pixels/bit	BER
Correlation	0.2	1024	64	0.2
Correlation	0.2	512	128	0.13
Correlation	0.2	256	256	0.04
Covariance	0.2	1024	64	0.11
Covariance	0.2	512	128	0.05
Covariance	0.2	256	256	0.01
ML	0.1	256	256	0.0700
ML	0.11	256	256	0.0560
ML	0.12	256	256	0.0370
ML	0.13	256	256	0.0310
ML	0.14	256	256	0.0270
ML	0.15	256	256	0.0150
ML	0.16	256	256	0.0110
ML	0.17	256	256	0.0090
ML	0.175	1024	64	0.09
ML	0.175	910	72	0.086
ML	0.175	819	80	0.076
ML	0.175	745	88	0.065
ML	0.175	683	96	0.057
ML	0.175	630	104	0.055
ML	0.175	585	112	0.045
ML	0.175	546	120	0.04
ML	0.175	512	128	0.036
ML	0.175	482	136	0.032
ML	0.175	455	144	0.027

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML	0.175	431	152	0.025
ML	0.175	410	160	0.02
ML	0.175	390	168	0.017
ML	0.175	372	176	0.015
ML	0.175	356	184	0.013
ML	0.175	341	192	0.012
ML	0.175	328	200	0.011
ML	0.175	315	208	0.009
ML	0.175	303	216	0.0085
ML	0.175	293	224	0.008
ML	0.175	283	232	0.007
ML	0.175	273	240	0.006
ML	0.175	264	248	0.0055
ML	0.175	256	256	0.005
ML	0.18	256	256	0.0035
ML	0.19	256	256	0.0025
ML	0.2	1024	64	0.0750
ML	0.2	910	72	0.0600
ML	0.2	819	80	0.0460
ML	0.2	745	88	0.0400
ML	0.2	683	96	0.0340
ML	0.2	630	104	0.0280
ML	0.2	585	112	0.0180
ML	0.2	546	120	0.0170
ML	0.2	512	128	0.0160
ML	0.2	482	136	0.0140
ML	0.2	455	144	0.0130
ML	0.2	431	152	0.0120
ML	0.2	410	160	0.0080
ML	0.2	390	168	0.0075

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML	0.2	372	176	0.0072
ML	0.2	356	184	0.0070
ML	0.2	341	192	0.0060
ML	0.2	328	200	0.0050
ML	0.2	315	208	0.0045
ML	0.2	303	216	0.0035
ML	0.2	293	224	0.0027
ML	0.2	283	232	0.0024
ML	0.2	273	240	0.0023
ML	0.2	264	248	0.0020
ML	0.2	256	256	0.0016
ML	0.21	256	256	0.0010
ML	0.22	256	256	0.0008
ML	0.225	1024	64	0.06
ML	0.225	512	128	0.008
ML	0.225	256	256	0.0004
ML+BCH(15,7,2)	0.175	952	32	0.18
ML+BCH(15,7,2)	0.175	476	64	0.06
ML+BCH(15,7,2)	0.175	238	128	0.005
ML+BCH(15,7,2)	0.2	952	32	0.13
ML+BCH(15,7,2)	0.2	476	64	0.03
ML+BCH(15,7,2)	0.2	238	128	0.0014
ML+BCH(31,16,3)	0.175	1056	32	0.2
ML+BCH(31,16,3)	0.175	528	64	0.086
ML+BCH(31,16,3)	0.175	256	128	0.0047
ML+BCH(31,16,3)	0.2	1056	32	0.153
ML+BCH(31,16,3)	0.2	528	64	0.04
ML+BCH(31,16,3)	0.2	256	128	0.0009
ML+BCH(63,30,6)	0.175	960	32	0.2
ML+BCH(63,30,6)	0.175	480	64	0.075

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML+BCH(63,30,6)	0.175	240	128	0.003
ML+BCH(63,30,6)	0.2	960	32	0.16
ML+BCH(63,30,6)	0.2	480	64	0.03
ML+BCH(63,30,6)	0.2	240	128	0.00002
ML+BCH(127,64,10)	0.175	1024	32	0.2
ML+BCH(127,64,10)	0.175	512	64	0.1
ML+BCH(127,64,10)	0.175	256	128	0.002
ML+BCH(127,64,10)	0.2	1024	32	0.16
ML+BCH(127,64,10)	0.2	512	64	0.043
ML+BCH(127,64,10)	0.2	256	128	-
ML+BCH(255,123,19)	0.175	984	32	0.1800
ML+BCH(255,123,19)	0.175	875	36	0.1600
ML+BCH(255,123,19)	0.175	787	40	0.1500
ML+BCH(255,123,19)	0.175	716	44	0.1400
ML+BCH(255,123,19)	0.175	656	48	0.1300
ML+BCH(255,123,19)	0.175	606	52	0.1200
ML+BCH(255,123,19)	0.175	562	56	0.1100
ML+BCH(255,123,19)	0.175	525	60	0.1000
ML+BCH(255,123,19)	0.175	492	64	0.0900
ML+BCH(255,123,19)	0.175	463	68	0.0800
ML+BCH(255,123,19)	0.175	437	72	0.0700
ML+BCH(255,123,19)	0.175	414	76	0.0600
ML+BCH(255,123,19)	0.175	394	80	0.0400
ML+BCH(255,123,19)	0.175	375	84	0.0350
ML+BCH(255,123,19)	0.175	358	88	0.0320
ML+BCH(255,123,19)	0.175	342	92	0.0200
ML+BCH(255,123,19)	0.175	328	96	0.0100
ML+BCH(255,123,19)	0.175	315	100	0.0070
ML+BCH(255,123,19)	0.175	303	104	0.0040
ML+BCH(255,123,19)	0.175	292	108	0.0030

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML+BCH(255,123,19)	0.175	281	112	0.0020
ML+BCH(255,123,19)	0.175	271	116	0.0010
ML+BCH(255,123,19)	0.175	262	120	0.0005
ML+BCH(255,123,19)	0.175	254	124	0.0003
ML+BCH(255,123,19)	0.175	246	128	0.0001
ML+BCH(255,123,19)	0.2	984	32	0.16
ML+BCH(255,123,19)	0.2	492	64	0.05
ML+BCH(255,123,19)	0.2	246	128	-
ML+BCH(511,250,31)	0.175	1000	32	0.195
ML+BCH(511,250,31)	0.175	500	64	0.11
ML+BCH(511,250,31)	0.175	250	128	0.001
ML+BCH(511,250,31)	0.2	1000	32	0.1500
ML+BCH(511,250,31)	0.2	889	36	0.1400
ML+BCH(511,250,31)	0.2	800	40	0.1200
ML+BCH(511,250,31)	0.2	727	44	0.1000
ML+BCH(511,250,31)	0.2	667	48	0.0900
ML+BCH(511,250,31)	0.2	615	52	0.0800
ML+BCH(511,250,31)	0.2	571	56	0.0600
ML+BCH(511,250,31)	0.2	533	60	0.0500
ML+BCH(511,250,31)	0.2	500	64	0.0200
ML+BCH(511,250,31)	0.2	471	68	0.0150
ML+BCH(511,250,31)	0.2	444	72	0.0100
ML+BCH(511,250,31)	0.2	421	76	0.0080
ML+BCH(511,250,31)	0.2	400	80	0.0040
ML+BCH(511,250,31)	0.2	381	84	0.0020
ML+BCH(511,250,31)	0.2	364	88	0.0010
ML+BCH(511,250,31)	0.2	348	92	0.0005
ML+LDPC	0.1	512	64	0.2400
ML+LDPC	0.1	256	128	0.22
ML+LDPC	0.11	512	64	0.2000

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML+LDPC	0.11	256	128	0.17
ML+LDPC	0.12	512	64	0.1700
ML+LDPC	0.12	256	128	0.02
ML+LDPC	0.13	512	64	0.1500
ML+LDPC	0.13	256	128	0.00005
ML+LDPC	0.14	512	64	0.1000
ML+LDPC	0.15	512	64	0.0800
ML+LDPC	0.16	512	64	0.0200
ML+LDPC	0.17	512	64	0.0020
ML+LDPC	0.175	1024	32	0.1800
ML+LDPC	0.175	964	34	0.1600
ML+LDPC	0.175	910	36	0.1400
ML+LDPC	0.175	862	38	0.1300
ML+LDPC	0.175	820	40	0.1200
ML+LDPC	0.175	780	42	0.1100
ML+LDPC	0.175	745	44	0.1000
ML+LDPC	0.175	712	46	0.0900
ML+LDPC	0.175	683	48	0.0700
ML+LDPC	0.175	655	50	0.0600
ML+LDPC	0.175	630	52	0.0400
ML+LDPC	0.175	607	54	0.0200
ML+LDPC	0.175	585	56	0.0100
ML+LDPC	0.175	565	58	0.0050
ML+LDPC	0.175	546	60	0.0030
ML+LDPC	0.175	529	62	0.0010
ML+LDPC	0.175	512	64	0.0004
ML+LDPC	0.2	1024	32	0.1200
ML+LDPC	0.2	964	34	0.1100
ML+LDPC	0.2	910	36	0.0800
ML+LDPC	0.2	862	38	0.0600

METHOD	γ	# OF BITS	Pixels/bit	BER
Table B.1 continued...				
ML+LDPC	0.2	820	40	0.0300
ML+LDPC	0.2	780	42	0.0200
ML+LDPC	0.2	745	44	0.0060
ML+LDPC	0.2	712	46	0.0020
ML+LDPC	0.2	683	48	0.0010
ML+LDPC	0.2	655	50	0.0001
ML+LDPC	0.2	630	52	0.00005
ML+LDPC	0.2	512	64	0.000006

Table B.2. DCT technique simulation results

METHOD	γ	# OF BITS	Pixels/bit	BER
Correlation	0.2	1024	64	0.22
Correlation	0.2	512	128	0.15
Correlation	0.2	256	256	0.07
Covariance	0.2	1024	64	0.17
Covariance	0.2	512	128	0.11
Covariance	0.2	256	256	0.04
ML	0.2	1024	64	0.13
ML	0.2	512	128	0.066
ML	0.2	256	256	0.01
ML+BCH(63,30,6)	0.2	960	32	0.23
ML+BCH(63,30,6)	0.2	480	64	0.125
ML+BCH(63,30,6)	0.2	240	128	0.015
ML+LDPC	0.2	1024	32	0.2
ML+LDPC	0.2	512	64	0.07
ML+LDPC	0.2	256	128	0.0003

REFERENCES

1. Cox, J., M. L. Miller and J. A. Bloom, *Digital Watermarking*, Morgan Kauffman Publishers, ISBN 1-55860-714-5, 2002.

2. Piva, A., M. Barni, F. Bartolini and V.Cappellini,"Application-driven Requirements for Digital Watermarking Technology", *Proc. of EMMSEC 98*, Bordeaux, France September 28-30, 1998.

3. Jonathan, K. Su, F. Hartung and B. Girod, *Digital watermarking of Text, Image and Video Documents*, http://www-nt.e-technik.uni-erlangen.de.

4. Piva, A., M. Barni, F. Bartolini and V. Cappellini, "Capacity of the watermarking channel: How many bits can be hidden within a digital image", *Security and Watermarking of Multimedia Contents, Proc. of SPIE*, Wong, Delp, Vol. 3657, pp 437-448, San Jose, CA, 1999.

5. MacKay, D. J. C, "Good Error-Correcting Codes Based on Very Sparse Matrices", *IEEE Trans. Information Theory*, Vol. 45, pp. 399-431, March 1999.

6. Zhao, J. and E. Koch,"Embedding Robust Labels into Images for Copyright Protection", *Proc. Of Intellectual Congress on Intellectual Property Rights for Specialized Information Knowledge and New Technologies*, Vienna, Austria, , pp.242-251, Aug. 21-25, 1995.

7. Barni, M., F. Bartolini and A. Piva, "Copyright protection of digital images by means of frequency domain watermarking", *Mathematics of Data/Image Coding, Compression and Encryption, Proceedings of SPIE*, vol. 3456, pp. 25-35, July 21-22 1998.

8. Juan, R. H. M. A. and F. Perez-Gonzalez, ”DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure”, *IEEE Trans. Image Processing*, Vol. 9, pp. 55-68, January 2000.

9. Podilchuk, C. I. and W. Zeng,” Perceptual watermarking of still images”, *Electronic Proceedings of the IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, Princeton, New Jersey, June 1997.

10. Swanson, M. D., B. Zhu and A. H. Tewfik, ”Robust data hiding for images”, *7th IEEE Digital Signal Processing Workshop*, pages 37-40, 1996.

11. Swanson, M. D., B. Zhu and A. H. Tewfik, ”Transparent robust image watermarking”, *SPIE Conf. on Visual Communications and Image Proc*., volume III, pages 211-214, 1996.

12. Piva, A., M. Barni, F. Bartolini, V.Cappellini and A. De Rosa, “Improving the Robustness of Non-additive Watermarks Through Optimum Detection Theory”, *Proc. SPIE*, Vol.3971, 2000.

13. Johnson, N. and S. Kotz, Continuous Univariate Distributions, Vol. 1, Houghton Mifflin Company, Boston, 1970.

14. Birney, K.A. and T.R. Fischer,”On the modeling of DCT and sub-band image data for compression”, *IEEE Trans. Image Processing*, Vol. IP-4, pp. 186-193, Feb 1995.

15. Zinger, S., Z. Jin, H. Maitre and B. Sankur, "Optimization of Watermarking Performances Using Error Correcting Codes and Repetition", *CMS’2001: Communications and Multimedia Security*, May 2001, 229-240, Darmsadt, Germany.

16. Darbon, J., B. Sankur and H. Maitre, " Error Correcting Code Performance For Watermark Protection, Security and Watermarking of Multimedia Contents, *SPIE*,volume 4314, San Jose (CA,USA), Jan 2001.

17. Wilson, S. G., *Digital Modulation and Coding*, Prentice Hall 1996, ISBN 0-13-210071.

18. Gallager, R. G., ”Low Density Parity Check Codes”, *IRE Trans. Information Theory*, Vol. IT-8, pp. 21-28, Jan 1962.

REFERENCES NOT CITED

Balado F., J. R. Hernandez and F. Perez, Approaching the capacity limit in image watermarking, *Signal Processing* 81 (2001), p. 1215-1238.

Barni, M., *et al*., "A DCT-Domain System for Robust Image Watermarking", *Signal Processing*, Vol.66, No.3, May 1998, pp.357-372.

Chiou-Ting, H. and J. Wu,"Hidden signatures in images*", IEEE Signal Processing Society 1996 International Conference on Image Processing (ICIP'96)*, volume III, pages 743-746, Lausanne, Switzerland, September 1996.

Cox, I. J., J. Kiliani, F. T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. Image Processing*, Vol. 6, pp. 1673-1685, June 1997.

Edmund, Y. L., "A Mathematical Analysis of the DCT Coefficient Distributions for Images", *IEEE Trans. Image Processing*, Vol. 9, pp.1661-1666.

Hernandez, J. R. and F. Perez-Gonzalez, "Statistical Analysis of Watermarking Schemes for Copyright Protection of Images", *Proc. IEEE*, July 1999.

Ingemar, J. C., J. Killian, T. Leighton and T. Shamoon, "A secure, robust watermark for multimedia", *Workshop on Information Hiding*. Newton Institute, University of Cambridge, May 1996.

Ingemar, J. C. and M. L. Miller, "A review of watermarking and the importance of perceptual modeling", *Proc. of Electronic Imaging '97*, February 1997.

Jajodia, S. and N.F.Johnson, "Exploring Steganography: Seeing the Unseen", *IEEE Computer*, Vol.31, No.2,pp.26-34, feb.1998.

Knopp, R. and A. Robert, ”Detection Theory and Digital Watermarking ”, *Proc. SPIE*, Vol. 3971, pp 14-23 2000.

Koch, E., J. Rindfrey and J. Zhao,”Copyright Protection for Multimedia Data”, *Digital Media and Electronic Publishing*, Academic Press, London, 1996, pp. 203-213.

Köprülü, F. İ., *Application of LDPC Codes To Watermarking Channels*, MSc Thesis, Boğaziçi University Library, 2001.

Piva, A. *et. al*., "DCT Based Watermark Recovery Without Resorting to the Uncorrupted Original Signal", *Proc. IEEE International Conf. on Image Processing*, ICIP-97, Vol.1, pp.520-523.

Shannon, C.E., ”A Mathematical Theory of Communication”, *Bell Syst. Tech. J*., Vol. 27, pp.379423, 623656, 1948.

Sklar, B., *Digital Communications: Fundamentals ans Applications*, Prentice Hall 2001, ISBN 0-13-084788-7.

Swanson, M. D., *et al*., "Multimedia data Embedding and Watermarking Technologies", *Proc. of the IEEE*, Vol.86, No.6, June 1998, pp.1064-1087.

Trees V. H. L., *Detection, Estimation and Modulation Theory, Part 1*, Wiley,New York, 1968.