Informationstechnik

Muhammad Kashif Samee

Robust Watermarking and its Applications to Communication Problems

2012

Robust Watermarking and its Applications to Communication Problems

Von der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität Dortmund
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
eingereicht von

Muhammad Kashif Samee

Tag der mündlichen Prüfung:	17.04.2012
Hauptreferent:	Univ.-Prof. Dr.-Ing. J. Götze
Korreferent:	Univ.-Prof. Dr.-Ing. C. Wietfeld

Arbeitsgebiet Datentechnik, Technische Universität Dortmund

Abstract

Digital watermarking has recently gained an intense interest in research and applications. An invisible and secret signal, called watermark, is added to the host data. With the help of this watermark issuer of the data can be unveiled, unauthorised users can be identified, illicit copying can be avoided, any attempt to temper with the data can be detected and many other security services can be provided. In this thesis, the relations and differences between watermarking and communication systems are elaborated. Based on these results new methods for both watermarking and communication are derived.

A new blind, robust and reversible watermarking scheme based on Code Division Multiple Access (CDMA) is presented in this thesis. Using this scheme watermark is arithmetically added to spatial domain or frequency domain. Watermark is extracted by using spreading codes only. Proposed watermarking scheme is simple, computationally efficient and can be applied to any image format.

A novel idea that watermark can be part of the image is presented. By using watermark, which is a part of an image, digital watermarking can be used beyond simple security tasks. A part of an image is selected and embedded in the whole image as watermark. This watermarked image is attacked (transmitted or compressed). By using the extracted watermark and attacked selected part image quality can be assessed or jpeg quantization ratio can be estimated or even image can be equalized blindly.

Furthermore, CDMA based watermarking is used to authenticate radio frequency signal. Spreaded watermark is added in the form of noise to the modulated radio frequency signal. If this noise is increased, watermarked signal automatically becomes a scrambled signal. Later watermark is extracted and by using reversibility of proposed scheme watermark is removed. Once the watermarked is removed original signal is restored, hence descrambled.

Acknowledgements

This thesis was written while I was working as a Ph.D. student at the Information Processing Laboratory of the Dortmund University of Technology. I would like to thank Professor Jürgen Götze, the head of the laboratory, for all his support, supervision and guidance. I thank him for his constructive suggestions and ideas because of which my work improved a lot. I would also like to thank him for creating an open and relaxed atmosphere, and for providing excellent working conditions.

Furthermore, I am very pleased to thank Professor Christian Wietfeld (Communication Networks Institute) for his interest in my work, his comments and his time. I am also grateful to Higher Education Commission (HEC) of Pakistan for funding my Ph.D. I am grateful to all my present and former colleagues for providing such a stimulating atmosphere at the laboratory. It was a pleasure to share so much time with you. I would especially like to thank Jan Geldmacher for all his support and help during my stay in Germany.

I would like to thank my wife for her love, patience, constant support and encouragements during the long years of my Ph.D. Finally, I would like to thank my parents, whatever I am and whatever I did is just because of them. This thesis is dedicated to my father.

Contents

List of Figures

List of Tables

The ink (used for writing) of the scholar is more sacred than the blood of the martyr.

Prophet Muhammad (peace be upon him)

1 Introduction

Cryptography (information hiding) is a very old technique used by man kind during wars as well as in normal circumstances. Cryptography books are filled with examples of such methods. Old Greek messengers tattooed messages to their shaved heads. These messages were concealed when the hairs grew back. Messages were read after shaving the head again. Wax tables were scraped down to bare wood and the messages were scratched there. When the tables were re-waxed messages became hidden [1]. As the time passed these cryptographic techniques improved in all aspects i.e. speed, capacity and security.

With the rapid development of the world wide web, emergence of the broadband networks and the cheap digital devices the usage of the multimedia data increased drastically during the last decade. Digital (multimedia) data is very easy to copy and it is often copied without considering the copyright. Because of this the distributors or the owners of the multimedia data forced to use such schemes which prevent digital right violations automatically. Digital watermarking is one of the well known techniques serving this cause. Conventional cryptography systems only allow valid key holder, to use the encrypted data. Once the data is decrypted it is vulnerable to manipulation, reproduction and retransmission. Therefore, conventional cryptography systems provide less data privacy, which a publisher requires to confront unauthorized reproduction. On the other hand, digital watermarking is a technique which permanently adds security information to host data. This information remains there even after decryption process[1]. Digital watermarking has recently become a very active area of research. Properties of a good watermarking algorithm depend on its application. However, general properties of a watermarking algorithm as described in [2] are as follows:

[1] an exception is reversible watermarking

- **Unobtrusive:** Watermark should be perceptually invisible, it should not alter host data significantly.

- **Robust:** Watermark should withstand intentional as well as unintentional attacks. It must be difficult (ideally impossible) to remove the watermark with no or partial knowledge.

- **Universal:** Same watermarking scheme should apply to all three media (image, audio and video). This is potentially helpful in the watermarking of multimedia products.

- **Unambiguous:** Retrieval of watermark should unambiguously identify the owner.

1.1 Application of Watermarking Schemes

Watermarking is an enabling technology for a number of applications [3, 4, 5]. Various possible watermarking applications are [6]:

- **Authentication and tamper-proofing:** This is the most common application of digital watermarking. Watermark is used to verify the authenticity and integrity of digital items. If the extracted watermark is incorrect, it can be concluded that the original data is tempered. Certain watermarking algorithms used for authentication can provide tampered region localization, e.g. can identify the tempered region of modified image.

- **Owner identification and proof of ownership:** Watermark carry information about legal owner or distributor or any copyright holder of digital item. Extracted watermark can be used for notifying a user that the item is copyrighted, for tracking illicit copies of the item, or can be used to proving ownership in case of a legal dispute.

- **Broadcast monitoring:** Watermark is used for various functions that are related to digital media broadcasting. Embedded information can be used to verify that the broadcasting of com-

mercials took place as scheduled. It can be used for automated royalty collection schemes, or can be used for audience metering (number of users who watched or listen to a certain broadcast). Broadcast monitoring is usually performed by automated monitoring stations.

- **Transaction tracking:** In this application, a unique watermark is embedded in each copy of a digital item that is distributed. Watermark is not only used to carry information about owner but also used to mark the specific transaction copy. This unique embedded information can be used for identification of entities that illegally distributed the digital item or did not adopt required measures for copying. For example, every copy of a movie distributed to various theaters is watermarked with unique information. Later, if a camera recording of that movie is available in the market, with the help of a unique watermark, it can be identified from which theater the movie is recorded.

- **Usage control:** Watermarking can play active role in controlling the terms of use of the digital content. Watermark can be used in conjunction with appropriate compliant devices to prevent unauthorized recording of digital items (copy control) or playback of unauthorized copies (playback control). [7] showed that digital watermarking can be used for DVD copy and playback control by content scrambling.

- **Persistent item identification:** Digital watermarking is used for associating an identifier with a digital item. This identifier is used in conjunction with appropriate databases to convey various information about the digital item. This information can provide certain benefit to user, e.g. user can get access to free services and products, thus, discouraging user from removing the watermark. If this watermark is removed or digital item is illegally distributed user will not get added value provided by the watermark.

- **Enhancement of legacy systems:** Embedded data can be used for enhancement of functionality or information provided by legacy systems while ensuring backwards compatibility. For example, watermark capable of enabling stereoscopic viewing to stereo-enabled receivers can be embedded in conventional digital

TV broadcast. Conventional TV receiver would continue to receive the conventional signal with non perceptible degradations.

1.2 Classification of Watermarking Schemes

On the basis of robustness watermarking schemes are classified in three main categories [6]:

- **Robust:** Watermarking schemes are designed in such a way that watermark resist host signal manipulations (called attacks) and are usually used for Intellectual Property Rights (IPR) protection applications. Obviously, no single watermarking scheme withstands all type of modifications. So, robustness refers to a subset of all possible attacks and up to certain degree of host signal degradation.

- **Fragile:** In fragile watermarking schemes, watermarks are designed to be vulnerable to all attacks, i.e., they become undetectable by even the slight host data modification.

- **Semi-fragile:** This class of watermarking schemes provides selective robustness to a certain set of attacks which are considered allowable, while vulnerable to others. Practically, all robust watermarks are actually semi-fragile, but the selective robustness is not a requirement imposed by system designer but something that can not be avoided.

On the basis of watermark detection watermarking schemes can be categorized into two main classes:

- **Non-oblivious:** In non-oblivious watermarking schemes original data (image) is required during the watermark extraction process. The general approach of non-oblivious watermarking schemes is usually a comparison based algorithm, i.e. the watermark is embedded by altering some coefficients and extracted by comparison between watermarked and original image [8, 9].

- **Oblivious:** In oblivious/blind watermarking original data (image) is not required during the watermark extraction process. In oblivious (blind) watermarking schemes more sophisticated algorithms are used to extract the watermark without using the original image.

1.3 Watermarking and Communication

Digital watermarking is a process in which a signal is embedded in another signal. It can also be described as imperceptible information transmitted through a side channel [10]. Watermarking system is very similar to a communication system. Therefore, it is worthwhile to elaborate the relations and differences between watermarking and communication systems. Based on these results new methods for both watermarking and communication are derived. In the following this comparison is mainly done for CDMA based watermarking and CDMA based communication systems.

1.3.1 Simple Spread Spectrum Watermarking and Communication

[11] defines spread spectrum communication as follows:

> "Spread spectrum is a mean of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by a code which is independent of the data, and synchronized reception with the code at the receiver is used for despreading and subsequent data recovery."

In spread spectrum communication, data is spread with a spreading code before transmission and in spread spectrum watermarking, the watermark is spread before insertion into the cover image. The insertion and later extraction of the watermark from the image may disturb the watermark as the data are disturbed during the transmission.

Unintentional attacks (such as image processing operations, compression etc.) on the watermarked image can be considered as added noise during transmission in spread spectrum communications. Malicious attacks (threats to security) are similar to both. The comparison between spread spectrum communication and spread spectrum watermarking is shown in Fig. 1.1.

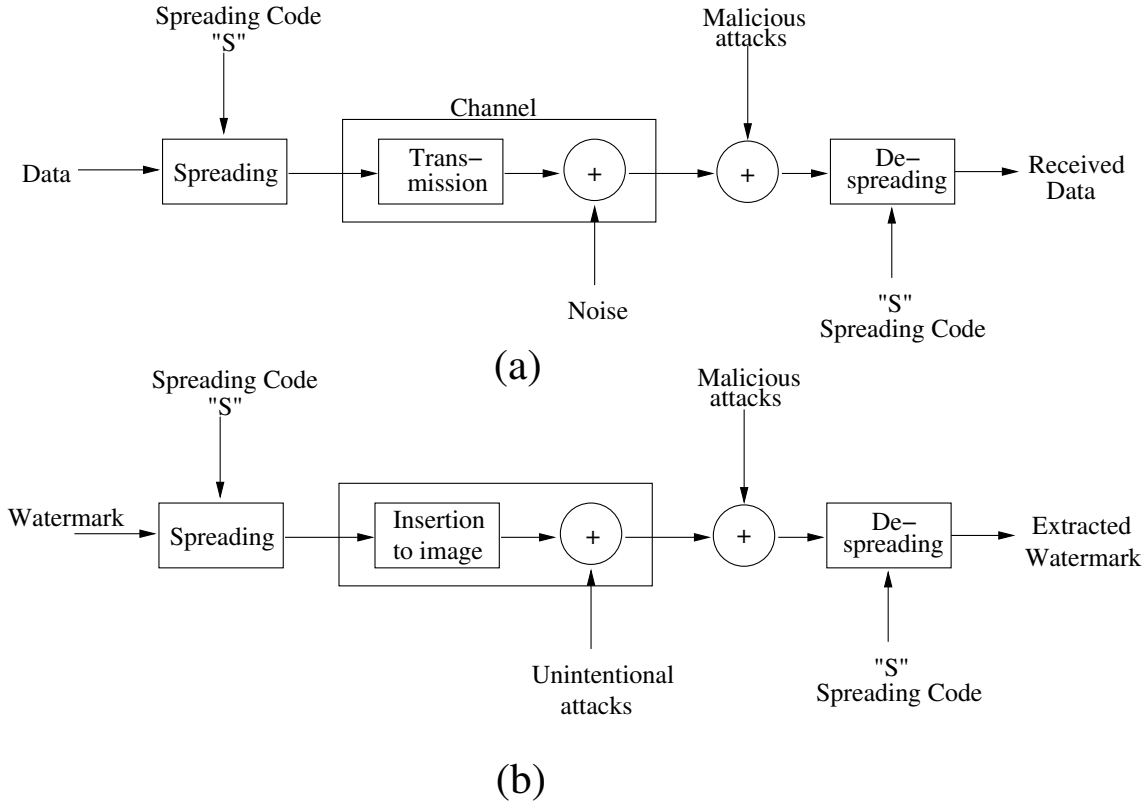


Figure 1.1: *(a)Simple spread spectrum communication. (b)Simple spread spectrum watermarking.*

1.3.2 DS-CDMA Based Spread Spectrum Watermarking and Communication

In DS-CDMA spread spectrum communication, multiple orthogonal spreading codes are used to serve multiple users. By doing so, every user can have the advantage of using the full bandwidth. As the image has limited pixels it can be considered as providing the limited bandwidth to carry data (watermark). Multiple small watermarks can be

used or a large watermark can be divided into parts. Every watermark (or part) is spread with different mutually orthogonal spreading codes, and inserted simultaneously into the cover image (Fig. 1.2). Every part is spread over the whole image and therefore, the length of every spreading code can be increased. Hence, one obtains more security. If k orthogonal spreading codes are used instead of one, each spreading code can be k-times longer and the eavesdropper has to break k codes.

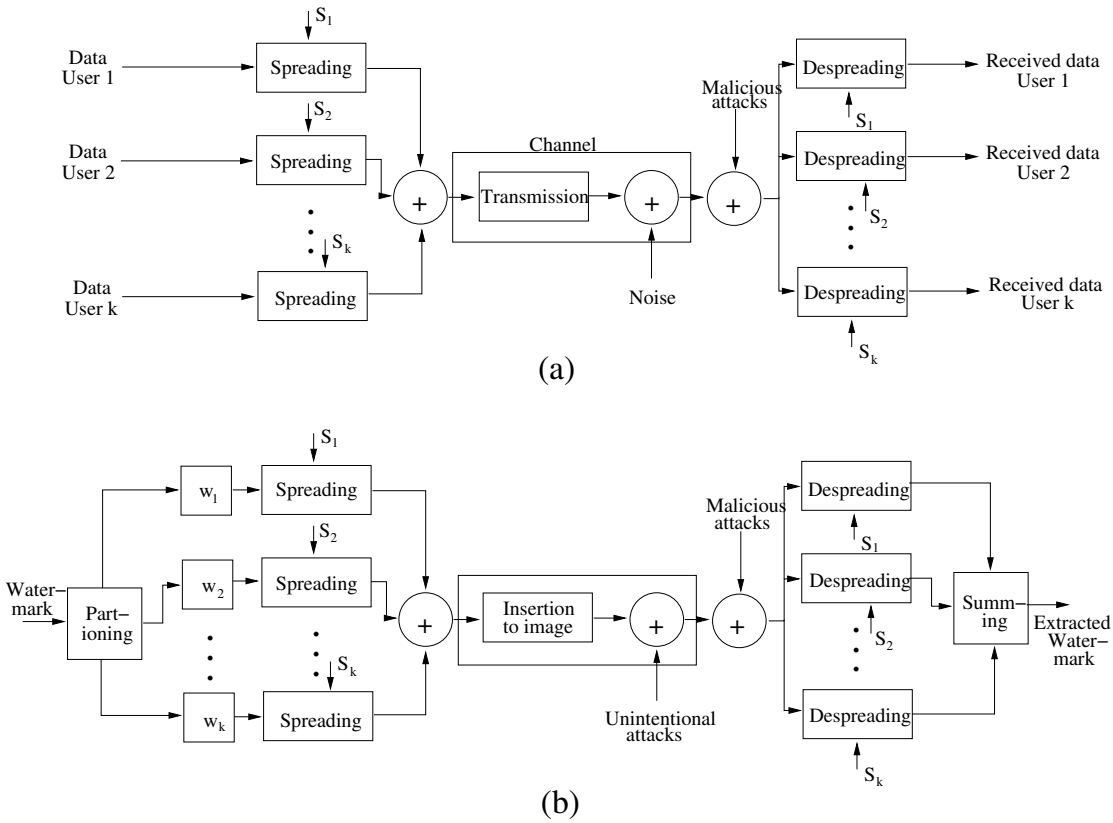


Figure 1.2: *(a)DS-CDMA based spread spectrum communication. (b)DS-CDMA based spread spectrum watermarking.*

1.4 Overview and Contribution

1.4.1 Overview

In chapter 2, a blind, robust and secure watermarking technique based on DS-CDMA is presented. By using this technique a large watermark can be embedded in an image and it can be extracted at very high PSNR value. Multiple spreading codes are used to make this scheme secure and more robust against a variety of attacks. In practice, very few watermarking schemes use multiple spreading codes [12, 13]. Although most of the schemes use the name CDMA, but they actually used only one spreading code. In proposed scheme, spreaded watermark is arithmetically added to spatial domain or frequency domain (both DCT or DWT). A new approach is used to select the coefficients in the DCT or DWT domain, where the watermark is to hide, which significantly reduces BER in extracted watermark [14].

In chapter 3, the effect of Error Correcting Codes (ECC) on an oblivious CDMA based watermarking scheme is studied. Obviously, ECC is employed to increase the robustness of watermarking algorithms [15, 16], but in order to accommodate the additional bits of the ECC, the strength of watermark signal has to be reduced to keep the PSNR value constant. Therefore, introducing ECC and decreasing the strength of the watermark signal contradict each other concerning robustness. In this chapter, it is studied whether it is feasible to apply computationally complex ECC at the cost of watermark strength for CDMA based watermarking schemes. Furthermore, a new technique called "by parts interleaving" is introduced. Spreaded watermark bits are in parts interleaved in different regions of the image to make the algorithm robust against geometric attacks [17].

The biggest disadvantage of general watermarking schemes is that they permanently distort original data. Reversible watermarking techniques allow the restoration of original data, after watermark detection [18, 19]. Watermark is arithmetically added in the proposed CDMA based watermarking scheme. Therefore, it is not difficult to remove it after detection. In proposed algorithm, watermark is extracted by using spreading codes only. After extraction, the watermark can be removed

from watermarked data only by using same spreading codes. Furthermore, the original watermark is not required during watermark removal process [20, 21].

In chapter 4, two algorithms based on digital watermarking are proposed:

1. Reduced reference image quality assessment for JPEG distortion.

2. Blind quantization ratio estimation for JPEG images.

Reduced Reference Image Quality Assessment for JPEG Distortion

Objective Image Quality Assessment (IQA) aims to automatically measure the quality degradation perceived by human eyes. A new reduced-reference image quality assessment algorithm for JPEG distortion (distortion specific metric) is proposed. This algorithm uses an 8×8 block of the original image (as a reduced-reference) and embeds this block as a watermark. To assess the quality of the image this block is extracted from the watermarked distorted image and is compared with the corresponding block from the same image. Proposed scheme results in the quality factor with which an image was originally compressed, hence gives a direct measure for quality [22].

Blind Quantization Ratio Estimation for JPEG Images

Most image acquisition and editing tools use the JPEG standard for image compression. A simple quantization ratio estimation scheme based on digital watermarking is proposed. An 8×8 pixel block is selected from the original image and inserted in the whole image using digital watermarking. If the receiver does not know the quantization ratio (quality factor) with which image is compressed, it can apply a set of all possible quantization ratios to the received compressed image. The inserted watermark can be extracted correctly only with the same quantization ratio which was used to compress the image. Hence image

can be decompressed (reconstructed) without the prior knowledge of quantization ratio [24].

In chapter 5, three watermarking based equalization algorithms are presented, for three different scenarios:

- **Algorithm I:** when channel specific training sequence or specific watermark is required,
- **Algorithm II:** when watermark/training sequence can be any sequence (can be a part of data),
- **Algorithm III:** when original data is required after equalization.

Algorithm I

Equalization is a technique used to cope Intersymbol Interference (ISI) [25]. A blind equalization method for images based on digital watermarking is presented. This method uses the watermark of a transmitted watermarked signal as a reference training sequence. DS-CDMA based spread spectrum watermarking scheme is used here. In this method, watermark is sent through the channel along with the watermarked image. On receiving side, watermark is extracted from the watermarked image and by using received and extracted watermark channel is estimated. Therefore, the receiver does not require to know the training sequence in advance. Afterwards, received data (watermarked image) is equalized by using Normalized LMS algorithm. Simulations have proven that this scheme is very effective in correcting errors from received watermarked images[26].

Algorithm II

In algorithm II, a chunk of data is selected from the data to be transmitted. This chunk of data is hidden in the entire data using digital watermarking. For watermarking, DS-CDMA based spread spectrum watermarking scheme is used. On receiving side, watermark is extracted

from the watermarked data. With the help of received chunk of data and the extracted watermark, which is actually extracted version of the selected chunk of data, the channel is equalized by using Normalized LMS algorithm. In this method, neither receiver requires to know the training sequence in advance nor the sender requires to send training sequence. Proposed algorithm can simultaneously be used for usual watermarking applications and blind equalization [27, 28].

Algorithm III

A blind equalization method based on reversible watermarking is presented. Core of algorithm III is same as algorithm II. In algorithm III, however, reversible watermarking is used. Therefore, after equalization, watermark can be removed from the corrected watermarked data to recover the original data [29].

In chapter 6, a watermarking scheme for physical layer authentication of radio frequency signals is presented. This scheme is blind, robust and reversible and it is based on CDMA. In proposed algorithm, spreaded watermark is arithmetically added to the modulated signal before transmission. At the receiver watermark is extracted before demodulation, by using spreading codes only. After extraction, the watermark can be removed from watermarked data only using the same spreading codes. High intensity watermark can also serve as scrambler [30].

1.4.2 Major Contributions

Major contributions of this thesis are:

- A novel blind, robust and secure digital watermarking scheme based on CDMA is proposed.
- Proposed scheme is transformed to a reversible watermarking scheme.
- Effect of Error Correcting Codes on CDMA based watermarking

is discussed.

- A new idea that the watermark can be a part of the original data (self reference watermarking) is proposed, and the advantages of this scheme are elaborated.

- It is shown that digital watermarking can be used for image quality assessment.

- Digital watermarking is used for implementing an algorithm for blind quantization factor estimation of JPEG compressed images.

- With the help of digital watermarking blind equalization algorithms for images are developed.

- Authentication of radio frequency signals is done using digital watermarking and it is shown how high intensity watermarking serves as scrambling.

2 CDMA Based Watermarking

2.1 Introduction

With the emergence of wide bandwidth wireless networks, mobile Internet is set to provide a significant channel of multimedia content distribution. In the absence of proper digital rights management systems, there is a real risk of interception, manipulation, misuse and unauthorized distribution of information. Digital watermarking is one of those techniques used for copy rights protection of multimedia data. Digital watermarking has become a very active area of research in recent years.

An important quality of a good watermarking scheme is security, i.e. how robust a watermarking scheme is against malicious attacks. A watermarking algorithm similar to spread spectrum communication systems fulfills this need, because hidden information in spread spectrum communications is mistaken for noise, such that it goes undetected by an evedropper. Information is spread by a spreading code in spread spectrum systems and in Direct Sequence Code Division Multiple Access (DS-CDMA) systems multiple orthogonal spreading codes are used. Code Division Multiple Access (CDMA) systems are considered as one of the most secure communication systems. The proposed DS-CDMA based algorithm uses multiple spreading codes and can carry more payload (large watermark) than any other simple spread spectrum based algorithm [31, 32, 33, 34, 35, 36] (although some of these methods are also named CDMA-based, the term CDMA is somewhat misleading, since all these methods actually use one single spreading code). A few schemes like [12, 13] used CDMA in original sense. [12] implemented watermarking in spatial domain and [13] used wavelet domain for watermark insertion. In proposed algorithm spatial domain or frequency domain (DCT or DWT) can be used for watermark hiding. In case of frequency domain, every bit is hidden in as similar as possible frequency

coefficients, which significantly improves Bit Error Rate (BER).

A very important topic of DS-CDMA based watermarking is the selection of the spreading codes. The algorithm, which is proposed in this chapter, is based on simple "zero mean code", unlike PN code patterns [31], Gold sequences [32], and decimal sequences [33]. As already mentioned above, the "Multiple Access" property of CDMA is emphasized. Multiple orthogonal zero mean codes are used to improve security and robustness against a number of attacks. The watermark is permuted before insertion to the image. Watermark can be added to spatial domain as well as frequency domain (both DCT and Wavelet). In case of DCT, watermarking is done by altering middle frequency coefficients to have a trade off between perceptual transparency and robustness against compression [37]. Experimental results have shown that with an intelligent selection of frequency coefficients (both in DCT and Wavelet domain), BER improves a lot.

2.2 CDMA Based Digital Watermarking

2.2.1 Watermark as a Vector

First of all the watermark should be converted into a sequence of bits. By doing so it can be treated in the same way as a stream of bits is treated in communication systems. Therefore, the watermark, which is usually a binary image represented as a two-dimensional matrix, is converted into a vector by arranging the rows of the matrix sequentially into a long vector (Fig. 2.1).

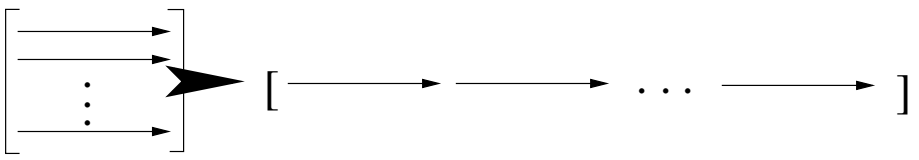


Figure 2.1: *Watermark conversion from matrix to vector.*

2.2.2 Zero Mean Code

The spreading codes $\boldsymbol{s_i} = [s_1, s_2, \ldots, s_n]$ should follow the following two conditions:

$$s_i \quad \epsilon \quad \{-1, 1\} \tag{2.1}$$

$$\sum_{i=1}^{n} s_i \quad = \quad 0 \tag{2.2}$$

(2.1) can be exempted and floating point numbers can be used to improve security. But in this case complexity increases, as in watermark insertion and extraction multiplications are required. On the other hand, if (2.1) and (2.2) both hold, only additions are required for watermark insertion and extraction. For multiple spreading codes, codes should be orthogonal and their cross correlation or inner product should be zero.

$$< \boldsymbol{s_i}, \boldsymbol{s_j} >= \boldsymbol{s_i} \cdot \boldsymbol{s_j^T} = 0 \quad if\ i \neq j \tag{2.3}$$

2.2.3 Spreading

Before spreading, the zeros of binary watermark $\boldsymbol{w} = [b_1, b_2, \ldots, b_m]$ are replaced by minus ones while the ones remain untouched, such that $b_i \epsilon \{-1, 1\}$. This is called antipodal bits in communication systems. Let $\boldsymbol{i_j}$ be the vector formed in spatial, DCT or Wavelet domain (formation of $\boldsymbol{i_j}$ vectors are discussed in Sec. 2.3). $\boldsymbol{i_j}$ is the vector in which a data bit is inserted. The length of $\boldsymbol{i_j}$ and $\boldsymbol{s_i}$ must be same. Then, $\boldsymbol{i'_j}$ is the vector of modified coefficients containing a spreaded data bit:

$$\boldsymbol{i'_j} = \boldsymbol{i_j} + \alpha b_1 \boldsymbol{s_1} \tag{2.4}$$

where α is the gain factor.

In case of multiple orthogonal codes, $\boldsymbol{i'_j}$ can contain k spreaded data bits:

$$\boldsymbol{i'_j} = \boldsymbol{i_j} + \alpha[b_1 \boldsymbol{s_1} + b_2 \boldsymbol{s_2} + \ldots + b_k \boldsymbol{s_k}] \tag{2.5}$$

2.2.4 Despreading (Extraction of Data)

The data bits can be extracted by computing the cross correlation with the respective spreading codes. In case of one spreading code:

$$\begin{aligned} < \boldsymbol{i'_j}, \boldsymbol{s_1} > &= \boldsymbol{i'_j} \cdot \boldsymbol{s_1^T} \\ &= \boldsymbol{i_j} \cdot \boldsymbol{s_1^T} + \alpha b_1 \boldsymbol{s_1} \cdot \boldsymbol{s_1^T} \end{aligned}$$

Since $(\boldsymbol{s_1} \cdot \boldsymbol{s_1^T})$ is always positive and "α" is also a positive quantity, the sign of $(\alpha b_1 \boldsymbol{s_1} \cdot \boldsymbol{s_1^T})$ is determined by b_1. If the magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_1^T})$ is less than $(\alpha \boldsymbol{s_1} \cdot \boldsymbol{s_1^T})$ the sign of $(\boldsymbol{i'_j} \cdot \boldsymbol{s_1^T})$ is determined by b_1. Therefore,

$$b_1 = sign < \boldsymbol{i'_j}, \boldsymbol{s_1} > \quad if\ |\boldsymbol{i_j} \cdot \boldsymbol{s_1^T}| < |\alpha \boldsymbol{s_1} \cdot \boldsymbol{s_1^T}| \tag{2.6}$$

If the condition in (2.6) is not fulfilled a bit error may occur. This is just the implementation of the simple "Matched Filter".

In case of multiple orthogonal spreading codes the bit b_i is extracted by using spreading code $\boldsymbol{s_i}$:

$$\begin{aligned} < \boldsymbol{i'_j}, \boldsymbol{s_i} > &= \boldsymbol{i'_j} \cdot \boldsymbol{s_i^T} \\ &= \boldsymbol{i_j} \cdot \boldsymbol{s_i^T} + \alpha[b_1 \boldsymbol{s_1} \cdot \boldsymbol{s_i^T} + \ldots + b_k \boldsymbol{s_k} \cdot \boldsymbol{s_i^T}] \end{aligned}$$

By using (2.3) one obtains:

$$< \boldsymbol{i'_j}, \boldsymbol{s_i} >= \boldsymbol{i_j} \cdot \boldsymbol{s_i^T} + \alpha b_i \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}$$

Again b_i is determined as follows:

$$b_i = sign < \boldsymbol{i'_j}, \boldsymbol{s_i} > \quad if\ |\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}| < |\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}| \tag{2.7}$$

So, all the bits hidden in $\boldsymbol{i'_j}$ can be extracted by their respective spreading codes. Again, if the condition in (2.7) is not fulfilled a bit error may occur.

2.3 Formation of Vectors i_j

2.3.1 Spatial Domain

$\boldsymbol{i_j}$ vectors should be formed in such a way that magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_i^T})$ remains as small as possible. The magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_i^T})$ will be small, if spreading codes are long enough. As number of ones and minus ones are equally distributed in spreading codes $\boldsymbol{s_i}$, according to probability theory, the magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_i^T})$ will be small if the spreading codes are long. This holds even if the elements of $\boldsymbol{i_j}$ are just random numbers. Simulations have shown that 256 bit long spreading code is enough to detect the bits correctly. In case of spatial domain watermarking, pixel values can be considered just random values. $\boldsymbol{i_j}$ vectors can be formed any where, simple formation of $\boldsymbol{i_j}$ vectors are row-wise or column-wise. In case of row-wise $\boldsymbol{i_j}$ vectors, elements of same spatial row form an $\boldsymbol{i_j}$ vector.

2.3.2 DCT Domain

Selection of Frequency Coefficients

Different possibilities for the selection of the frequency coefficients into which the watermark is inserted are presented in [2, 8, 9, 12]. It is suggested that the watermark should be inserted in perceptually more significant areas, i.e. in low frequency coefficients. On the other hand it is also suggested that watermark should be inserted in middle frequency coefficients, because by altering low frequency coefficients the quality of the watermarked image is more affected. Everyone agrees, however, that higher frequency coefficients should not be used for watermark embedding, as high frequency coefficients are usually discarded in the compression process.

In the presented method there is another condition:

$$|\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}| < |\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}| \tag{2.8}$$

If this condition is violated a bit error is expected. Considering (2.2), the magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_i^T})$ is close to zero, if the elements of $\boldsymbol{i_j}$ are similar. So, $\boldsymbol{i_j}$ should be formed in such a way that the elements of $\boldsymbol{i_j}$ are as similar as possible. If 8×8 blocked DCT is applied to the cover image, the elements in the lower frequency area vary a lot more than the elements in middle or higher frequency area. But higher frequency coefficients should not be used, because of the above mentioned reasons. Therefore, the middle frequency coefficients are chosen. By selecting the middle frequency coefficients we have better perceptual transparency, the algorithm is robust against compression, and above all condition (2.8) is fulfilled with high likelihood.

$\boldsymbol{i_j}$ Vectors

The vectors $\boldsymbol{i_j}$ are formed in such a way that similar coefficients from different transformed blocks lay in the same vector. Selecting the same middle frequency rows from different blocks can be a good choice to from $\boldsymbol{i_j}$ vectors (Fig. 2.2). Selecting the same columns can be another possibility. However, a better similarity is obtained if the $\boldsymbol{i_j}$ vectors are formed by selecting middle frequency coefficients along the diagonal (Fig. 2.3) or by selecting these middle frequency coefficients with some sophisticated algorithm like in [8] (Fig. 2.4).

2.3.3 DWT Domain

Figures 2.5(a) and 2.5(b) shows how frequency coefficients are arranged in single layer wavelet domain. Here:

cA : Approximation coefficients matrix
cH : Horizontal details coefficients matrix
cV : Vertical details coefficients matrix
cD : Diagonal details coefficients matrix

Approximation coefficients (cA) can be further transformed to get layer two approximation coefficients cA2 and horizontal, vertical and diagonal details coefficients cH2, cV2 and cD2 respectively (Figures 2.5(c) and 2.5(d)).

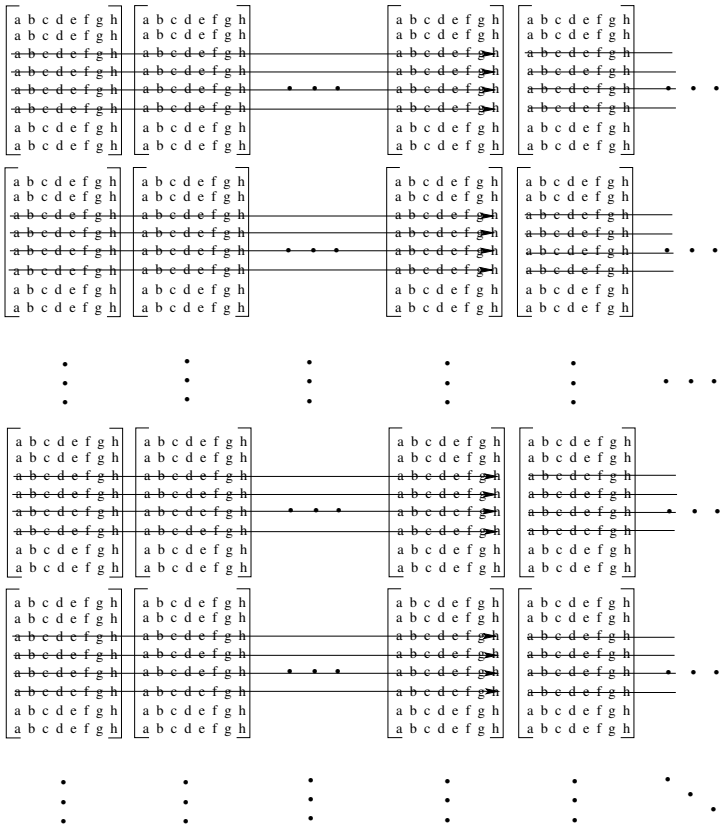


Figure 2.2: *Row-wise formation of $\boldsymbol{i_j}$ vectors. Every matrix represents a* 8×8 *transformed DCT block. Every matrix represents a* 8×8 *transformed DCT block.*

As discussed in previous section, the magnitude of $(\boldsymbol{i_j} \cdot \boldsymbol{s_i^T})$ can be minimized if the elements of $\boldsymbol{i_j}$ vectors are mutually similar. Mutually similar elements can be obtained if $\boldsymbol{i_j}$ vectors are formed in rows of same coefficient matrix. In case of single layer wavelet transformation, $\boldsymbol{i_j}$ can be formed in the rows of cA, cH, cV or cD (as shown in figures 2.6(a) and 2.6(b)). In case of 2 layer wavelet transformation different formations of $\boldsymbol{i_j}$ are shown in figures 2.6(c) and 2.6(d). Watermark can be added to different coefficient matrices simultaneously but elements of each $\boldsymbol{i_j}$ vector should be in the same coefficient matrix (as shown in figures 2.6(e) and 2.6(f)).

2.4 Watermarking Algorithm

Fig. 2.7 shows how watermark is inserted in an image. Every watermark bit is spread using a zero mean spreading code before insertion. Orig-

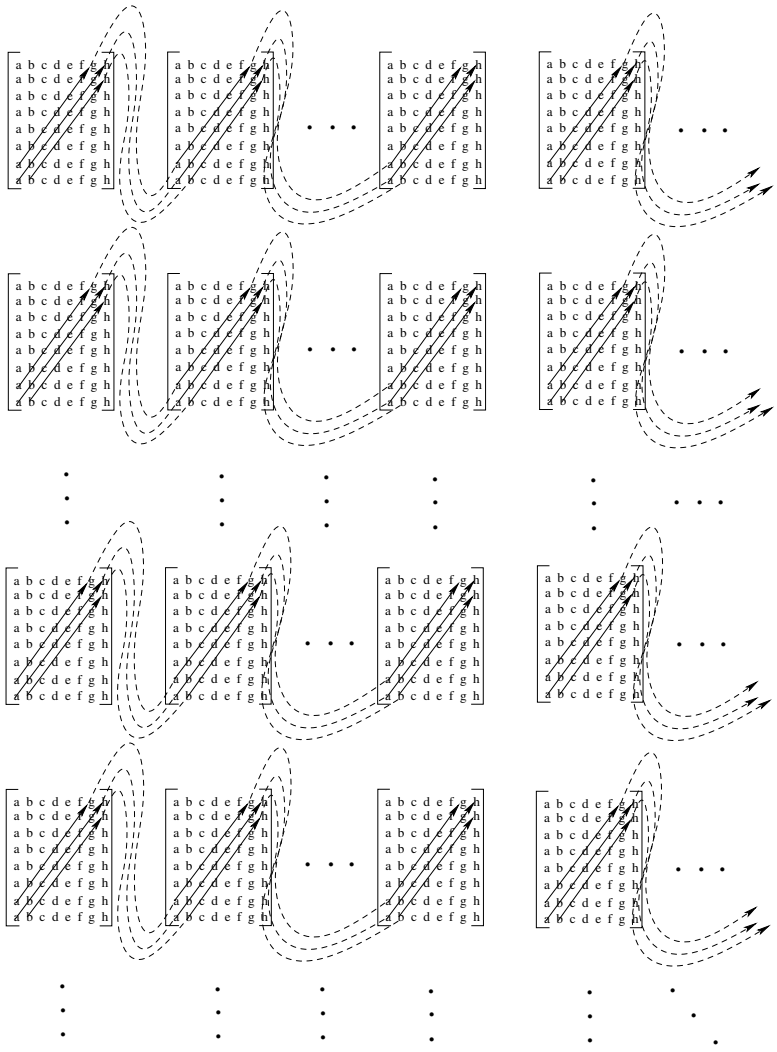


Figure 2.3: *Diagonal-wise formation of* $\boldsymbol{i_j}$ *vectors. Every matrix represents a* 8×8 *transformed DCT block. Every matrix represents a* 8×8 *transformed DCT block.*

inal image is transformed to frequency domain. Spreaded watermark is arithmetically added to frequency coefficients in Wavelet domain or DCT domain. In case of spatial domain watermarking, watermark is added to pixel values. After the addition of spreaded watermark bits the modified frequency coefficients are transformed back to get watermarked image. Fig. 2.8 explains the extraction process. Watermarked (attacked) image is transformed to frequency domain and same coefficients are selected. Just by calculating cross correlation between selected coefficients and spreading codes watermark is extracted. So, watermark is extracted blindly only by using spreading codes (which can be provided in secret key).

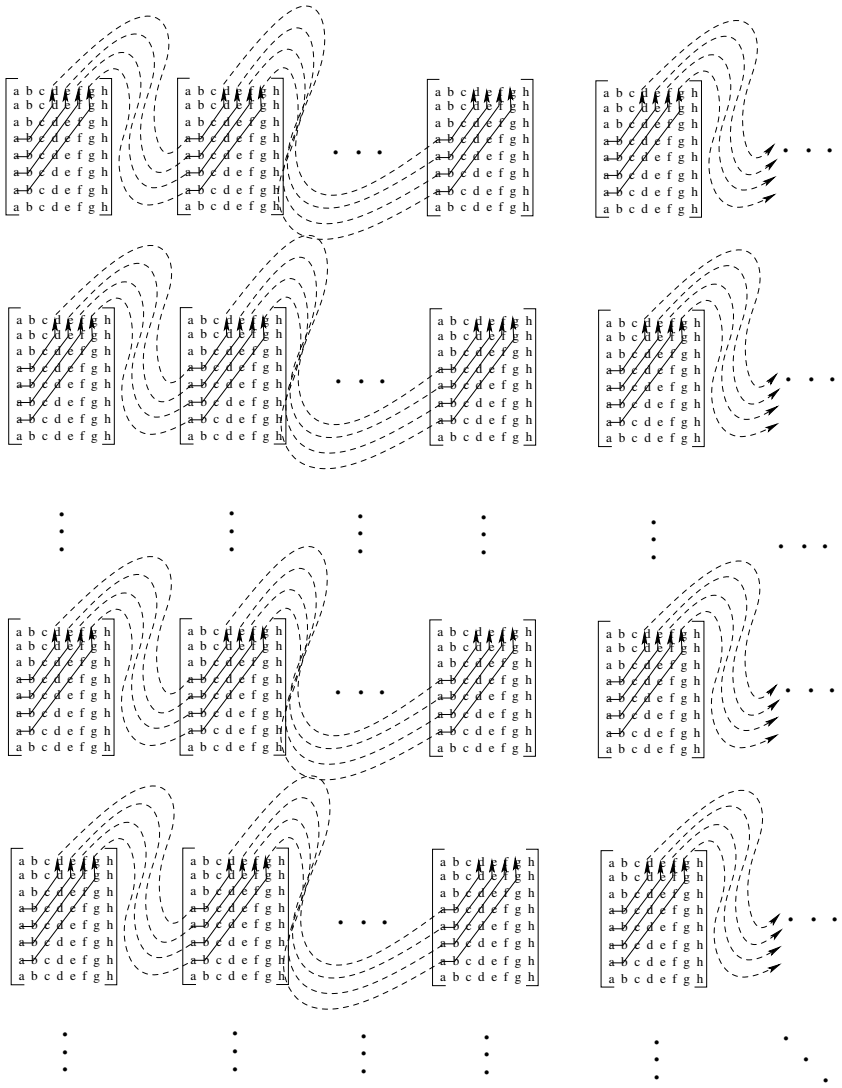


Figure 2.4: *Some sophisticated algorithm for the formation of* $\boldsymbol{i_j}$ *vectors. Every matrix represents a* 8×8 *transformed DCT block.*

2.4.1 Insertion of Watermark

Let $\boldsymbol{X} = [x_{ij}]$ be a gray level image of size $N \times N$ and $\boldsymbol{W^{in}} = [w_{ij}^{in}]$ be the binary watermark of size $M \times M$. First of all 8×8 blocked Discrete Cosine Transformation (DCT) or Discrete Wavelet Transformation (DWT) is applied to the image $\boldsymbol{X}$, In case of spatial domain watermarking select image pixels to embed watermark. Therefore, we have the following options:

$$\boldsymbol{Y} = DCT(\boldsymbol{X}) \mid DWT(\boldsymbol{X}) \mid \boldsymbol{X}$$

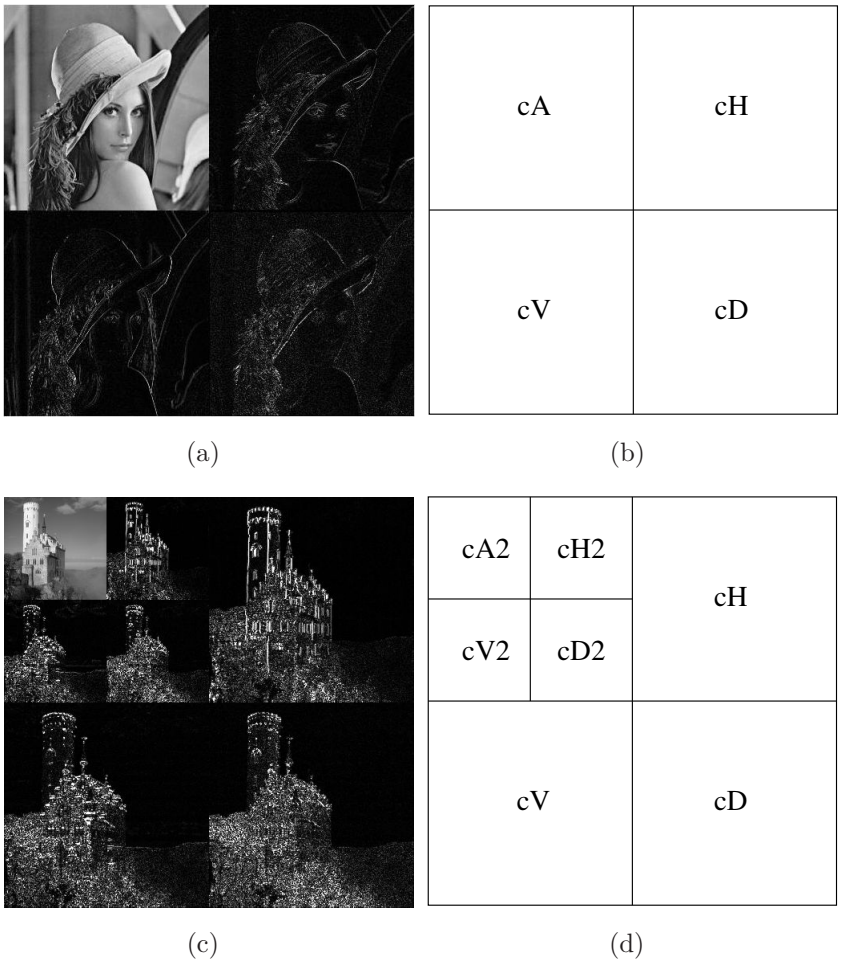


Figure 2.5: *(a) Single layer DWT. (b) Single layer DWT coefficient matrices. (c) Two layer DWT. (d) Two layer DWT coefficient matrices.*

Now binary bits of the watermark are converted into antipodal bits to form the representation of the watermark $\boldsymbol{W} = [w_{ij}]$, where $w_{ij} = \begin{cases} 1 & \text{if } w_{ij}^{in} = 1 \\ -1 & \text{if } w_{ij}^{in} = 0 \end{cases}$. The watermark bits are pseudo randomly permuted and the seed is recorded (see e.g. [8, 9]), which is later included in the private key.

$$\boldsymbol{W_{per}} = permute(\boldsymbol{W})$$

$\boldsymbol{W_{per}}$ is a matrix of antipodal bits. It is now converted into a row vector $\boldsymbol{w_{per}}$ of length $m = M \cdot M$ (Fig. 2.1). Divide this vector $\boldsymbol{w_{per}}$ in "k"

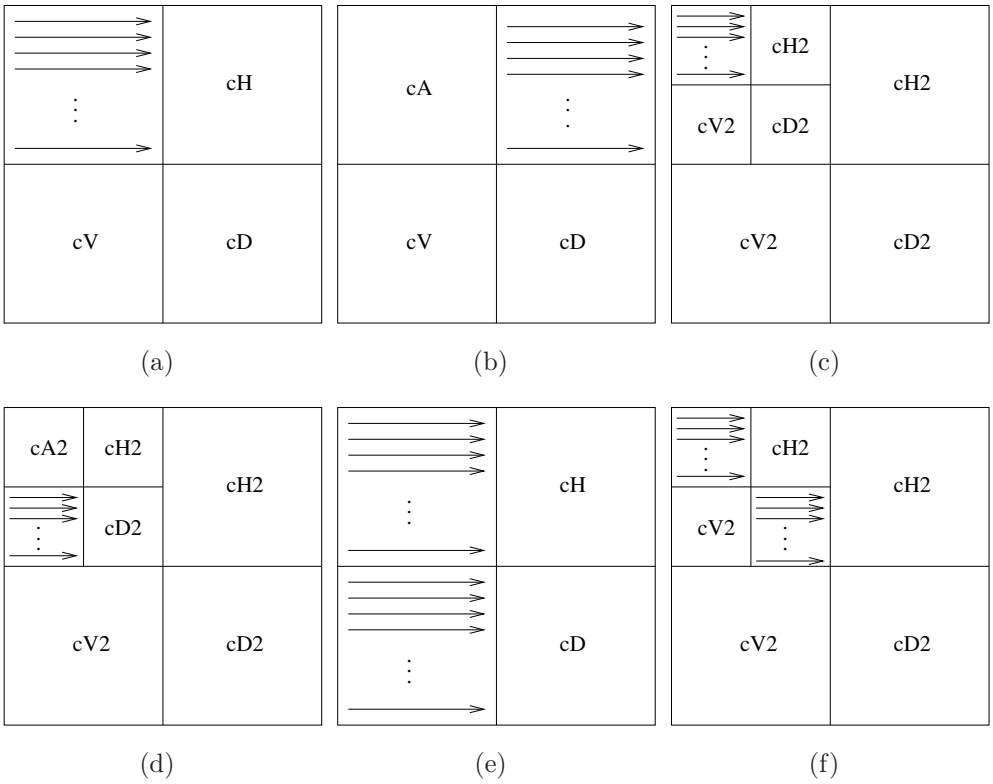


Figure 2.6: *Different formation of $\boldsymbol{i_j}$ vectors in DWT domain.*

parts each of length "l", where $k \cdot l = m$

$$\begin{aligned} \boldsymbol{w_{per}} &= [\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_k}] \\ \boldsymbol{w_1} &= [b_{11}, b_{12}, \ldots, b_{1l}] \\ \boldsymbol{w_2} &= [b_{21}, b_{22}, \ldots, b_{2l}] \\ &\vdots \\ \boldsymbol{w_k} &= [b_{k1}, b_{k2}, \ldots, b_{kl}] \end{aligned}$$

Select "k" zero mean orthogonal spreading codes $\boldsymbol{s_i}$, $1 \leq i \leq k$. Form "l" $\boldsymbol{i_j}$ vectors (see Sec. 2.3). Length of $\boldsymbol{s_i}$ and $\boldsymbol{i_j}$ must be same. Watermark is inserted by modifying $\boldsymbol{i_j}$ to $\boldsymbol{i'_j}$ according to:

$$\boldsymbol{i'_j} = \boldsymbol{i_j} + \alpha[b_{1j}\boldsymbol{s_1} + b_{2j}\boldsymbol{s_2} + \ldots + b_{kj}\boldsymbol{s_k}] \tag{2.9}$$

All $\boldsymbol{i_j}$ vectors are modified according to (2.9). Now replace every $\boldsymbol{i_j}$ by $\boldsymbol{i'_j}$ in $\boldsymbol{Y}$ to form $\boldsymbol{Y'}$. By applying Inverse Discrete Cosine Transformation (IDCT) or Inverse Discrete Wavelet Transformation (IDWT) to $\boldsymbol{Y'}$ the watermarked image $\boldsymbol{X'}$ is obtained. For spatial domain watermarking $\boldsymbol{Y'}$ is already a watermarked image $\boldsymbol{X'}$. Therefore, we obtain depending

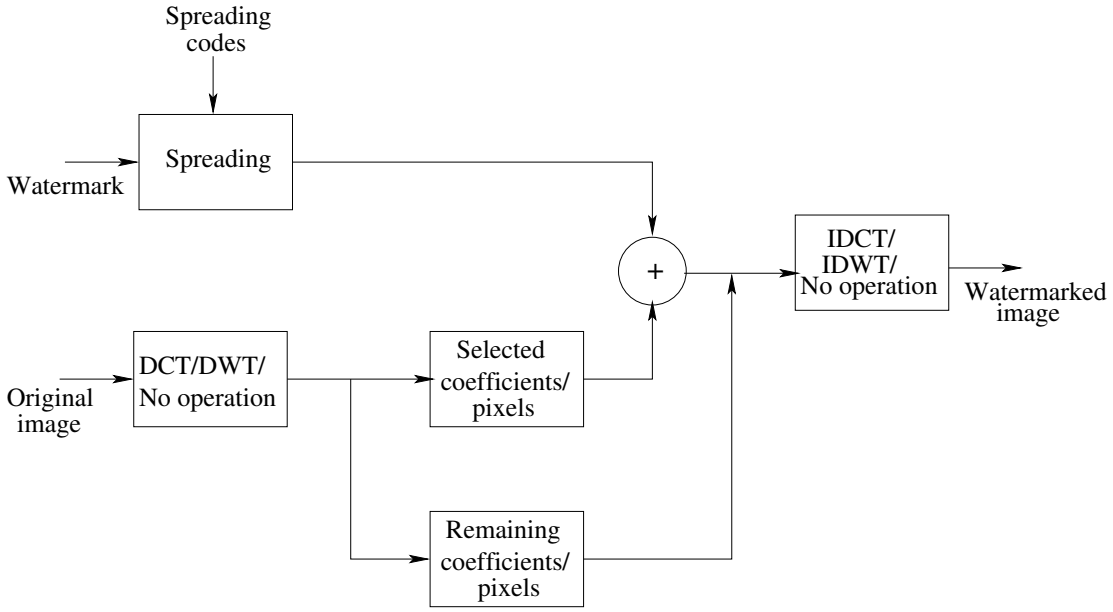


Figure 2.7: *Insertion of watermark.*

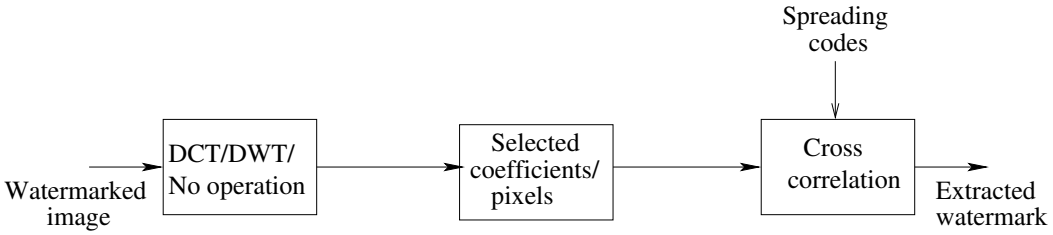


Figure 2.8: *Extraction of watermark.*

on the used transformation/spatial domain.

$$\boldsymbol{X'} = IDCT(\boldsymbol{Y'}) \mid IDWT(\boldsymbol{Y'}) \mid \boldsymbol{Y'}$$

2.4.2 Extraction of Watermark

Take the watermarked image $\boldsymbol{X'}$ transform to same domain used for watermark insertion.

$$\hat{\boldsymbol{Y}} = FDCT(\boldsymbol{X'}) \mid FDWT(\boldsymbol{X'}) \mid \boldsymbol{X'}$$

Use the same vectors $\hat{\boldsymbol{i}}_j$ of $\hat{\boldsymbol{Y}}$ that were used for the insertion and calculate the cross correlation with the spreading codes. If cross correlation

value is positive the detected bit is "1" and if the value is negative then the detected bit is "-1", i.e.

$$\hat{b}_{ij} = sign < \hat{\boldsymbol{i}}_{\boldsymbol{j}}, \boldsymbol{s_i} >$$

All parts of $\hat{\boldsymbol{w}}_{\boldsymbol{per}}$ can be extracted by calculating cross correlation between all $\hat{\boldsymbol{i}}_{\boldsymbol{j}}$ vectors and all spreading codes $\boldsymbol{s_i}$. Arrange these parts as $[\hat{\boldsymbol{w}}_{\boldsymbol{1}}, \hat{\boldsymbol{w}}_{\boldsymbol{2}}, \ldots, \hat{\boldsymbol{w}}_{\boldsymbol{k}}]$ to form the vector $\hat{\boldsymbol{w}}_{\boldsymbol{per}}$ and later rearrange it into the matrix $\hat{\boldsymbol{W}}_{\boldsymbol{per}}$. Inverse permute $\hat{\boldsymbol{W}}_{\boldsymbol{per}}$ with the same seed as used in insertion to get $\hat{\boldsymbol{W}}$.

$$\hat{\boldsymbol{W}} \;=\; Inverse\ permute(\hat{\boldsymbol{W}}_{\boldsymbol{per}})$$

Replace minus ones with zeros in $\hat{\boldsymbol{W}}$ to get the extracted watermark $\boldsymbol{W}^{\boldsymbol{out}}$.

2.4.3 Quality of Image and Similarity of Watermarks

The quality of the gray scale watermarked image is measured by the Peak Signal to Noise Ratio (PSNR).

$$\begin{aligned} \text{PSNR} &= 10log\frac{255^2}{MSE}\ db \\ &= 20log\frac{255}{RMSE}\ db \end{aligned} \tag{2.10}$$

Here, MSE is mean square error and $RMSE$ is root mean square error between original and watermarked pixel values.The similarity of the original and the extracted watermark is measured in terms of Bit Error Rate (BER).

$$\text{BER} = \frac{erroneous\ bits}{total\ number\ of\ bits} \tag{2.11}$$

2.4.4 Capacity of Watermark

The number of watermark bits that can be inserted in an $N \times N$ image is given by:

$$n = \frac{N^2 k}{l} \tag{2.12}$$

Here,

$$\begin{aligned} n &: \text{number of watermark bits} \\ k &: \text{number of spreading codes} \\ l &: \text{length of spreading codes} \end{aligned}$$

In case of single layer wavelet transformation if watermark is inserted in one coefficient matrix (cA, cH, cV or cD), the number of available frequency coefficients is $\frac{N^2}{4}$. In case of DCT watermarking, if only middle frequency coefficients are used, available coefficients are $\frac{N^2}{2}$.

2.4.5 Robustness

Equation (2.8) shows that the robustness of the proposed watermarking scheme directly depends on the value of α and the length of the spreading codes l. Since frequency coefficients, hence length of spreading codes, are limited, the robustness is controlled by α.

2.5 Experimental Results

The watermarking scheme, presented in this chapter, is tested against a variety of attacks by using checkmark 1.2 [38]. In all the experiments thirty 512×512 gray scale images are used (see appendix A for original image). Fig. 2.9(a) to 2.9(f) show the watermarked images inserted in cA, cH, cV, cD, DCT coefficients and spatial domain, respectively. Watermarks are added at PSNR value of 40db in all the images. In case

of DCT, $\boldsymbol{i_j}$ vectors are formed in 3rd, 4th, 5th and 6th rows of 8×8 DCT coefficients blocks. In spatial domain watermarking, watermark is added to 256 rows of the images. Checkmark 1.2 only tells whether the watermark is detected or not. So, the watermark detector is programmed at threshold 70%. In case of 70% threshold, the watermark is considered as detected if BER ≤ 0.3. Tables 2.1 to 2.3 present percentage of correctly detected watermarks. Every scheme is tested on all thirty images using a 32×32 (1024 bits long) binary watermark. Tables 2.1 shows the experimental results, each entry shows percentage of correctly detected watermark from 30 images. Watermark inserted in spatial domain is not very robust. Watermark inserted in cA gives the best result, it is far more robust then the watermark inserted in cH, cV, cD and even in DCT. Watermark inserted in cH, cV, or cD are not very robust. However, if watermark is inserted in cH, cV or cD watermarked image is more crisp than the cA based watermarked image. So, if robustness is not the issue cH, cV, and cD can also be used for watermark insertion.

Figure 2.9: *(a) Watermarked image using cA. (b) Watermarked image using cH. (c) Watermarked image using cV. (d) Watermarked image using cD. (e) Watermarked image using DCT domain. (f) Watermarked image using spatial domain.*

The following experiments are done again on all thirty images using 64 bits long watermark. Watermark is inserted in cA2, cH2, cV2 and cD2

Table 2.1: 1024 bits Watermark, threshold 70%

Attacks (no. of attacks)	Spatial	DCT	cA	cH	cV	cD
reSample(30)	100%	100%	100%	100%	93%	100%
Filtering(90)	100%	100%	100%	100%	100%	100%
ColorReduce(60)	88%	92%	65%	95%	83%	97%
MAP(180)	85%	74%	92%	67%	67%	32%
Wavelet(300)	0%	63%	74%	47%	61%	43%
JPEG(360)	98%	79%	98%	39%	71%	25%
ML(210)	50%	30%	74%	10%	12%	1%
Remodulation(120)	6%	0%	78%	0%	8%	0%
Copy(30)	0%	10%	60%	0%	10%	0%
Average(1380)	57%	61%	84%	44%	55%	33%

(one 256 bits long spreading code is used). In case of DCT, watermark is inserted in 4th row of DCT coefficients (one 512 bits long spreading code is used). In spatial domain watermarking, watermark is added to 64 rows of the images (one 512 bits long spreading code is used). The watermark detector is programmed at thresholds 70% and 90%. Average of all experimental results are shown in tables 2.2 and 2.3. Images are watermarked at PSNR value of 40db in all the experiments. Tables 2.2 and 2.3 shows that DWT based watermarking scheme is more robust than DCT and spatial domain watermarking. Results show that proposed 2 layer DWT based algorithm is very robust against a variety of attack, no matter watermark is inserted in cA2, cH2, cV2 or cD2. However cA2 still gives the best result. Results show that this scheme is very robust against jpeg compression as well as wavelet compression. Checkmark 1.2 compresses the watermarked images with a quality factor as low as 10%. Table 2.2 shows that, in case of cA2, average 100% and 87% of the watermarks are detected against jpeg and wavelet compression respectively. Details of all the attacks can be seen in [38].

Table 2.2: 64 bits Watermark, threshold 70%

Attacks (no. of attacks)	Spatial	DCT	cA2	cH2	cV2	cD2
reSample(30)	100%	100%	100%	100%	97%	100%
Filtering(90)	100%	100%	100%	100%	100%	100%
ColorReduce(60)	100%	100%	100%	100%	100%	100%
MAP(180)	76%	100%	100%	100%	97%	100%
Wavelet(300)	70%	88%	87%	93%	91%	96%
JPEG(360)	90%	99%	100%	100%	99%	99%
ML(210)	30%	83%	100%	84%	96%	84%
Remodulation(120)	0%	0%	100%	29%	93%	36%
Copy(30)	0%	0%	0%	0%	0%	0%
Average(1380)	66%	84%	95%	88%	94%	89%

Table 2.3: 64 bits Watermark, threshold 90%

Attacks (no. of attacks)	Spatial	DCT	cA2	cH2	cV2	cD2
reSample(30)	100%	100%	100%	100%	87%	97%
Filtering(90)	100%	100%	100%	100%	98%	100%
ColorReduce(60)	95%	93%	87%	93%	78%	97%
MAP(180)	52%	89%	91%	92%	84%	92%
Wavelet(300)	50%	61%	55%	70%	64%	78%
JPEG(360)	64%	89%	97%	98%	89%	92%
ML(210)	29%	40%	84%	65%	76%	64%
Remodulation(120)	0%	0%	70%	3%	63%	1%
Copy(30)	0%	3%	17%	0%	37%	0%
Average(1380)	52%	67%	81%	76%	78%	76%

Figure 2.10: *(a)Lena image. (b) Milk drop image. (c) Gold hill image. (d)64 bits Watermarked Image at PSNR=40db using DCT. (e)1024 bits Watermarked Image at PSNR=40db using DCT. (f)4096 bits Watermarked Image at PSNR=40db using DCT.*

2.5.1 Comparison

Most of the people who implemented CDMA based watermarking schemes used small watermarks, compared to the one used here (longest 4096 bits long) Fig. 2.10. In [12], Vassaux used comparatively long watermarks (longest 2048 bits), but BER is quite high (more than 0.3 for 4 spreading codes at wPSNR=40db)[1]. As suggested, by applying smart selection of frequency coefficients BER can be fairly reduced. Extracted watermarks from Lena, Milk drop and Gold hill (at 40db and 45db) by using proposed scheme for 4 spreading codes are shown in Fig. 2.11. BER is considerably less than 30% even at PSNR=45db.

A 64 bits long watermark spread with 8 mutually orthogonal spreading codes is inserted in Lena image. This watermarked image is compressed by using JPEG compression. The results are compared with Xin algorithm [13] (Fig. 2.12). Both watermarked images have PSNR value of 40db, used CDMA-based watermarking and hide 64 bits long

[1]This is according to data provided in [12].

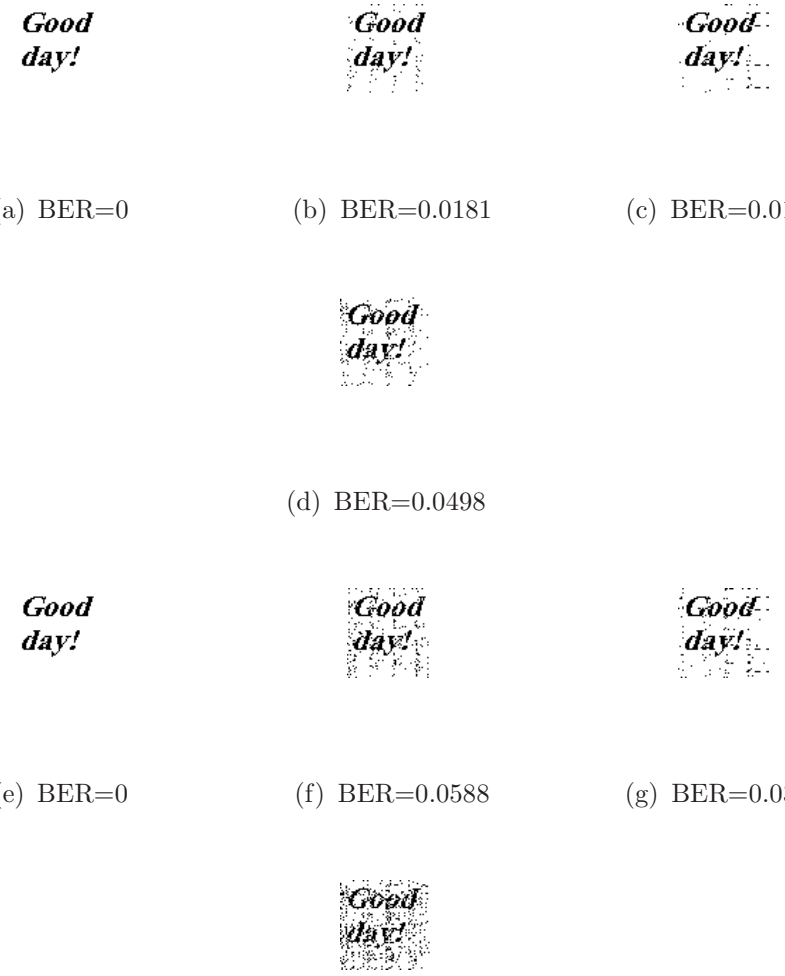
watermarks. By using the proposed algorithm there is no bit error up to 15% quality factor. It can easily be seen how robust CDMA based scheme becomes if watermark is inserted in specially selected frequency coefficients.

2.5.2 Multiple Spreading codes

A watermark can be embedded in an image by using one spreading code. However by using multiple orthogonal spreading codes watermarking scheme becomes more robust and carry more payload. Fig. 2.12 shows that watermarking scheme implemented using 8 spreading codes is more robust than watermarking implemented using one spreading code in DCT domain.

2.6 Conclusions

A robust, oblivious and secure digital watermarking scheme is proposed in this chapter. This scheme is based on CDMA technique and zero mean code. Simulations proved that this algorithm is a lot better than other simple spread spectrum (CDMA based with one spreading code) watermarking algorithms. The algorithm is robust against unintentional attacks as well as malicious attacks. By using multiple spreading codes the scheme is more secure and can carry more payload. Because of an intelligent selection of frequency coefficients, where the watermark is to hide, this algorithm becomes robust against a number of attacks. Although spatial domain and frequency domain both can be used for watermark insertion. But simulations have shown that spatial domain is not a good choice for watermarking. So further only frequency domain watermarking is discussed and enhanced.



(a) BER=0 (b) BER=0.0181 (c) BER=0.0198

(d) BER=0.0498

(e) BER=0 (f) BER=0.0588 (g) BER=0.0376

(h) BER=0.1282

Figure 2.11: *(a)* 64×64 *bits original watermark. (b)Extracted watermark from Lena Image at 40db. (c)Extracted watermark from Milk drop image at 40db. (d) Extracted watermark from Gold hill image at 40db. (e)* 64×64 *bits original watermark. (f) Extracted watermark from Lena Image at 45db. (g) Extracted watermark from Milk drop image at 45db. (h) Extracted watermark from Gold hill image at 45db.*

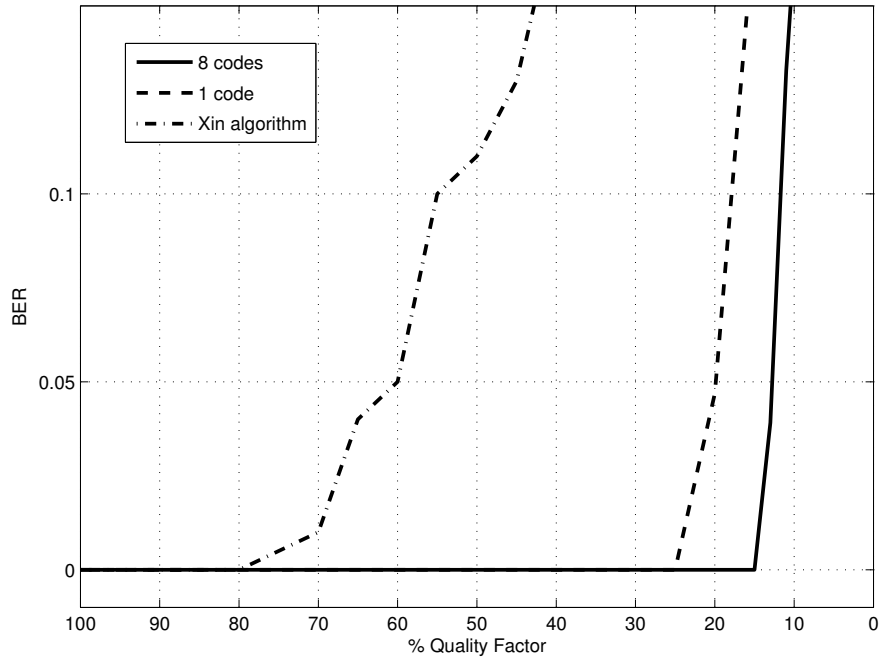


Figure 2.12: *Comparison between proposed algorithm and Xin algorithm against JPEG compression.*

3 Enhancing CDMA Based Watermarking

3.1 Introduction

Biggest disadvantage of general watermarking algorithms in practice is that they permanently distort original image. In sensitive applications like military or medical imagery original image is required after watermark extraction. This requirement creates another category of watermarking schemes called reversible watermarking schemes. In these schemes, the watermark can be removed completely after watermark extraction (detection) to retrieve original image back.

In proposed algorithm, previous chapter, spreaded watermark is arithmetically added, in spatial or frequency domain. Therefore, it is not difficult to remove it after watermark detection [39]. The extracted watermark is spread again using same spreading codes and subtracted from the modified coefficients to get original image back. In proposed algorithm, watermark can be extracted and removed only by using spreading codes. Original watermark is not required during extraction as well as removal process.

Proposed algorithm is very flexible and watermark can be added at a wide range of PSNR values. It is very robust even if watermark is added at high PSNR value. So, it can be used as an irreversible watermarking algorithm for normal distribution of images. When a very strong watermark is needed, watermark is added at low PSNR value and visible on watermarked image, but it looks like noise. If a very noisy watermark is added, watermarked image becomes useless for unauthorized users. Authorized users can detect the watermark and later remove it to get noise free original image.

Compared to previous reversible watermarking schemes, proposed scheme has certain advantages. Visible watermarks are usually added to specific areas of image [40], but in proposed algorithm visible watermark is spread over whole image in the form of noise. Reversible watermarking schemes use properties of image [18], [19] and thus are image format dependent. Some of these schemes were not applicable to compressed images. Therefore, some more reversible watermarking schemes were presented for compressed images [41]. Proposed algorithm is independent of image format. Watermark is arithmetically added in frequency domain. It does not matter whether the frequency coefficients are quantized or not. Presented algorithm is very robust, unlike most of reversible watermarking scheme which are fragile [42], [43]. In proposed algorithm original watermark is not needed during watermark removal process. Therefore, this scheme is perfect for medical applications where patient name and date of birth are embedded as a watermark.

During the past few years a lot of research has been done to improve digital watermarking. Many researchers, very truly, consider the watermarking systems similar to a communication system [10]. Therefore, in order to improve the robustness of the digital watermarking schemes, many existing techniques in communication systems are also applied to the watermarking algorithms. Channel coding (application of Error Correcting Codes (ECC)) is one of the main examples. A very important fact which is continuously being neglected is that in order to apply ECC in watermarking algorithms one has to reduce the strength of the watermark per bit, as the number of bits in the coded watermark is increased compared to the number of bits in the original watermark. This is necessary to maintain a similar perceptual transparency. So, ECC is increasing the robustness but at the same time because of ECC the robustness decreases as the magnitude of watermark signal per bit is decreasing.

In [15], [16], [44] and [45] Turbo codes are used while in [15] and [46] Bose-Chaudhuri-Hocquenghen (BCH) codes are used for increasing robustness against attacks. In [47] Low Density Parity Check (LDPC) codes are used to improve payload. But none of these works considers the fact that because of applying these coding schemes one has to reduce the strength of the watermark.

Length of the spreading codes is a direct measure of watermark

strength in CDMA based watermarking schemes. In this chapter channel coding is applied at the expense of a reduction of the length of the spreading codes. Therefore, one can easily judge whether channel coding is useful or not in CDMA based watermarking. The simulation results put a clear question mark on the application of computationally complex channel coding in watermarking algorithms under strong attacks. ECC can correct the errors when they are small in numbers, at that very point watermark is easily readable even without ECC. When the attacks grow stronger, ECC collapse and resultant BER is worse than BER without ECC.

Furthermore, a new strategy of by parts interleaving of spreaded bits is introduced. [44] concluded that with the addition of channel coding the watermarking algorithms become weak against geometric attacks. By applying by-parts interleaving, BER can be improved against geometric attacks (Sec. 3.4), whether long spreading codes are used or shorter spreading codes with channel coding are applied. If these mutually similar frequency coefficients are selected from different regions of the image, the watermarking scheme becomes more robust against geometric attacks.

3.2 Reversible Watermarking Algorithm

Fig. 3.1 explains the watermark removal process. Extracted watermark is spread again using same spreading codes and subtracted from the selected frequency coefficients of watermarked image. After applying inverse transformation original image is obtained. So, only by using spreading codes watermark can be removed. If the intensity of the watermark "α" is known, watermark can be removed after extraction using:

$$\boldsymbol{i_{j2}} = \hat{\boldsymbol{i}}_{\boldsymbol{j}} - \alpha[\hat{b}_1\boldsymbol{s_1} + \hat{b}_2\boldsymbol{s_2} + \ldots + \hat{b}_k\boldsymbol{s_k}] \tag{3.1}$$

Now all $\hat{\boldsymbol{i}}_{\boldsymbol{j}}$ are replaced by $\boldsymbol{i_{j2}}$ again to get original image back. Another advantage of the proposed scheme is that original watermark is not needed during watermark removal process.

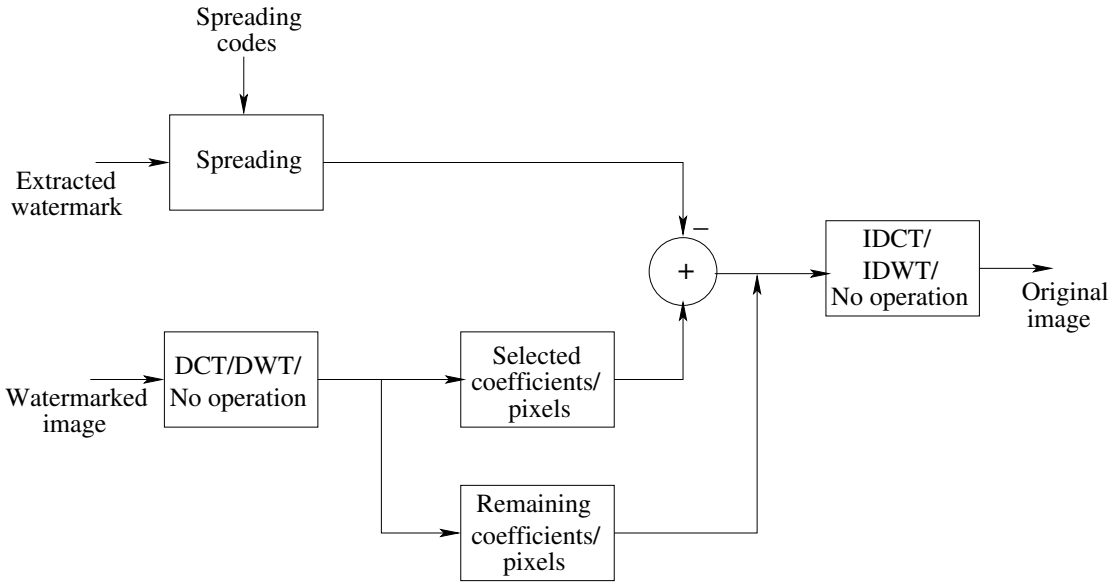


Figure 3.1: *Watermark removal.*

3.3 Addition of Channel Coding

Equation (2.6) shows that bits of the watermark are correctly detected if $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$ is large. To make the algorithm robust $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$ should be as large as possible. $(\boldsymbol{s_i} \cdot \boldsymbol{s_i^T})$ is always positive and $|\boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$ is large if the length of $\boldsymbol{s_i}$ is large. If α is considered as constant the length of the spreading code $\boldsymbol{s_i}$ is the only quantity which determines the value of $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$. Hence the length of the spreading codes is the measure of the robustness in CDMA based algorithms. Obviously there is an upper bound on the length of the spreading codes as the number of frequency coefficients is limited.

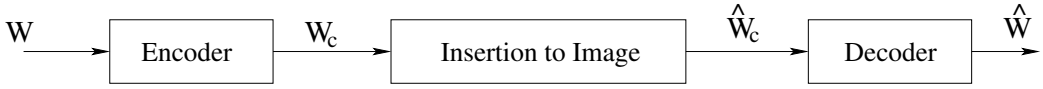


Figure 3.2: *Channel Coding in Digital Watermarking.*

Fig. 3.2 shows how addition of channel coding works. Here

$$length\ of\ \boldsymbol{W_c}\ >\ length\ of\ \boldsymbol{W} \tag{3.2}$$

where, $\boldsymbol{W_c}$ is the coded watermark and $\boldsymbol{W}$ is the original watermark. Therefore, in order to accommodate higher number of bits of $\boldsymbol{W_c}$, the

Table 3.1: BER in extracted watermarks for variable length spreading codes under JPEG compression attacks (PSNR=40db).

Quality Factor	Spreading Code Lengths		
	128	256	512
80%	0.1023	0.0439	0.0151
70%	0.1634	0.0925	0.0459
60%	0.2353	0.1545	0.0865

length of the spreading codes must be shorted accordingly. How decreasing the length of spreading codes affects the detection of the watermark is shown in tables 3.1 and 3.2. Table 3.1 elaborates how change in the length of spreading codes effects on BER under JPEG compression and table 3.2 shows change in BER under added random noise. Here 512×512 Lena image, 30 different 1024 bit long watermarks and 512 bit long spreading codes are used. To observe the BER under variable length spreading codes, in case of 128 bits long spreading codes 384 bits of original spreading codes are replaced by dummy bits and in case of 256 bits long spreading codes remaining 256 bits are dummy. In the band of BER 4% to 20% relation between length of spreading codes and BER is almost linear, i.e. by doubling the spreading codes BER becomes half. Entries 1 in both the tables shows, in the region below 4% by increasing spreading code by a factor n/k, BER improves more than a factor k/n. Region above 20% is not considered as in this region probability of error by using ECC is very high. The probability equation for errors using linear block codes (n,k,t) in extracted watermark according to [48] is:

$$P(E) = \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{3.3}$$

If ECC is applied and number of bits are increased by a factor n/k, probability of error also increased by a factor n/k, as spreading code is decreased by a factor k/n. So, for calculating effective probability of

Table 3.2: BER in extracted watermarks for variable length spreading codes under added random noise attack (PSNR=40db).

PSNR	Spreading Code Lengths		
	128	256	512
23.5db	0.1340	0.0595	0.0139
21.5db	0.1788	0.0971	0.0356
20.0db	0.2249	0.1409	0.0652

error equation (3.3) can be modified as:

$$P(E) = \sum_{i=t+1}^{n} ({}_{i}^{n})(\frac{n}{k} \times p)^{i}(1 - \frac{n}{k} \times p)^{n-i} \tag{3.4}$$

Fig. 3.3 shows different BCH codes and their respective effective BCH codes after increasing the probability of error by a factor n/k. It can be seen even if a strong linear block code is applied, because of reduction of spreading code, the effective probability of error is same. So, ECC can improve BER when it is about 2-3% (without using ECC) and afterwards probability of error shoots by using ECC.

3.4 By-parts Interleaving

Interleaving [49] is another concept from communication, which is used in proposed watermarking algorithm. To make the algorithm more robust against geometric attacks the $\boldsymbol{i_j}$ vectors can be formed by using by-parts interleaving over the whole image, e.g. selecting same rows from different blocks in different regions, in DCT domain, of the image. Fig. 3.4 shows an example. By-parts interleaving can be applied to all the different ways for forming $\boldsymbol{i_j}$ vectors discussed in Sec. 2.3.

By forming $\boldsymbol{i_j}$ vectors in such a way that it consists of similar coefficients but in different regions makes the algorithm more robust against geometric attacks. Consider a geometric attack in which the lower portion of the watermarked image is shrink but the upper portion is untouched or less shrink. Probability of correct detection of a watermark bit which is spread in both upper and lower regions of the image is

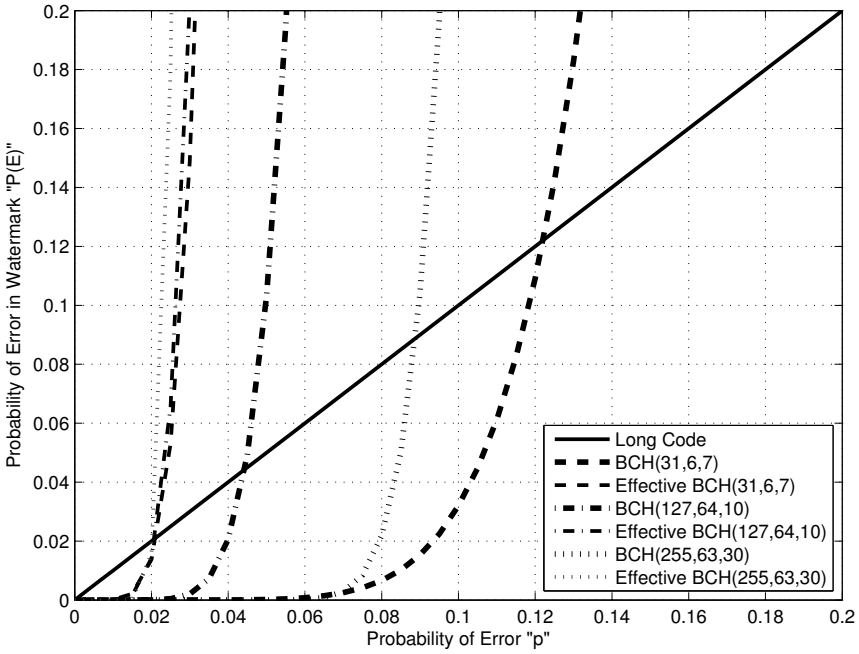


Figure 3.3: *Probability of errors in BCH coded watermarks versus effective probability of errors in BCH coded watermarks.*

higher than a bit which is spread in just one region. This is confirmed by experimental results (Sec. 3.5.3).

3.5 Experimental Results

3.5.1 Reversible Watermarking

DCT Domain

In the following experiments 512×512 gray scale images (Lena, Milk drop and Gold hill) are used. A binary watermark 32×32 (1024 bits long) is spread using four 512 bits long mutually orthogonal spreading codes. $\mathbf{i_j}$ vectors are formed in 3rd, 4th, 5th and 6th rows of 8×8 DCT coefficients blocks (Fig.2.2). Column 2 of tables 3.3 to 3.5 show the PSNR values of watermarked images at different α values. All

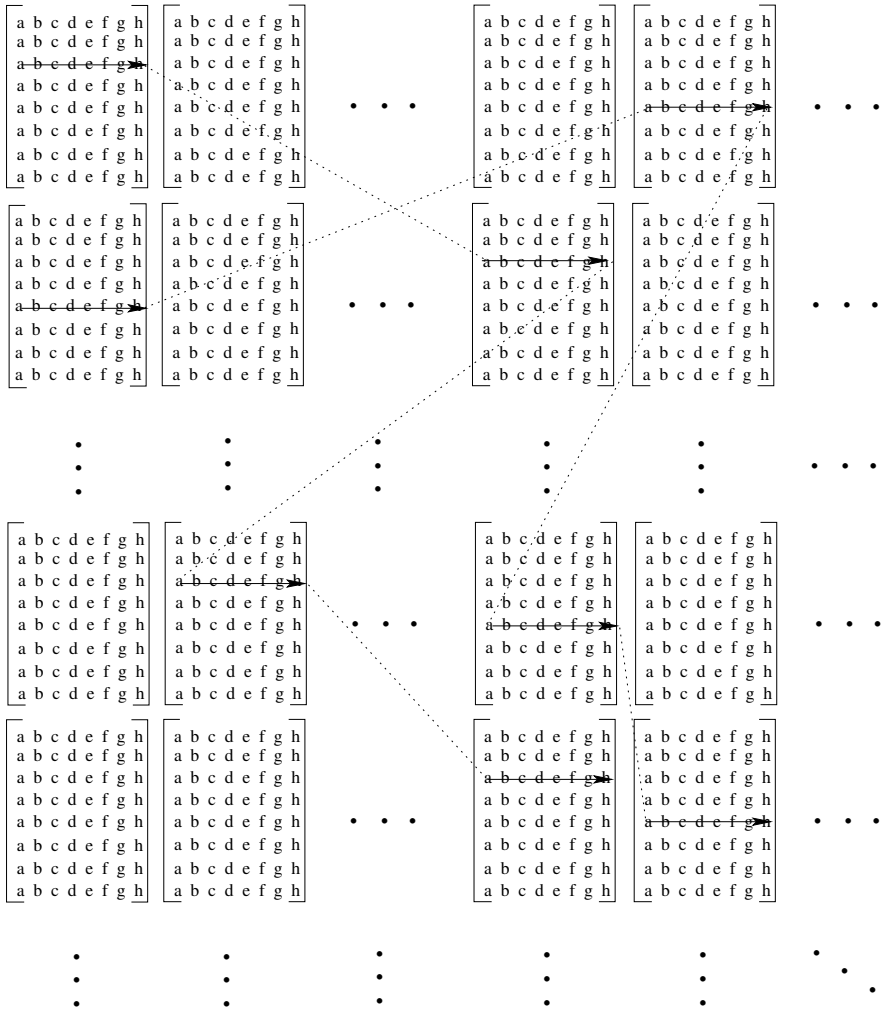


Figure 3.4: i_j *vectors are in parts interleaved in row-wise formation. Every matrix represents a* 8×8 *transformed DCT block.*

PSNR values are with respect to original images. Column 3 presents BER in extracted watermarks. Column 4 shows PSNR values after watermark removal using respective extracted watermarks. Tables 3.3 to 3.5 show that the PSNR values of restored images are so high (in some cases infinite) that the distortions can be considered as perceptually transparent. Generally, for PSNR values, anything over 40db is not visible.

Fig.3.5(a) presents a watermarked image which is very noisy (PSNR = 27.4328db). This image is watermarked using a 32×32 (1024 bits long) binary watermark Fig.3.5(e). This image is so noisy that it is useless for unauthorized users. Extracted watermark with no bit errors is shown in Fig.3.5(f). Fig.3.5(b) shows image after watermark re-

moval, which is identical to original image (PSNR=70.3462). Fig.3.5(c) and Fig.3.5(d) present zoom in version of Fig.3.5(a) Fig.3.5(b). Fig.3.6 shows the same results with a 64×64 (4096 bits long) watermark. Now the spreading codes are 128 bits long.

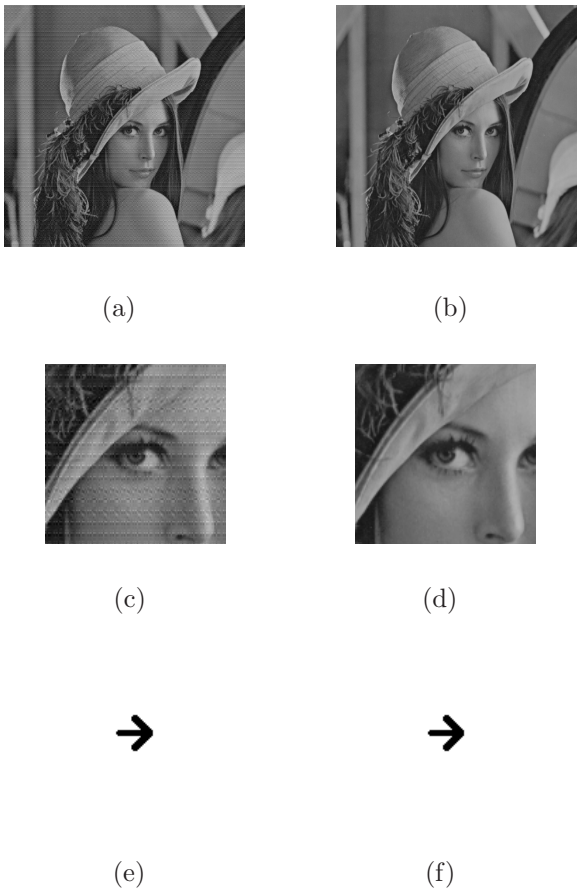


Figure 3.5: *(a)Watermarked Lena image (PSNR=27.4328db). (b)Lena after watermark removal (PSNR=70.3462). (c)Zoom in version of 3.5(a). (d)Zoom in version of 3.5(b). (e)*32×32 *original watermark. (f)Extracted watermark (BER=0%).*

Wavelet Domain

In the following experiment 512×512 gray scale Lena image is used. A binary watermark 32×32 (1024 bits long) is spread using four 256 bits long mutually orthogonal spreading codes. $\boldsymbol{i_j}$ vectors are formed in rows of single layer DWT coefficients matrices (Fig.2.6(a)). Tables 3.6 shows experimental results of reversible watermarking algorithm, here watermark is added to cA matrix. Tables 3.7 to 3.9 present results when

Table 3.3: Lena image.

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.005	42.9034	0.39 %	60.3451
0.007	40.0435	0%	Infinite
0.01	36.9268	0%	Infinite
0.02	30.9708	0%	Infinite
0.03	27.4328	0%	70.3462

Table 3.4: Milk drop image.

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.005	42.9034	1.46 %	55.0323
0.007	40.0435	0.29%	59.6756
0.01	36.9274	0%	87.1311
0.02	30.9974	0%	59.8520
0.03	27.5148	0%	50.4257

Table 3.5: Gold hill image.

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.005	42.9034	2.44 %	52.8087
0.007	40.0435	0.59%	56.0854
0.01	36.9268	0%	Infinite
0.02	30.9740	0%	73.5252
0.03	27.4588	0%	56.7638

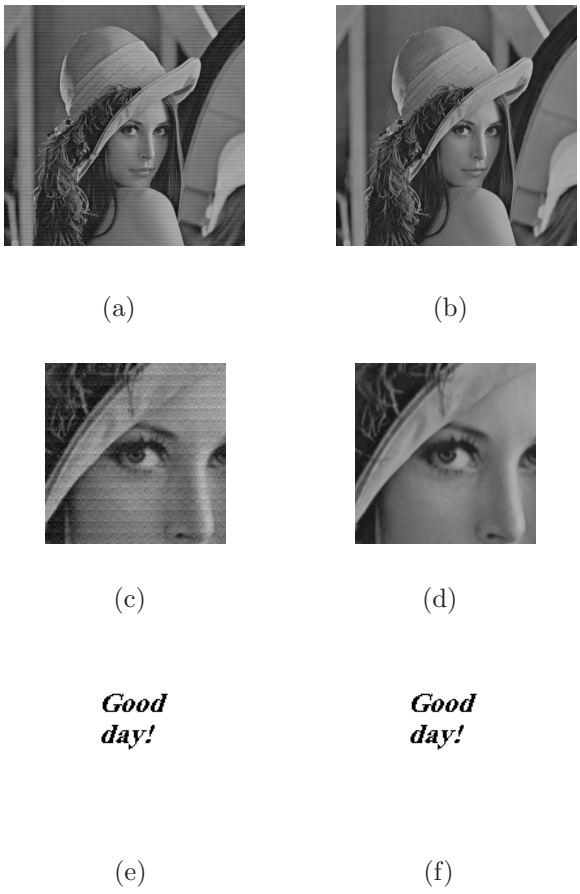


Figure 3.6: *(a)Watermarked Lena image (PSNR=27.4433db). (b)Lena after watermark removal (PSNR=84.4629). (c)Zoom in version of 3.6(a). (d)Zoom in version of 3.6(b). (e)*64×64 *original watermark. (f)Extracted watermark (BER=0%).*

watermark is added to cH, cV and cD matrices respectively. Column 2 of tables 3.6 to 3.9 show the PSNR values of watermarked images at different α values. All PSNR values are with respect to original image. Column 3 presents BER in extracted watermarks. Column 4 shows PSNR values after watermark removal using proposed algorithm. Tables 3.6 to 3.9 show that the PSNR values of restored images are so high (in some cases infinite) that the distortions can be considered as perceptually transparent. Generally, for PSNR values, anything over 40db is not visible.

Fig.3.7(a) presents a watermarked image ($\boldsymbol{i_j}$ vectors are formed in rows of cA) which is very noisy (PSNR=23.0344db). This image is watermarked using a 32×32 (1024 bits long) binary watermark Fig.3.7(e). Extracted watermark with no bit errors is shown in Fig.3.7(f). Fig.3.7(b)

shows image after watermark removal, which is identical to original image (PSNR=60.1466db). Fig.3.7(c) and Fig.3.7(d) present zoom in version of Fig.3.7(a) Fig.3.7(b).

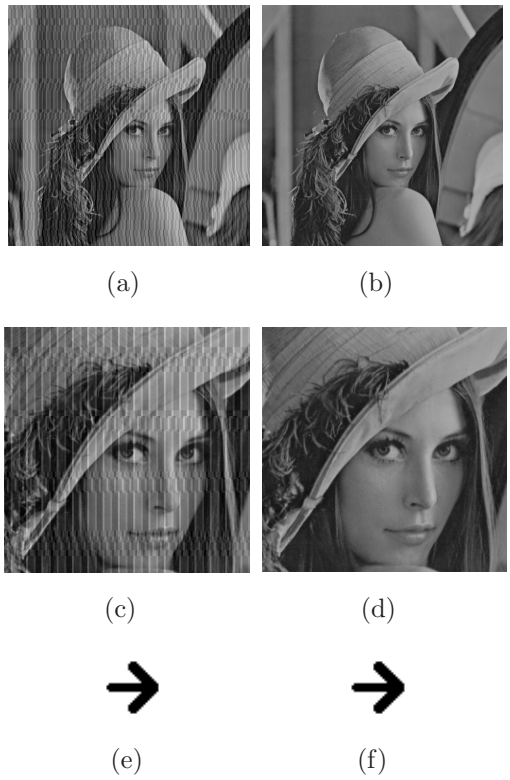


Figure 3.7: *(a) Watermarked Lena image (PSNR=23.0344db). (b) Lena after watermark removal (PSNR=60.0344). (c) Zoom in version of 3.7(a). (d) Zoom in version of 3.7(b). (e)* 32×32 *original watermark. (f) Extracted watermark (BER=0%).*

3.5.2 Using Error Correcting Code (ECC)

In the following experiments 512×512 (8 bits/pixel, gray scale) Lena image and 32×32, 1024 bits long (binary) watermarks are used. All experiments are done using 4 spreading codes, in DCT domain. In case of long spreading code 512 bits long spreading codes are used. For all coding schemes (BCH, LDPC and Turbo) a code rate of 1/2 is used, hence the length of the spreading codes is half of the original spreading codes i.e. 256 bits. In all experiment where coding is done using BCH code, BCH(127,64,10) is used. In case of Turbo codes generator matrix G = [111;101] is used and Log-MAP decoder with five iterations is used for decoding. In all experiments spreading codes are interleaved in 128

Table 3.6: Reversible watermarking using cA

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.008	42.1102	8.98 %	46.5459
0.01	39.3084	5.37%	45.7735
0.03	30.3403	0%	Infinite
0.05	25.9056	0%	57.7612
0.07	23.0344	0%	60.1466
0.09	20.9353	0%	49.7728

Table 3.7: Reversible watermarking using cH

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.008	42.1102	0%	Infinite
0.01	39.3084	0%	Infinite
0.03	30.3403	0%	Infinite
0.05	25.9330	0%	57.2685
0.07	23.0378	0%	58.6406
0.09	20.9503	0%	48.3645

Table 3.8: Reversible watermarking using cV

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.008	42.1102	0%	Infinite
0.01	39.3084	0%	Infinite
0.03	30.3403	0%	Infinite
0.05	25.9324	0%	57.3089
0.07	23.0383	0%	58.5603
0.09	20.9512	0%	48.2786

Table 3.9: Reversible watermarking using cD

α	PSNR (db) watermarked image	BER extracted watermark	PSNR (db) after watermark removal
0.008	42.1102	0%	Infinite
0.01	39.3084	0%	Infinite
0.03	30.3403	0%	Infinite
0.05	25.9320	0%	57.3175
0.07	23.0380	0%	58.5854
0.09	20.9509	0%	48.3143

bits parts. $\boldsymbol{i_j}$ vectors are formed in 3rd, 4th, 5th and 6th rows of 8×8 blocks of DCT coefficients.

JPEG Compression

In this experiment 30 different watermarks are tested against JPEG compression attack. Fig. 3.8 shows that the long code is clearly better than LDPC code. BCH code is only better when errors are less than 2% (same as presented in theory) afterwards long spreading codes are better. Convolution code is a little better but overall behavior of all the codes is the same. PSNR value is 40db in all tested images.

Uniqueness of Watermark

30 different watermarks are used and BER at different PSNR values are shown in Fig. 3.9. Again the crossing point of BCH and long code is somewhere between 2-3%. Convolution code is a little better when errors are small. But when errors are less watermark can be easily readable even without ECC. At high PSNR values long spreading codes give better results.

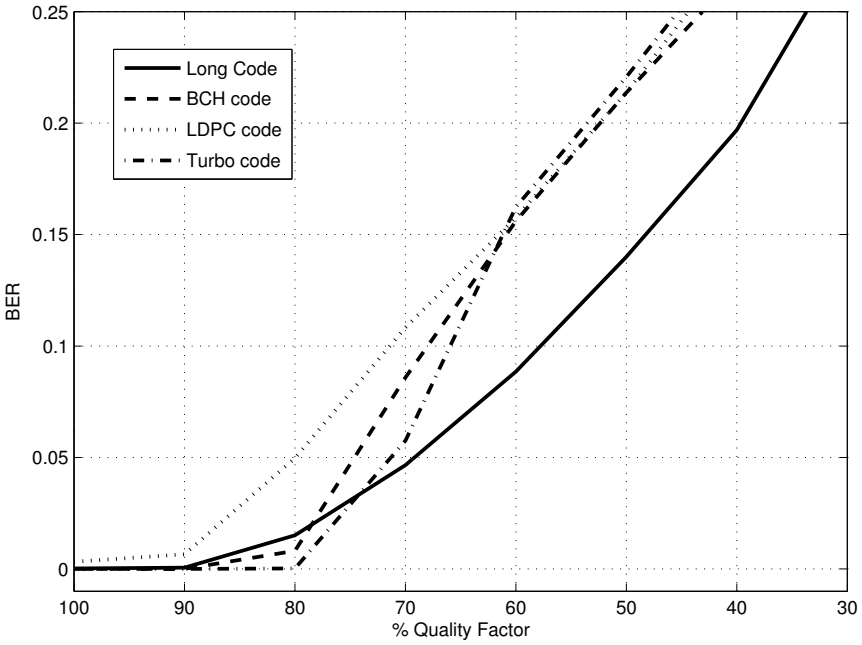


Figure 3.8: *Results under JPEG compression attack.*

Combined Attack

A combined test is also performed. Watermarked images at 40db using long spreading codes and short spreading codes with ECC are cropped[1] (25%) then compressed (quality factor 70%) and finally scaled[2] to 50% of their original sizes. Extracted watermark by using long code is better than channel coded watermarks under extreme attacks (Fig. 3.10).

3.5.3 Using By-parts Interleaving

The algorithm is also tested against copyright attacks that include geometric attacks. In case of geometric attacks, original size of the image is provided to the watermark detector as a side information. So whatever size of the image it gets, it resizes that image to its original size before extracting the watermark. Providing the dimensions of the original im-

[1] all the cropped pixels are replaced by white pixels (pixel value=1)

[2] nearest neighbor interpolation is used to scale the image in both directions

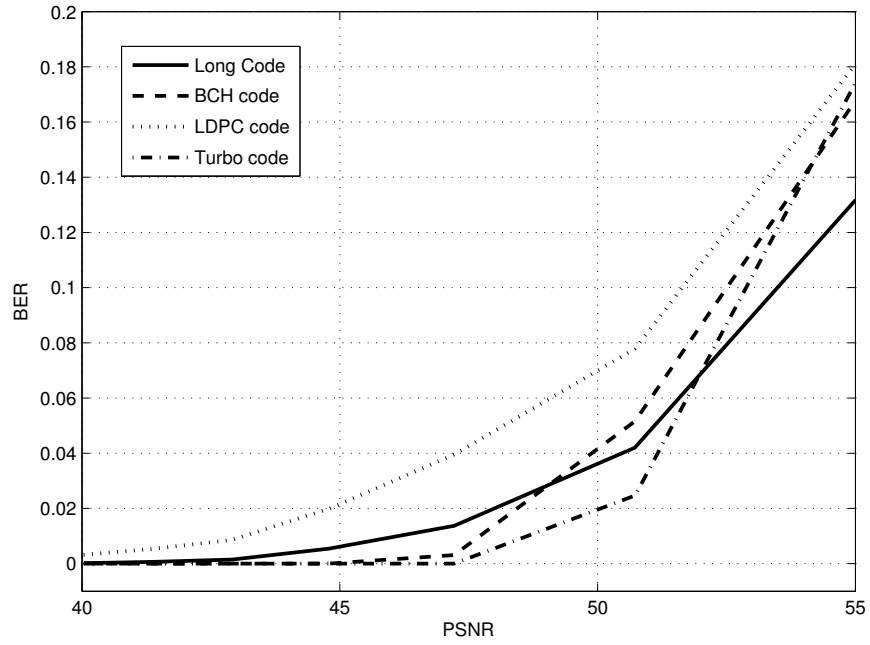


Figure 3.9: *Results at different PSNR values.*

age should not be a problem. It can very easily be sent to the watermark detector in the private key along with the spreading codes. Results are shown in table 3.10 (PSNR=40db, 64 bits watermark, threshold 51% and without using by-parts interleaving). Same tests are then performed again with by-parts interleaving (table 3.11). Every bit is interleaved in four equal parts. There is a significant improvement against geometric attacks if by-parts interleaving is used.

3.6 Conclusions

In this chapter, CDMA based watermarking scheme is enhanced by using by-parts interleaving. It is also shown that CDMA based watermarking scheme can easily be transformed into reversible watermarking scheme. Reversible version of proposed algorithm can extract the watermark and later recover the original image. Watermark can be extracted and removed only by using spreading codes. Original watermark is not required during extraction as well as removal process. Furthermore, how redundant bits in channel coding affect CDMA based watermark-

Attacks (no. of attacks)	Lena	Milk Drop	Gold Hill	Average
colorReduce(1)	100%	100%	100%	100%
Wavelet(10)	100%	100%	100%	100%
TemplateRemove(1)	100%	100%	100%	100%
Scale(42)	100%	100%	100%	100%
Row_col(49)	100%	100%	100%	100%
ML(7)	100%	100%	100%	100%
MAP(6)	100%	100%	100%	100%
JPEG(12)	100%	100%	100%	100%
Filtering(3)	100%	100%	100%	100%
Bending(2)	100%	100%	100%	100%
Aspectratio(35)	100%	100%	100%	100%
Shearing(14)	100%	100%	86%	95%
RotationScale(21)	71%	57%	62%	63%
Linear(21)	67%	62%	33%	54%
Rotation(21)	95%	19%	43%	52%
SampleDownUp(4)	50%	50%	50%	50%
Warping(28)	43%	50%	50%	48%
Projective(70)	23%	43%	30%	32%
Crop(28)	11%	21%	46%	26%
Remodulation(4)	0%	0%	25%	8%
Copy(1)	0%	0%	0%	0%
Collage(2)	0%	0%	0%	0%
Average(1146)	71%	68%	69%	70%

Table 3.10: 64 bits Watermark, threshold 51%

Attacks (no. of attacks)	Lena	Milk Drop	Gold Hill	Average
colorReduce(1)	100%	100%	100%	100%
Wavelet(10)	100%	100%	100%	100%
TemplateRemove(1)	100%	100%	100%	100%
Scale(42)	100%	100%	100%	100%
SampleDownUp(4)	100%	100%	100%	100%
Row_col(49)	100%	100%	100%	100%
MAP(6)	100%	100%	100%	100%
JPEG(12)	100%	100%	100%	100%
Filtering(3)	100%	100%	100%	100%
Bending(2)	100%	100%	100%	100%
Aspectratio(35)	100%	100%	100%	100%
Shearing(14)	100%	100%	86%	95%
ML(7)	86%	86%	86%	86%
Collage(2)	100%	50%	50%	67%
Warping(28)	36%	79%	82%	65%
Crop(28)	82%	50%	54%	62%
Projective(70)	53%	56%	54%	54%
Rotation(21)	95%	10%	48%	51%
Linear(21)	67%	57%	29%	51%
RotationScale(21)	90%	19%	14%	41%
Remodulation(4)	0%	0%	0%	0%
Copy(1)	0%	0%	0%	0%
Average(1146)	82%	73%	73%	76%

Table 3.11: 64 bits Watermark with by-parts interleaving, threshold 51%

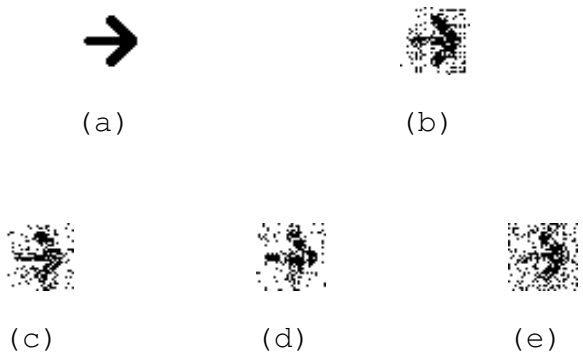


Figure 3.10: *(a)Original Watermark. Extracted Watermarks using (b)Long spreading code, (c)BCH code, (d)LDPC code (e)Convolution code.*

ing is discussed in this chapter. Simulations have shown that it is not favorable to use ECC at the expense of the length of the spreading codes. Under strong attacks long spreading codes are better than short spreading codes with ECC. Another important issue is complexity of the algorithm. ECC also increases the computational complexity of watermarking schemes a lot. So, it is better to use longer spreading codes in CDMA based watermarking schemes instead of ECC.

4 Image Quality Assessment

4.1 Introduction

Objective Image Quality Assessment (IQA) aims to provide an automatic and efficient system to evaluate quality. Based on the existence of source image, IQA can be classified into: Full Reference (FR), Reduced Reference (RR) and No Reference (NR) IQA methods. The FR assumes that the undistorted reference image exists and is fully available. FR method uses the reference image to predict the quality degradation of the distorted medium which eases the process substantially and provides superior quality prediction performance. In the RR method, the reference image is not fully available. Instead, certain features are extracted from the reference image and employed by the quality assessment system as side information to evaluate the quality of the distorted image. The NR method does not have any information about the reference image. The quality assessment is based only on the distorted image [50]. These methods should produce objective scores well correlated with subjective quality scores obtained by the human visual system. The most widely used objective image quality metrics are Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE), although they have been criticized because they do not correlate well with Human perception [6].

Image quality assessment algorithms are needed for three types of application [51]:

1. For optimization purposes, where one maximizes quality at a given cost.

2. For comparative analysis between different alternatives.

3. For quality monitoring in real-time applications.

Considerable volume of research has developed objective image/video quality metrics that incorporate Human Visual System (HVS). However, most of the proposed metrics based on HVS require the existence of the reference image (FR) [52, 53, 54, 55].

Wang et al. proposed a general reduced reference image quality assessment based on statistics computed for natural images in wavelet transform domain [56]. In this method, a generalized Gaussian Density function is used to model the marginal statistics of the coefficients in wavelet sub-bands. Then the parameters of the fitting model are employed as RR features. This method achieves notable success when tested with individual distortion types (JPEG2000 compression, JPEG compression, blurring and white Gaussian noise).

In the last decade, the usage of multimedia contents increased exponentially, due to rapid growth of the Internet. JPEG is one of the most popular and widespread image formats in Internet. JPEG is a lossy compression, it means JPEG compressed images, even after reconstruction, are distorted images with respect to original images. JPEG distortion is inversely proportional to the quality factor used. If JPEG quality factor is formulated one can estimate how much the image is distorted because of the compression. In this chapter, a scheme is proposed that can measure the quality of an image which is compressed using JPEG. This scheme results in the quality factor with which an image was originally compressed, hence give a direct measure for quality. Proposed scheme is very help in the situations where images are only available as bitmaps. It can be used to estimate JPEG quantization ratio with which image was originally compressed. Furthermore, by using proposed algorithm JPEG quantization ratio can also be estimated. To use JPEG coded images, knowledge of quantization ratio is required. Sometimes the the knowledge of quantization ratio is not available, for example the header information is not available. Proposed scheme can estimate the quantization ratio, which was used to compress the original image.

In this chapter, digital Watermarking is used for a new reduced reference image quality assessment scheme for JPEG compression. This scheme uses one block from the original image as a reduced reference

and inserts it over the whole image using digital watermarking. To apply JPEG compression, before DCT transformation, image is divided into 8×8 pixel blocks, the selected block can any of these 8×8 pixel blocks. Although any 8×8 pixel block can be selected, in proposed algorithm, first (top left) 8×8 pixel block from the image is used to implement reduced reference image quality assessment algorithm. This selected block is converted to bits and inserted over the whole image using digital watermarking. For example, in case of gray scale images, every pixel is represented by 8 bits. One 8×8 pixel block is composed of 512 bits, so 512 bits long watermark is embedded in the image, to construct watermarked image. This block can be extracted at the receiver side by watermark extraction process. Then this block is used to generate a set of all possible compressed matrices with all possible quantization matrices. At the same time, the corresponding block from the distorted image is used to calculate the real reconstructed block by applying Discrete Cosine Transformation (DCT) to this block. The main idea for Image Quality Assessment metric is the comparison between the two matrices obtained for same block. It is later checked to which degree, they are similar. The index of matrix which has most similarities yields the quality factor.

Digital watermarking can also be used for estimating the quantization ratio of single JPEG compressed images. One 8×8 pixel block is embedded in the image, same way as described above. Afterwards, JPEG compression is applied on watermarked image, and watermarked image is transmitted. Assume that the receiver does not know the quantization ratio, with which image is compressed. At the receiver side, a set of all possible quantization ratios are applied to the received compressed image. Later, watermark is extracted from all the images, and compared with the selected block. The inserted watermark can be extracted correctly only with the same quantization ratio which was used to compress the original image. Presented algorithm can be considered as a blind JPEG decompression algorithm, as receiver can decompress (reconstruct) the image without knowing the quantization ratio (quality factor) with which the image was originally compressed. Proposed quantization estimation scheme is simple and computationally efficient unlike other complex estimation algorithms [57, 58]. Proposed algorithm can also be used for applications like forgery detection [59, 60]. The knowledge of the quantization ratio used in the JPEG compression is sometimes required at the receiver. It might be used in the cancel-

lation of blocking and ringing artifacts [61, 62]. Furthermore, proposed scheme can simultaneously be used for quantization ratio estimation and usual watermarking applications [63]. The only condition is that the watermark must be a part of the image.

4.2 Overview of the JPEG compression standard

The JPEG compression and decompression is illustrated in Fig. 4.1. First, the original image is split into 8×8 block and DCT is applied. Then the DCT coefficients matrices are divided by the desired quantization matrix. The compression starts with quantization and as a result the output matrix contains many zeros. After that, zig-zag coding is usually used to encode the compressed matrix into bit stream. For decompression, the received compressed image data is decoded and converted back to the compressed matrix. This compressed matrix is multiplied with the same quantization matrix which was used for compression. Finally, the IDCT is applied and the image is reconstructed [64].

The general procedure for JPEG is as follow:

1. The image is split into 8×8 blocks of pixels.

2. Working from left to right, top to bottom, the DCT is applied to each block.

3. Each block is compressed through quantization.

4. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.

5. When desired, the image is reconstructed through decompression, a process that uses the Inverse Discrete Cosine Transform (IDCT).

The DCT equation computes the i, j^{th} entry of the DCT of an image is:

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) * \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) cos \left[\frac{(2x+1)i\pi}{2N} \right] cos \left[\frac{(2y+1)i\pi}{2N} \right] \right] \tag{4.1}$$

$$C(u) = \left\{ \begin{matrix} \frac{1}{\sqrt{2}} & \text{if} \quad u = 0 \\ 1 & \text{otherwise} \end{matrix} \right\} \tag{4.2}$$

where $p(x, y)$ is the x, y^{th} element of the image represented by the matrix p, N is the size of the block that the DCT is done on. The equation calculates one entry (i, j^{th}) of the transformed image from the pixel values of the original image matrix. Inverse DCT is calculated as follows:

$$p(x,y) = \frac{1}{\sqrt{2N}} * \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(i)C(j)D(i,j) cos \left[\frac{(2x+1)i\pi}{2N} \right] cos \left[\frac{(2y+1)i\pi}{2N} \right] \right] \tag{4.3}$$

For the standard 8×8 block that JPEG compression uses, N equals 8 and x and y range from 0 to 7. Therefore, Eq. (4.1) can be simplified as:

$$D(i,j) = \frac{1}{4} C(i)C(j) * \left[\sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) cos \left[\frac{(2x+1)i\pi}{16} \right] cos \left[\frac{(2y+1)i\pi}{16} \right] \right] \tag{4.4}$$

To get the matrix form of Eq. (4.1), the following equation is used:

$$\boldsymbol{T}_{i,j} = \left\{ \begin{matrix} \frac{1}{\sqrt{N}} & \text{if} \quad u = 0 \\ \sqrt{\frac{2}{N}} cos[\frac{(2j+1)i\pi}{2N}] & \text{otherwise} \end{matrix} \right\} \tag{4.5}$$

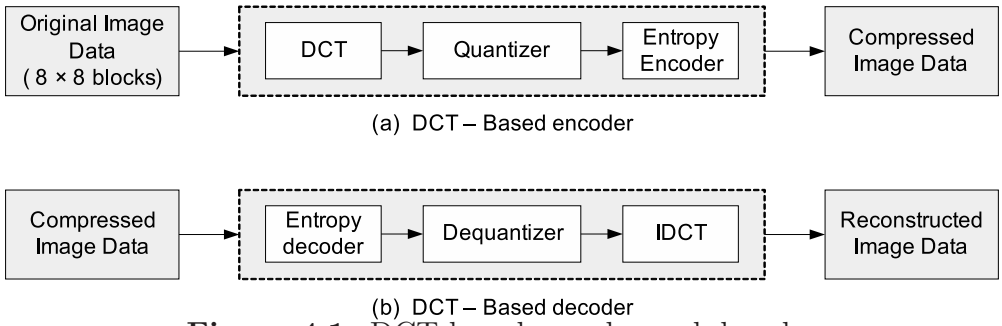


Figure 4.1: DCT based encoder and decoder

Table 4.1: Original and **Reconstructed** image blocks

Original(O)							
154	123	123	123	123	123	123	136
192	180	136	154	154	154	136	110
254	198	154	154	180	154	123	123
239	180	136	180	180	166	123	123
180	154	136	167	166	149	136	136
128	136	123	136	154	180	198	154
123	105	110	149	136	136	180	166
110	136	123	123	123	136	154	136
Reconstructed(N)							
149	134	119	116	121	126	127	128
204	168	140	144	155	150	135	125
253	195	155	166	183	165	131	111
245	185	148	166	184	160	124	107
188	149	132	155	172	159	141	136
132	123	125	143	160	166	168	171
109	119	126	128	139	158	168	166
111	127	127	114	118	141	147	135

4.3 Algorithms

4.3.1 Theoretical Background

Assume that one block ($\boldsymbol{O}$) from the original image is available and the corresponding compressed block ($\boldsymbol{N}$) is taken form the distorted image. The DCT is applied to both as follows:

$$\boldsymbol{R}_C = \boldsymbol{T}(\boldsymbol{N} - 128)\boldsymbol{T}' \tag{4.6}$$

$$\boldsymbol{D} = \boldsymbol{T}(\boldsymbol{O} - 128)\boldsymbol{T}' \tag{4.7}$$

where $\boldsymbol{T}$ is the DCT matrix for 8×8 block. Now using the matrices $\boldsymbol{Q}_j$, which represent the quantization tables, where for $j = 1 : 100$, $\boldsymbol{R}_j$

matrices are constructed as under:

$$\boldsymbol{C}_j = round(\frac{\boldsymbol{D}}{\boldsymbol{Q}_j}) \quad \text{for } j = 1:100 \tag{4.8}$$

$$\boldsymbol{R}_j = \boldsymbol{Q}_j \times \boldsymbol{C}_j \tag{4.9}$$

With the previous knowledge of the comparison ratio (in this case 50), which has produced the compressed image, the $\boldsymbol{R}_j$ matrices are generated for both $\boldsymbol{Q}_{50}$ and $\boldsymbol{Q}_{90}$. Now, comparisons between $\boldsymbol{R}_C$ and different $\boldsymbol{R}_j$ matrices are done as in Table 4.2. The $\boldsymbol{R}_j$ matrix can be considered as a sparse matrix due to the quantization process, while $\boldsymbol{R}_C$ matrix contains many small values due to the rounding process.

There is a relation between the $\boldsymbol{R}_j$ matrices and $\boldsymbol{R}_C$ matrix. This relation is at its maximum, only if the $\boldsymbol{R}_j$ matrix is generated with the same quantization factor that was originally used to produce the compressed image block ($\boldsymbol{R}_j$ for $\boldsymbol{Q}_{50}$).

4.3.2 Image Quality Assessment(IQA) Algorithm

In section 4.3.1, a relation between the original block and compressed version of this block has been concluded. This relation is usually unique for the same quantization factor. If only one 8×8 block is known to receiver, it can compute the quality factor by which received image is compressed. In the proposed algorithm, one 8×8 block is selected and it is embedded in the image using digital watermarking. Later, this image is compressed using whatever quality factor required. At receiver side, this block is extracted using watermark extraction process. This extracted block is used to generate a set of all possible $\boldsymbol{R}_j$ matrices with different quantization factors using equation (4.9). At the same time, the corresponding block ($\boldsymbol{N}$) from the compressed image is used to calculate the $\boldsymbol{R}_C$ matrix by applying Discrete Cosine Transform (DCT) to the $\boldsymbol{N}$ matrix using equation (4.6).

Now, an (8×8) $\boldsymbol{R}_C$ matrix is in one side and at the other side, hundred (8×8) $\boldsymbol{R}_j$ matrices are resulted from the extracted block using the standard quantization matrices ($\boldsymbol{Q}_1$ to $\boldsymbol{Q}_{100}$). The index of the

Table 4.2: Comparison between Original and Reconstructed image blocks

Resulting $\boldsymbol{R}_j$ and $\boldsymbol{R}_C$ and their division

$\boldsymbol{R}_C$

161	43	22	79	26	-1	1	-1
37	106	16	36	28	-2	1	0
-97	-67	17	-49	-40	-1	0	-1
-42	-84	-1	-29	-1	0	0	0
-37	24	-2	2	-1	1	-1	0
-2	2	-2	2	-2	1	-1	1
-1	2	-2	3	-2	1	-1	0
-2	1	-1	1	-1	1	0	0

$\boldsymbol{R}_{50}$ for $\boldsymbol{Q}_{50}$

160	44	20	80	24	0	0	0
36	108	14	38	26	0	0	0
-98	-65	16	-48	-40	0	0	0
-42	-85	0	-29	0	0	0	0
-36	22	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\boldsymbol{R}_{90}$ for $\boldsymbol{Q}_{90}$

162	40	20	72	30	16	-20	-12
30	108	9	32	30	-12	24	0
-93	-60	12	-45	-32	11	0	11
-39	-84	-4	-24	-10	17	0	0
-32	16	-7	-11	14	0	21	0
0	-14	11	0	32	21	0	0
0	0	16	0	-21	-24	0	20
-14	18	0	-20	22	0	0	20

$\alpha_1(\boldsymbol{Q}_{50})$ after removing non-finite and zero elements

1.00	0.97	1.10	0.98	1.08
1.02	0.98		0.94	1.07
0.98	1.03	1.06	1.02	1.00
1.00	0.98		1.00	
1.02	1.09			

$\alpha_1(\boldsymbol{Q}_{90})$ after removing non-finite and zero elements

0.99	1.07	1.10	1.09	
	0.98			0.93
1.04			1.08	
1.07	1.00			

matrix which best matches $\boldsymbol{R}_C$ yields the compression factor. After that, a search for the matrix (or matrices when they are equal) in the set $\boldsymbol{R}_j$ that best matches $\boldsymbol{R}_C$ matrix is processed. This can be done using different methods. In Fig. 4.2 an algorithm is presented for searching the best matching matrix. This algorithm is illustrated using two filtering steps.

The result of element wise division between the $\boldsymbol{R}_C$ matrix and every matrix of the set $\boldsymbol{R}_j$ is calculated and stored again in a new set $\alpha(\boldsymbol{Q}_j)$

$$\alpha(\boldsymbol{Q}_j) = \boldsymbol{R}_C \,./\, \boldsymbol{R}_j \quad , \text{for } j = 1:100, \tag{4.10}$$

At this point, the resulting matrices contain some non-finite elements (due to the division of zero). A replacement for the non-finite elements by zeros is processed. The resulting new set is stored again in $\alpha_1(\boldsymbol{Q}_j)$. Later on, $\alpha_1(\boldsymbol{Q}_j)$ elements are filtered between $(0.90 \leq \alpha_{xy} \leq 1.10)$ here, α_{xy} are elements of matrix $\alpha_1(\boldsymbol{Q}_j)$. Then the index(s) of the matrix (matrices) that have the maximum number of elements represent(s) one possible candidate for the distortion factors.

If the matching yields only one index (one quality factor), the first filtering process would be sufficient. Otherwise, a second filtering step is processed further. For multiple indexes, the $\boldsymbol{R}_C$ matrix is filtered by the replacement of the elements $(-1 \leq r_{xy} \leq 1)$ with zeros (this range is obtained after applying DCT for different images and observes the DCT coefficients values), here r_{xy} are elements of matrix $\boldsymbol{R}_{C1}$. Then the new $\boldsymbol{R}_{C1}$ matrix and the subset of $\boldsymbol{R}_{j1}$ matrices is converted to column vectors. Now, an element-wise comparison between column vectors is done. Finally, the index of the column with best matching represents the Quality factor.

4.3.3 Blind Quantization Ratio Estimation Algorithm

The above proposed algorithm can also be used to develop an algorithm for blind quantization estimation. If only one original 8×8 pixel block is known to the receiver, it can compute the quality factor by which

received image is compressed. In the proposed algorithm, one 8×8 pixel block is selected and it is embedded in the image using digital watermarking, as done in above quality assessment algorithm. Later, this image is compressed using the required quality factor. The resulting image is encoded into bits stream, transmitted and decoded back at the receiver side. If the receiver does not know the quality factor with which image is compressed, it applies a set of all quantization tables to the received compressed image. As a result, the corresponding set of watermarked distorted images ($\boldsymbol{Iwd}_n$ where $n \in \{1, 2, 3, 4, \ldots, 100\}$) has been generated for each of the received images. From every generated image two blocks have been extracted. One of these blocks is the extracted watermark $\boldsymbol{O}_n$ (where $n \in \{1, 2, 3, 4, \ldots, 100\}$), while the other block is the corresponding selected block from the watermarked distorted image $\boldsymbol{N}_n$.

The proposed algorithm (Fig. 4.3) is based on a comparison between every pair of blocks for each of the generated watermarked distorted image. One point must be mentioned here, that the inserted watermark can be extracted correctly *only* if the same quantization ratio, which was used to compress the original image, is used to decompress the received data.

For n-th image: DCT is applied on extracted watermark block $\boldsymbol{D}_n = \boldsymbol{T}(\boldsymbol{O}_n - 128)\boldsymbol{T}'$ and it is quantized $\boldsymbol{C}_n = round(\frac{\boldsymbol{D}}{\boldsymbol{Q}_n})$ to get $\boldsymbol{R}_n = \boldsymbol{Q}_n \times \boldsymbol{C}_n$. Now DCT is applied on the corresponding selected block to get $\boldsymbol{R}_{Cn} = \boldsymbol{T}(\boldsymbol{N}_n - 128)\boldsymbol{T}'$. After that, the ratio between $\boldsymbol{R}_{Cn}$ and $\boldsymbol{R}_n$ is calculated using $\alpha(\boldsymbol{Q}_n) = \boldsymbol{R}_{Cn} \,./\, \boldsymbol{R}_n$ and stored as a matrix. The $\alpha(\boldsymbol{Q}_n)$ matrix is filtered between $(0.90 \leq \alpha_{xy} \leq 1.10)$ and stored as $\alpha_1(\boldsymbol{Q}_n)$. $\alpha_1(\boldsymbol{Q}_n)$ is calculated for every image of set $\boldsymbol{Iwd}_n$. Only one image from the set of watermarked distorted images $\boldsymbol{Iwd}_n$ is correct, which has been decompressed with the same quantization ratio (which was used to compress it). This image is the one which has the maximum number of elements in $\alpha_1(\boldsymbol{Q}_n)$.

4.4 Information Embedding System

To embed a selected block into the image, a dithered uniform scalar quantization watermarking in wavelet transform domain is used as in [65]. This watermarking technique is a simple case of a class of quantization-index-modulation information embedding techniques which allow for blind decoding and achieve a good trade off between data hiding rate and robustness. A five-scale separable wavelet transform is used to decompose the reference image into 16 sub-bands, including the horizontal, vertical and diagonal sub-bands at each scale and a low frequency residual band. For inserting one bit of information $m \in \{0, 1\}$ into a wavelet coefficient c, the coefficient is altered according to the following rule:

$$c_q = Q(c + d(m)) \equiv Q^m(c), \tag{4.11}$$

where c_q is the altered coefficient, $Q(.)$ is a base quantization operator with step size Δ and $d(m)$ is dithering operator defined as:

$$d(m) = \begin{cases} -\Delta/4 & \text{if } \ m = 0 \\ \Delta/4 & \text{if } \ m = 1 \end{cases} \tag{4.12}$$

At the receiver side, a distorted coefficient c_d is obtained and used to estimate the embedded bit based on the minimum distance criterion:

$$\widehat{m}(c_d) = \arg \min_{m \in \{0,1\}} \|c_d - Q^m(c_d))\| \tag{4.13}$$

The hidden messages are embedded into the horizontal, vertical and diagonal sub-bands at the fifth scale of the wavelet decomposition. Some other watermarking algorithms can also be used to embed the selected block.

4.5 Experimental Results

4.5.1 Image Quality Assessment

This IQA metric has been tested on 28 512×512 gray scale images (the original images are chosen from the LIVE database [66]), and after the insertion of one 8×8 block as a watermark using [56], each of the resulting images has been distorted with different distortion factors (i.e. ranging from 51 to 100). As a result, a database consists of 1400 distorted images has been created. Due to watermarking limitation image under 50% quality factor are not tested. Using watermarking scheme presented in [65] it is difficult to extract 512 bit long watermark correctly under 50% quality factor.

Table 4.3 shows the experimental results for this algorithm with different filtering ranges and considering different accuracies of the number of wrong Image Quality Metrics (IQMs). The results use these two criterion $(0.90 \leq \alpha_{xy} \leq 1.10)$ and $(-1 \leq r_{xy} \leq 1)$, which are the best among the others. Only 24 records out of 1400 give wrong prediction with an accuracy of ± 3 (as a percentage less than 2%). Table 4.4 is the same as the previous table but in percentage. Also, the correlation between $\boldsymbol{R}_C$ and $\boldsymbol{R}_j$ is tested at the same dataset (by using the matlab command $corr2(\boldsymbol{R}_C, \boldsymbol{R}_j)$). In this case 62 records out of 1400 give wrong prediction with an accuracy of ± 3.

Table 4.3: Number of images with errors for different IQMs

Number of Errors using different Accuracy				
	± 3	± 2	± 1	0
$\boldsymbol{IQA}_1(\boldsymbol{R}_C(-1 \leq R_C \leq 1) = 0)$	62	81	155	340
$\boldsymbol{IQA}_2((0.90 \leq \alpha_{xy} \leq 1.15)$ &$(\boldsymbol{R}_C(-1 \leq r_{xy} \leq 1) = 0))$	32	59	143	367
$\boldsymbol{IQA}_2((0.90 \leq \alpha_{xy} \leq 1.10)$ &$(\boldsymbol{R}_C(-1 \leq r_{xy} \leq 1) = 0))$	*24*	*39*	*109*	*310*

$\boldsymbol{IQA}_1$ is computed using correlation function between the $\boldsymbol{R}_{C1}$ (which equals to $\boldsymbol{R}_C$ from the compressed image after setting all the elements that ranging from $-1 \leq r_{xy} \leq 1$ to zero) with $\boldsymbol{R}_{j1}$ and selecting the best match for the $\boldsymbol{R}_{C1}$. $\boldsymbol{IQA}_2$ shows experimental results using proposed

Table 4.4: Ratio of images with errors for different IQMs

Number of Errors using different Accuracy	±3	±2	±1	0
$\boldsymbol{IQA}_1(\boldsymbol{R}_C(-1 \leq R_C \leq 1) = 0)$	4%	6%	11%	24%
$\boldsymbol{IQA}_2((0.90 \leq \alpha_{xy} \leq 1.15)$ &$(\boldsymbol{R}_C(-1 \leq r_{xy} \leq 1) = 0))$	2%	4%	10%	25%
$\boldsymbol{IQA}_2((0.90 \leq \alpha_{xy} \leq 1.10)$ &$(\boldsymbol{R}_C(-1 \leq r_{xy} \leq 1) = 0))$	*2%*	*3%*	*8%*	*21%*

algorithm.

4.5.2 Quantization Ratio Estimation

This algorithm has been tested on 28, 512×512 gray scale images (the original images are chosen from the LIVE database [66]), and after the insertion of one 8×8 block as a watermark using [65], each of the resulting images has been compressed with different quantization ratio (i.e. $\boldsymbol{Q}_n$ where $n \in \{25, 30, 35 \ldots, 95, 100\}$). As a result, a database consists of 448 distorted images has been created. Using proposed algorithm, quantization ratio (quality factor) is estimated. Out of 448 images only 20 estimates are wrong. Table 4.5 shows the experimental results, wrong estimations are highlighted. When $\alpha_1(\boldsymbol{Q}_n)$ is empty the algorithm returns No Result (NR). This is happened when compression ratio is very high (quality factor is very small). At very small quality factors image is distorted so much that it actually destroys the watermark. When watermark is lost completely, obviously algorithm can not work. There are a few wrong estimations e.g. for parrots image, when image is compressed with a quality factor of 70%, proposed algorithm estimates the quality factor as 30%, this happens when most significant bits are detected wrong, although BER in extracted watermark is not high. When significant bits are detected wrong the reconstructed 8×8 block becomes very different from the original block which was used as a watermark.

This algorithm is also tested for quality factors $\leq 20\%$, but as watermark is completely destroyed, almost in all the cases algorithm returns

no result (NR).

4.6 Conclusions

A new metric for Reduced Reference JPEG Image Quality Assessment has been proposed. An 8×8 block from the original image is transmitted to the receiver as a watermark. This block is extracted at receiver side and by using extracted block and corresponding compressed block quality of a JPEG image is assessed. Performances are tested on 28-images, each of them distorted with fifty different quality factors (a total of 1400 images). In this test only 24 images yield quality scores with wrong prediction. This algorithm works within an accuracy of ± 3 and provides a direct measure for the quality. The proposed Image quality metric is developed for JPEG distortion. As proposed scheme is based on comparison between original block and distorted block, similar schemes can be developed for JPEG2000 or Gaussian Blur. To increase robustness of this metric, different watermarking technique can be used.

In this chapter, an algorithm is also proposed to decompress JPEG images at the receiver side blindly. The main idea is that the inserted watermark can be extracted correctly only with the same quantization ratio which was used to compress the original image. This algorithm is tested against 448 images. The percentage of wrong estimation is almost 4%. To increase the robustness of this algorithm, different watermarking technique can be used. In future, the proposed algorithm can also be used for image forgery detection. If the image is tempered the hidden block (watermark) may not be extracted correctly.

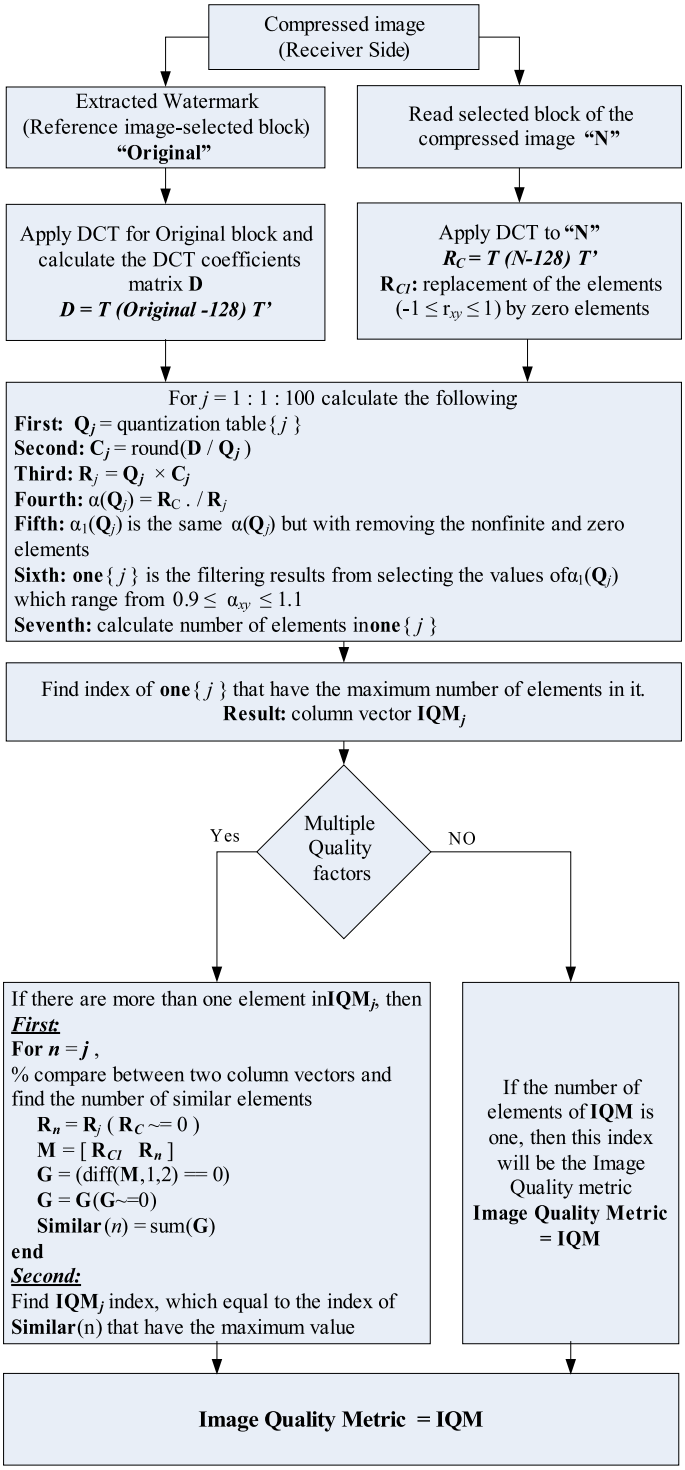


Figure 4.2: JPEG Image Quality Assessment Algorithm

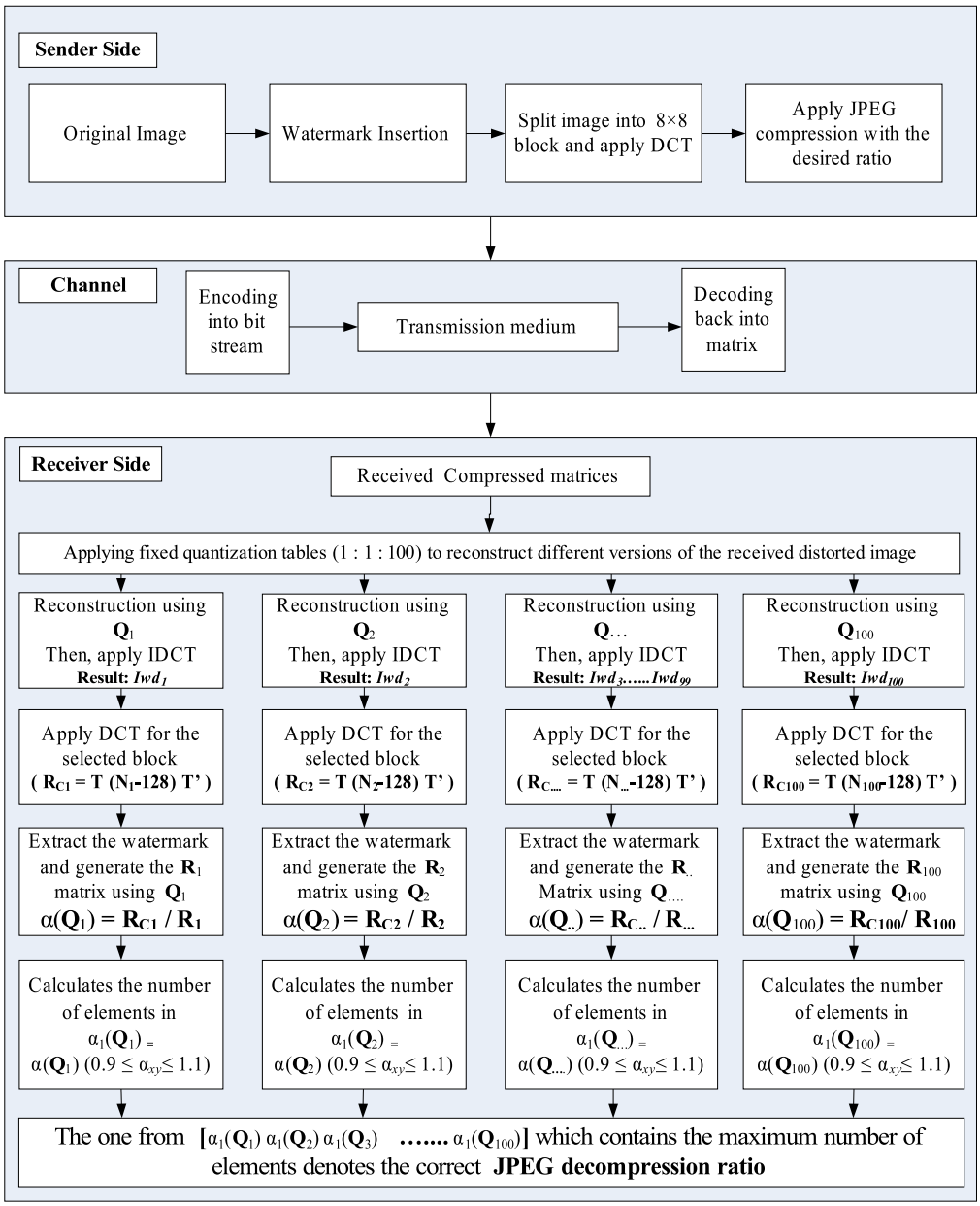


Figure 4.3: Blind JPEG Decompression Algorithm

Table 4.5: Estimated quantization ratios (quality factors) using proposed algorithm

Image name	**Quality factor**												
	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
coinsinfountain	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
ocean	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
statue	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
dancers	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
paintedhouse	**40%**	30%	35%	40%	45%	50%	**65%**	60%	**50%**	70%	75%	80%	85%	90%	95%	100%
stream	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
bikes	**NR**	**NR**	**NR**	**NR**	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
flowersonih35	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
parrots	**NR**	**NR**	**85%**	**NR**	45%	50%	55%	60%	65%	**30%**	75%	80%	85%	90%	95%	100%
studentsculpture	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
building2	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
plane	25%	30%	**50%**	40%	45%	50%	55%	**70%**	65%	70%	75%	80%	85%	90%	95%	100%
woman	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
house	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
rapids	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
womanhat	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
caps	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
lighthouse	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
sailing1	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
carnivaldolls	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
lighthouse2	**NR**	**NR**	**NR**	**NR**	45%	50%	55%	60%	65%	**55%**	75%	80%	85%	90%	95%	100%
sailing2	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
cemetry	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
manfishing	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
sailing3	**NR**	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
churchandcapitol	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
monarch	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
sailing4	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%

5 Channel Equalization Using Watermark as a Training Sequence

5.1 Introduction

Intersymbol Interference (ISI) is a very common type of distortion which a signal undergoes during transmission [25]. Trained equalization is a technique widely used to cope with ISI. A training sequence is sent over the channel and with the help of received and already known training sequence channel is equalized. Traditional trained equalization scheme works with simple algorithms like Least Mean Squares (LMS)[67]. Most crucial drawback of traditional trained equalization is that it consumes extra bandwidth. To save the bandwidth, some blind equalization techniques, which do not require training sequences, are used to equalize channels. These blind equalization schemes are usually based on computationally complex algorithms like Constant Modulus Algorithm (CMA)[68]. So the cost of saving bandwidth is increased computational complexity.

Digital watermarking can be used beyond simple security tasks if some meaningful watermark is embedded in host data. Tracing watermarks are used for blind quality assessment for multimedia communication in [69]. A multipurpose public-key cryptosystem based on image watermarking is developed in [70]. End-to-end QoS provision and control in wireless communication by means of digital watermarking is presented in [71]. Digital watermarking is used for dynamic and adaptive equalization in [72] and [73] respectively. However in [72] and [73] receiver should know the watermark (training sequence) in advance, which is not required in proposed scheme.

In comparison to blind equalization, traditional trained equalization

is known to be computationally efficient. But the most crucial drawback of trained equalization is that it consumes extra bandwidth. Another disadvantage is that the receiver must have prior information of the original training sequence. There are some blind equalization algorithms, where training sequences are not used and data can still be equalized [68], [74]. To design a blind equalization algorithm convergence of the algorithm and computational complexity are two important issues. Constant Modulus Algorithm (CMA) [68] is one of well known algorithm used for blind equalization. Disadvantage for CMA is its slow convergence. Therefore, researchers tried to lessen the computational complexity of CMA and tried to make convergence faster [75], [76].

In this work, a completely different approach to blind equalization is taken by using digital watermarking. The proposed method does not require training sequences to be known in advance and works with standard trained equalization algorithms (e.g. LMS) at low computational complexity. Furthermore, proposed scheme can simultaneously be used for blind equalization and usual watermarking applications.

Proposed schemes can be looked from two different viewpoints. First, how watermarking can be used for blind equalization (the receiver is not required to know the training sequence). Second, how digital watermarking can be used to correct errors in transmitted images. Additionally the proposed algorithms can simultaneously be used for blind equalization and usual watermarking applications. In this chapter, three watermarking based equalization algorithms are presented, for three different scenarios:

1. when channel specific training sequence or specific watermark is required,

2. when watermark/training sequence can be any sequence (can be a part of data),

3. when original data required after equalization.

In the first algorithm (algorithm I), training sequence is embedded in the data in the form of watermark. This training sequence is also sent over the channel. On receiving side, the watermark is extracted

and with the help of received training sequence and extracted watermark channel can be equalized. In this algorithm receiver does not required to know the training sequence in advance. Consider the following scenario: An image (data) distributor distributes watermarked images (data) containing different watermarks to several clients. Distributor also sends the watermark along with the watermarked image (data) through the channel. On the receiving sides, receivers only have a watermark extractor (all the receivers use the same extracting software). Therefore, the receivers can extract the watermark and use it as a reference training sequence. The watermark which distributor has sent through the channel can be treated as received training sequence. Now, the channel can be estimated and hence received, erroneous, watermarked images can be corrected.

In the second algorithm (algorithm II), a chunk of data, which is a part of the data actually to be transmitted through the channel, is used as a watermark. This chunk is embedded in the data as a watermark. With the help of received chunk and extracted chunk (extracted watermark) the channel can be equalized blindly.

In the third algorithm (algorithm III), reversible watermarking is used to recover the original data after equalization. Watermarking algorithm used for these blind equalization schemes, must have the following two qualities:

- Robust (to withstand distortions due to transmission).
- Carry enough payload (which can be used as training sequence).

CDMA based spread spectrum watermarking scheme fulfills the above requirements. The main advantage of spread spectrum watermarking is that each watermark bit is embedded in a number of pixels. Because of that it is proved to be robust in transmission. By using multiple orthogonal spreading codes, CDMA based watermarking scheme can carry more payload (large watermark). Normalized Least Mean Square (NLMS) algorithm [67] is used in this scheme for equalization. Other equalization algorithms e.g. Least Mean Square (LMS) and Recursive Least Square (RLS) can also be used.

5.2 Watermarking-Based Blind Equalization

Fig. 5.1 shows how trained equalization works. In order to equalize the data receiver must know the reference training sequence in advance. Error $e(n)$ is calculated with the help of received training sequence and reference training sequence. Now by minimizing $e(n)$, the inverse of the channel is estimated and weights (taps) $\mathbf{w}(n)$ are updated. The received data can be equalized by using $\mathbf{w}(n)$.

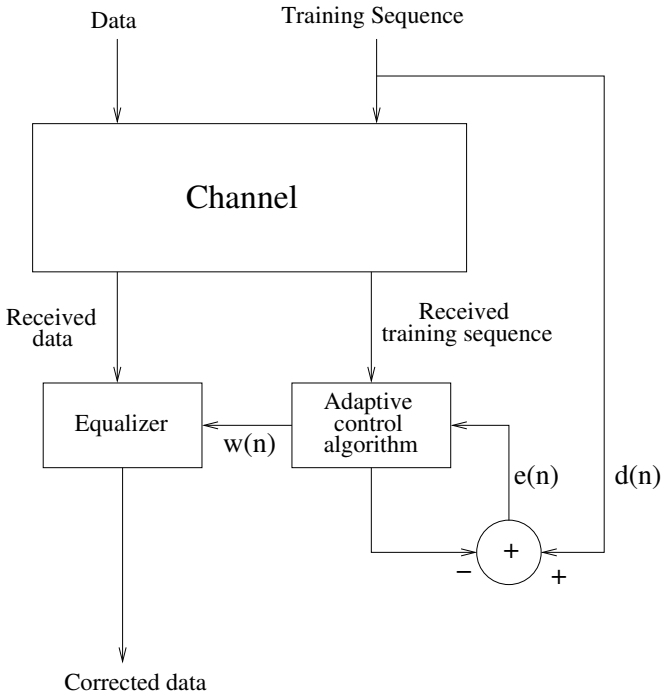


Figure 5.1: *Traditional trained equalization technique.*

Fig. 5.2 shows algorithm I for blind equalization, where a training sequence is hidden (or superimposed) in the data by using digital watermarking. After transmission, if this hidden training sequence is extracted from the received data without much errors, the extracted training sequence can be used as reference training sequence (which is supposed to be provided by the sender).

Fig. 5.3 shows algorithm II for blind equalization, where a selected chunk of data is hidden (or superimposed) in the entire stream of data (Fig. 5.4). After transmission, this hidden chunk of data is extracted from the received data. This extracted chunk can be used as a reference training sequence and the received chunk can be used as a received

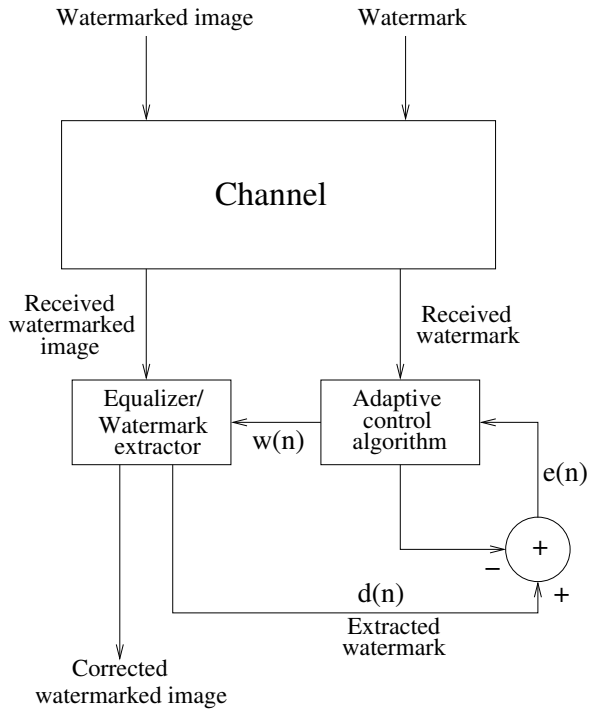


Figure 5.2: *Algorithm I for blind equalization.*

training sequence. Thus $e(n)$ can be calculated and adaptive control algorithm can be used to update $\mathbf{w}(n)$. Hence received data can be equalized blindly.

Algorithm III is shown in fig. 5.5. This algorithm is similar to algorithm II, the difference is that reversible watermarking is used to retrieve original image after equalization.

For equalization, in algorithm I, II and III, Normalized LMS algorithm is used. This algorithm works as follows [67]:

$$\begin{aligned} \textit{Parameter}: \quad M &= \text{number of taps} \\ a &= \text{positive constant} \\ \tilde{\mu} &= \text{adaption constant (step-size parameter)} \\ 0 < \tilde{\mu} &< 2 \end{aligned}$$

Initialization. If prior knowledge on the tap-weight vector $\hat{\mathbf{w}}(n)$ is available, use it to select an appropriate value for $\hat{\mathbf{w}}(0)$. Otherwise, set $\hat{\mathbf{w}}(0) = \mathbf{0}$.

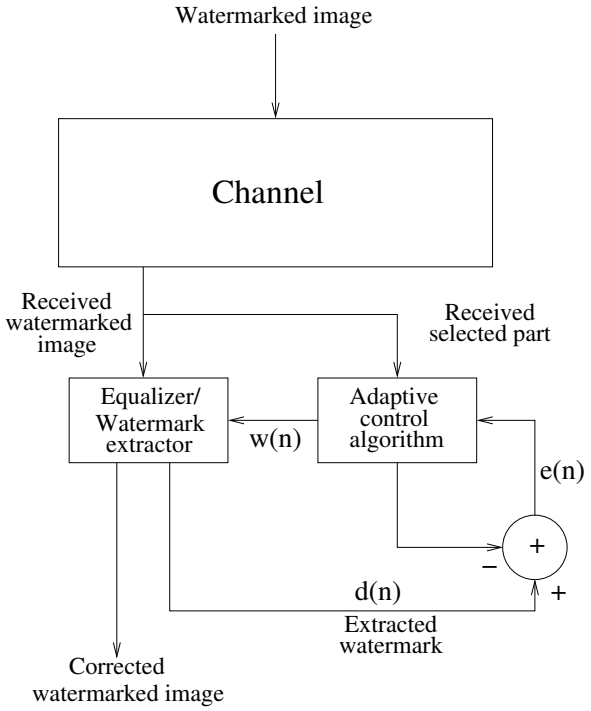


Figure 5.3: *Algorithm II for blind equalization.*

Data

$$(a)\text{Given}: \quad \begin{aligned} \mathbf{u}(n) &: \text{M-by-1 tap-input vector at time n} \\ d(n) &: \text{desired response at time n} \end{aligned}$$

$$(b)\text{To be computed}: \quad \hat{\mathbf{w}}(n+1) = \text{estimate of tap-weight vector at time n+1}$$

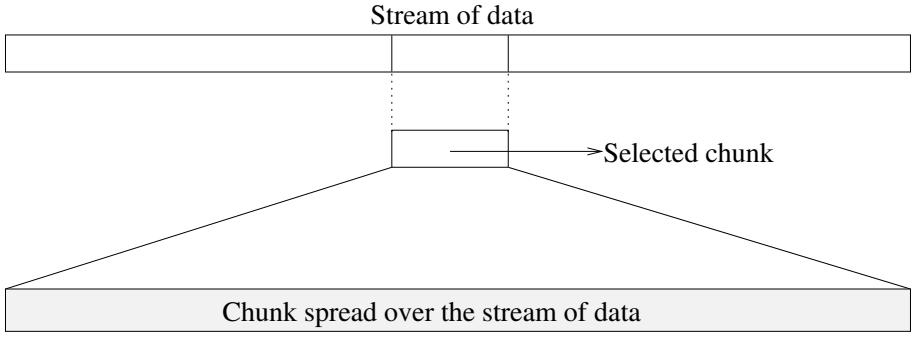


Figure 5.4: *Selecting a chunk of data and hiding it in the stream of data.*

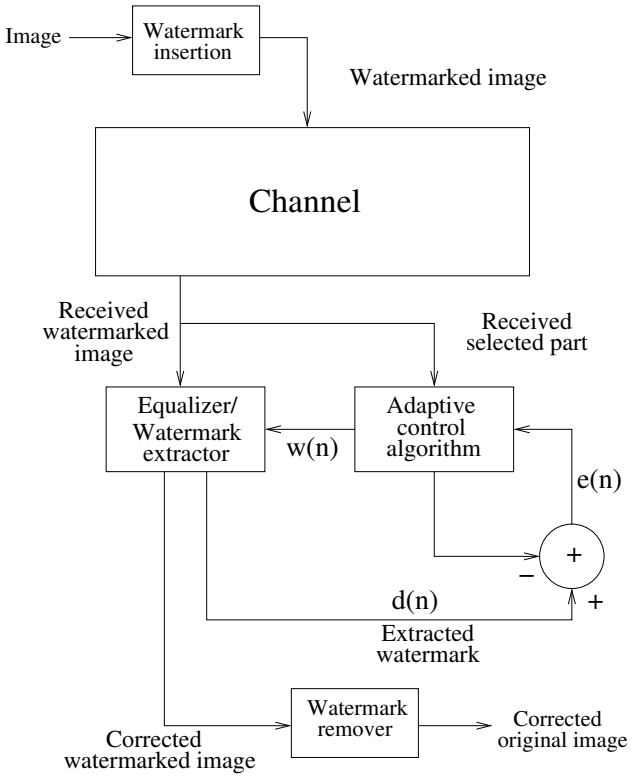


Figure 5.5: *Algorithm III for blind equalization.*

Computation : $n = 0, 1, 2, \ldots$

$$\begin{aligned} e(n) &= d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \\ \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{a + \parallel \mathbf{u}(n) \parallel^2}\mathbf{u}(n)e^*(n) \end{aligned} \tag{5.1}$$

Normalized LMS is a modified version of LMS algorithm. LMS algorithm also works fine in proposed blind equalization method. In LMS algorithm weights are updated according to the following equation [67]:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n) \tag{5.2}$$

where:

$$\begin{aligned} \mu &= \text{step-size parameter} \\ 0 < \mu &< \frac{2}{\text{tap-input power}} \\ \text{tap-input power} &= \sum_{k=0}^{M-1} E[|u(n-k)|^2] \end{aligned}$$

However simulations have shown that Normalized LMS algorithm works slightly better than LMS algorithm. LMS algorithm experiences a gradient noise amplification problem when $\mathbf{u}(n)$ is large, because the correction applied to the tap-weight vector $\hat{\mathbf{w}}(n)$ at iteration $n+1$ is directly proportional to the tap-input vector $\mathbf{u}(n)$. In normalized LMS algorithm correction applied to tap-weight vector $\hat{\mathbf{w}}(n)$ at iteration $n+1$ is normalized with respect to squared Euclidean norm of tap-input vector $\mathbf{u}(n)$ at iteration n.

5.2.1 Implementation Problem

A very crucial problem arose while implementing proposed scheme using digital watermarking. A part of an image is selected, and later it is spread over the whole image forming watermarked image. Obviously, watermarked image is slightly different from the original image. When this watermarked image is transmitted, channel can not be equalized. Because, the extracted watermark is a part of the original image and received part is from watermarked image. Hence both of them are different and the channel can not be equalized using extracted watermark and received part of the watermarked image.

One of the simplest solutions to this problem is to hide the selected part in the remaining image (other than selected part). However, this simple solution causes problems for usual watermarking applications, like copyrights protection etc., and for blind equalization. From general watermarking applications point of view, it is not good to leave a reference to the original image. Watermark should be spread over the whole image. If reference to the original image is present in the watermarked image, some denoising algorithm can detect watermark as noise. So watermark can be removed easily. Therefore, this scheme can not be used for usual watermarking applications.

This problem is solved by watermarking selected portion with some dummy bits first and then hide this dummy watermarked selected portion into remaining image (Fig. 5.6). Now this watermarked image has same level of noise over whole image and has no reference to the original image. Therefore, this watermarked image can simultaneously be used for blind equalization as well as for usual watermarking applications.

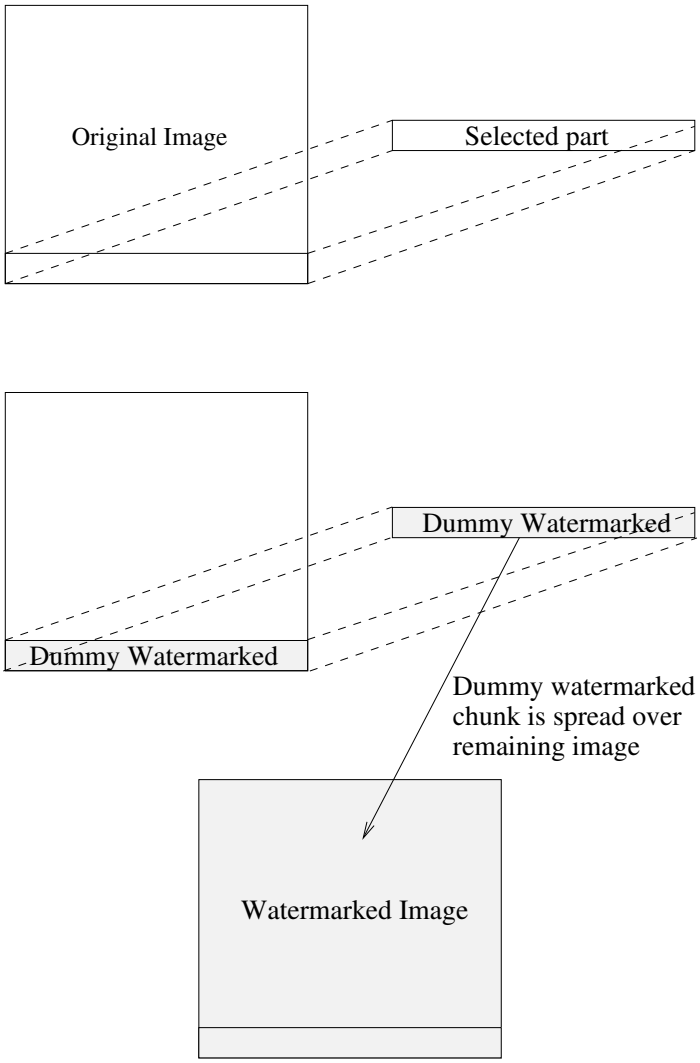


Figure 5.6: *Forming a watermarked image, considering implementation problem.*

5.2.2 Possible Security Issue

As watermark is a part of the watermarked image in algorithm II and III, it can be considered as a possible security issue. By using proposed scheme, watermark can be any part of watermarked image, which is very difficult to locate for an evedropper. Furthermore, because of proposed implementation scheme noise level (watermark intensity) is same over the whole image. So it is difficult to distinguish between selected part, watermark, and remaining watermarked image.

5.3 Experimental Results

In all the experiments 512×512 (8 bits/pixel, gray scale) images (Lena Fig. 5.7(a) and Milk drop Fig. 5.7(b)) are used. 512 bits long watermark is hidden in 3rd, 4th, 5th and 6th rows of 8×8 blocks of DCT coefficients. Two spreading codes, each of length 512, are used to spread the watermark bits. All the images are watermarked at 40db (here $\alpha = 0.01$). Using MATLAB, these images are transmitted over four different channels:

channel 1:	{0.986, 0.845, 0.237, 0.123+0.310i}
channel 2:	{0.986, 0.845, 0.537, 0.323, 0.123}
channel 3:	Frequency-flat (single path) Rayleigh fading channel, sample time 1×10^{-5} and maximum Doppler shift 0.09 Hz.
channel 4:	Frequency-flat (single path) Rician fading channel, sample time 1×10^{-5}, maximum Doppler shift 0.99 Hz and Rician factor equal to 1.

Elements of the vectors shown above, for channel 1 and 2, represent channel coefficients. Channels are simulated by arranging the above shown vectors into Teoplitz matrices.

(a) Lena image. (b) Milk drop.

Figure 5.7: *Original Lena and Milk drop images.*

Table 5.1: BER in extracted watermarks using algorithm I

channels	Received images		Equalized images	
	Lena %	Milk drop %	Lena %	Milk drop %
1	8.2	8.59	0	0
2	16.8	16.8	0	0
3	0.39	10.16	0	0
4	2.54	10.55	0	0

Table 5.2: Lena received and equalized using algorithm I

channels	PSNR		Erroneous bits	
	Received dB	Equalized dB	Received %	Equalized %
1	29.0929	77.5595	8.94	0.01
2	28.9833	69.6869	12.2	0.09
3	22.0178	Infinite	33.58	0.00
4	13.8834	42.9292	9.29	0.00

5.3.1 Algorithm I

First of all, training sequence is embedded in an image as watermark. 512 bits long training sequence (watermark) is sent over the channel followed by the watermarked image. On the receiving side, watermark is extracted from the watermarked image. The extracted watermark has a few erroneous bits (table 5.1) but the BER is good enough to use it as a reference training sequence. Normalized LMS with 8 weights, $\tilde{\mu} = 0.01$ and $\hat{\mathbf{w}}(0) = \mathbf{0}$ are used. The calculated weights $\hat{w}(n)$ are used to equalize the received image. Received and corresponding equalized Lena images are shown in fig. 5.8. Errors and PSNR values of received and equalized images for both Lena and Milkdrop are shown in table 5.2 and 5.3, respectively. Here, second and third columns show PSNR value of received and equalized images with respect to watermarked images. Fourth and fifth columns show percentage of erroneous bits in received and equalized images. Column 4 and 5 of Table 5.1 show the BER in the watermarks extracted from equalized images. These watermarks are error free and easily be used for usual watermarking applications.

Table 5.3: Milk drop received and equalized using algorithm I

channels	PSNR		Erroneous bits	
	Received dB	Equalized dB	Received %	Equalized %
1	29.2870	74.1472	8.51	0.03
2	29.1856	92.3162	11.65	0.00
3	23.0038	Infinite	33.58	0.00
4	13.8618	21.8373	9.29	1.27

5.3.2 Algorithm II

First eight rows of an image are watermarked with eight dummy bits. First 504 bits are selected from the dummy watermarked part and hidden in remaining rows of the image. These watermarked images (watermarked at 40db) are transmitted over the same four channels, mentioned above.

On the receiving side, watermark is extracted from the watermarked image. The watermark extracted from the received image, which forms the training sequence, has a few erroneous bits (column 2 and 3 of Table 5.4). The received selected part is treated as the received training sequence. $\hat{\mathbf{w}}(n)$ are calculated using Normalized LMS with $M = 8$ weights, step size $\tilde{\mu} = 0.01$ and $\hat{\mathbf{w}}(0) = \mathbf{0}$. The calculated weights $\hat{\mathbf{w}}(n)$ are used to equalize the received image. Received and corresponding equalized Lena and Milk drop images are shown in fig. 5.9 and fig. 5.10. Errors and PSNR values of received and corresponding equalized images are shown in Table 5.5 and 5.6. Here, second and third columns show PSNR value of received and equalized images with respect to watermarked images. Fourth and fifth columns show percentage of erroneous bits in received and equalized images. Column 4 and 5 of Table 5.4 show the BER in the watermarks extracted from equalized images.

Table 5.4: BER in extracted watermarks using algorithm II

channels	Received images		Equalized images	
	Lena %	Milk drop %	Lena %	Milk drop %
1	4.76	8.93	0	0
2	6.75	10.52	0	0
3	0.2	4.37	0	0
4	24.40	17.26	0	0

Table 5.5: Lena received and equalized using algorithm II

channels	PSNR		Erroneous bits	
	Received dB	Equalized dB	Received %	Equalized %
1	26.8461	87.1311	9.67	0.00
2	26.8677	86.5184	13.04	0.00
3	22.5406	102.3162	33.60	0.00
4	12.3285	88.8920	8.06	0.00

5.3.3 Algorithm III

504 bits long part is selected and is hidden as watermark in 3rd, 4th, 5th and 6th rows of 8×8 blocks of DCT coefficients of the remaining image. Two spreading codes, each of length 512 are used to spread each bit. These watermarked images (watermarked at 40db) are transmitted over same four channels mentioned above.

Table 5.6: Milk drop received and equalized using algorithm II

channels	PSNR		Erroneous bits	
	Received dB	Equalized dB	Received %	Equalized %
1	27.2704	83.6830	4.56	0.00
2	27.3037	74.7575	4.76	0.03
3	22.6232	102.3162	10.32	0.00
4	13.2914	102.3162	14.88	0.00

On the receiving side, watermark is extracted from the watermarked image. The watermark extracted from the received image, which forms the training sequence, has a few erroneous bits (column 2 and 3 of Table 5.7). However, the BER is good enough to use it as the reference training sequence. This is similar to decision feedback equalization, where there is also no exact training sequence available. The received selected part is treated as the received training sequence. $\hat{\mathbf{w}}(n)$ are calculated using Normalized LMS with $M = 8$ weights, step size $\tilde{\mu} = 0.01$ and $\hat{\mathbf{w}}(0) = \mathbf{0}$. The calculated weights $\hat{\mathbf{w}}(n)$ are used to equalize the received image. To get the original image back, watermark is removed from equalized images. Received and corresponding equalized Lena and Milk drop images are shown in fig. 5.11 and fig. 5.12. Respective PSNR values are shown in Table 5.8. Here, PSNR value of received and equalized images are with respect to original images.

Table 5.7: BER in extracted watermarks using Algorithm III

channels	Received images	
	Lena %	Milk drop %
1	6.15	4.37
2	8.13	5.75
3	1.19	10.12
4	17.06	17.06

Table 5.8: Images received and equalized using Algorithm III

channels	Lena		Milk drop	
	Received dB	Equalized dB	Received dB	Equalized dB
1	27.1290	42.4570	27.5911	45.7360
2	27.1481	42.3606	27.6239	44.0344
3	22.5211	47.9194	22.6699	60.4883
4	13.5889	60.5992	13.3220	57.5424

5.4 Conclusions

How watermarking can be used to equalize transmission channels only using the received data is discussed in this paper. Simulations have shown that proposed schemes can correct almost all the errors from received watermarked images. An important advantage of this scheme is that it works at the complexity of traditional trained equalization methods unlike other very complex blind equalization methods. Proposed schemes can simultaneously be used for blind equalization as well as for usual watermarking applications.

(a) Lena image.

(b) Watermarked Lena.

(c) Received through channel 1.

(d) Equalized image, received through channel 1.

(e) Received through channel 2.

(f) Equalized image, received through channel 2.

(g) Received through channel 3.

(h) Equalized image, received through channel 3.

(i) Received through channel 4.

(j) Equalized image, received through channel 4.

Figure 5.8: *Lena received and equalized using Algorithm I.*

(a) Lena image.

(b) Watermarked Lena.

(c) Received through channel 1.

(d) Equalized image, received through channel 1.

(e) Received through channel 2.

(f) Equalized image, received through channel 2.

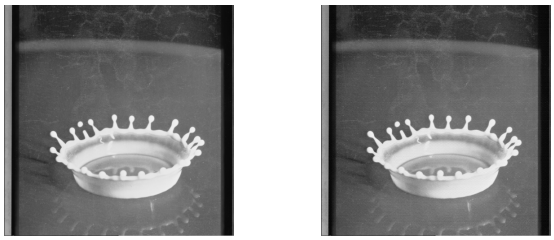
(g) Received through channel 3.

(h) Equalized image, received through channel 3.

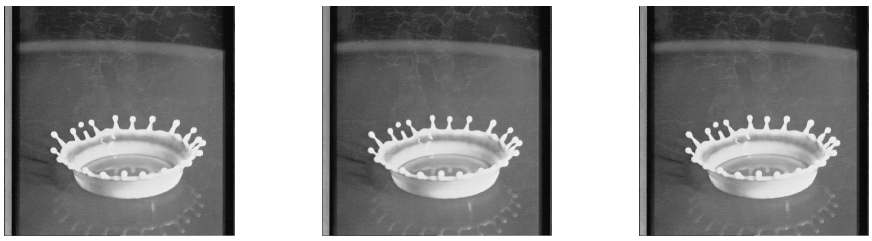
(i) Received through channel 4.

(j) Equalized image, received through channel 4.

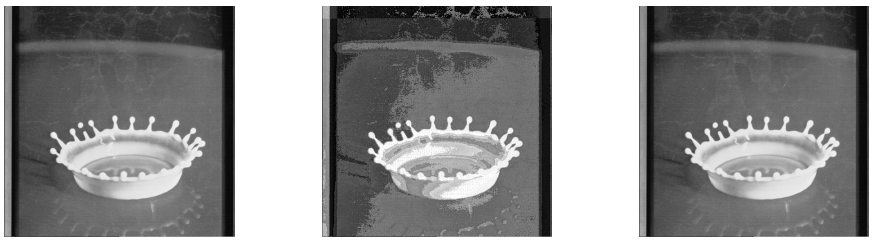
Figure 5.9: *Lena received and equalized using Algorithm II.*



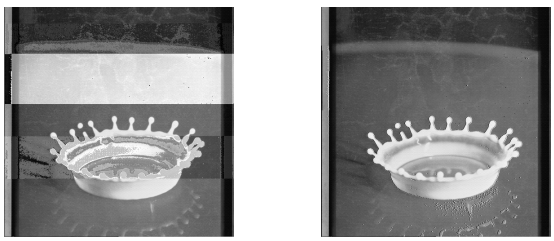
(a) Milk drop image. (b) Watermarked Milk drop.



(c) Received through channel 1. (d) Equalized image, received through channel 1. (e) Received through channel 2.



(f) Equalized image, received through channel 2. (g) Received through channel 3. (h) Equalized image, received through channel 3.



(i) Received through channel 4. (j) Equalized image, received through channel 4.

Figure 5.10: *Milk drop received and equalized using Algorithm II.*

(a) Lena image.

(b) Watermarked Lena.

(c) Received through channel 1.

(d) Equalized image, received through channel 1.

(e) Received through channel 2.

(f) Equalized image, received through channel 2.

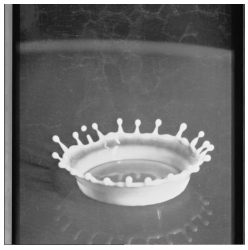
(g) Received through channel 3.

(h) Equalized image, received through channel 3.

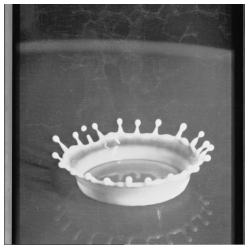
(i) Received through channel 4.

(j) Equalized image, received through channel 4.

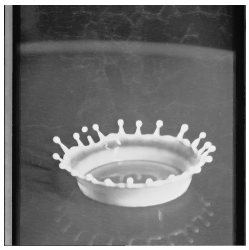
Figure 5.11: *Lena received and equalized using Algorithm III.*



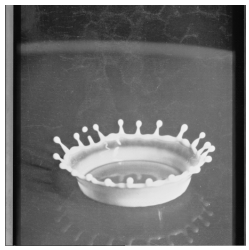
(a) Milk drop image.



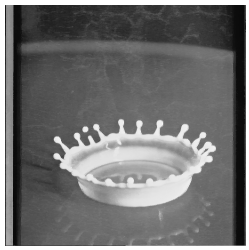
(b) Watermarked Milk drop.



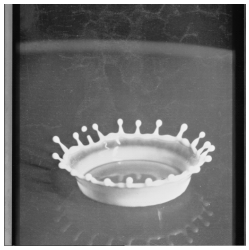
(c) Received through channel 1.



(d) Equalized image, received through channel 1.

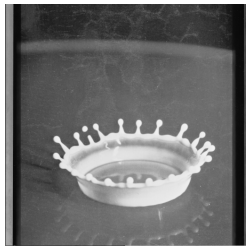


(e) Received through channel 2.

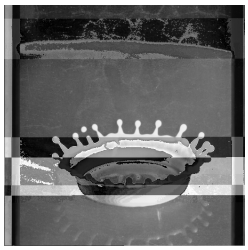


(f) Equalized image, received through channel 2.

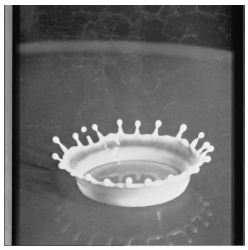
(g) Received through channel 3.



(h) Equalized image, received through channel 3.



(i) Received through channel 4.



(j) Equalized image, received through channel 4.

Figure 5.12: *Milk drop received and equalized using Algorithm III.*

6 Authentication and Scrambling of Radio Frequency Signals

6.1 Introduction

The CDMA based watermarking scheme presented in this thesis is independent of image format, which gives the flexibility that this scheme can also be modified to watermark radio frequency signals. Therefore in this chapter, a CDMA based watermarking is used for physical layer authentication of radio frequency signals. Each watermark bit is arithmetically added in a number of modulated data bits. Before demodulation this watermark can be extracted using same spreading codes. Therefore, signal can be authenticated before demodulation, and watermark is mistaken for noise to unauthorized receivers. Authorized users can detect the watermark and later remove it to get noise free original data. Proposed scheme can also be used beyond authentication. If the intensity of watermark is increased and signal is demodulated it results in wrong data. So, this automatically serves as a scrambler. Data can only be demodulated correctly, if watermark is detected and later removed using reversibility of proposed scheme.

Proposed scheme is independent of the used modulation scheme, i.e. watermark can be added to any modulated signal before transmission. Therefore, proposed watermarking algorithm is independent of the host signal, which is highly unlikely for other radio frequency watermarking algorithms. Watermark generator proposed in [77] is limited to FM signals. Watermarking presented in [78] is only for authentication and quality monitoring of analog AM signals. Radio frequency watermarking algorithm presented in [79] can be applied to OFDM wireless networks only. In proposed algorithm scrambling is done at physical layer, unlike usual schemes where scrambling is done at higher layers

[80].

6.2 Reversible Watermarking Algorithm

Fig. 6.1 shows how watermark is inserted in the data. Every watermark bit is spread using a zero mean spreading code before insertion. Spreaded watermark is arithmetically added to modulated selected bits (or to all the bits). Fig. 6.2 explains the extraction process.From the received watermarked data and same bits are selected. Just by calculating cross correlation between selected bits and spreading codes watermark is extracted. So, watermark is extracted blindly only by using spreading codes (which can be provided in secret key). Fig. 6.3 explains the watermark removal process. To retrieve original data, extracted (or already known) watermark is spread again using same spreading codes and subtracted from the selected bits of watermarked data. So only by using spreading codes watermark can be removed. Furthermore the original watermark is not required during watermark removal process.

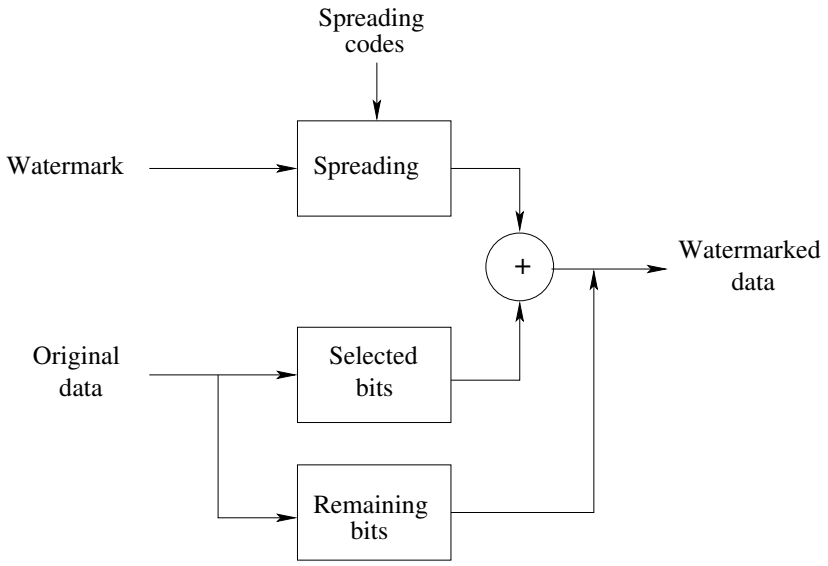


Figure 6.1: *Insertion of watermark.*

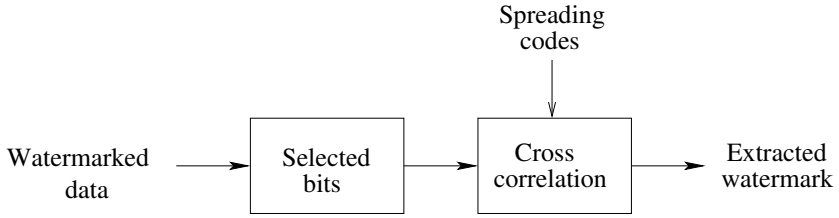


Figure 6.2: *Extraction of watermark.*

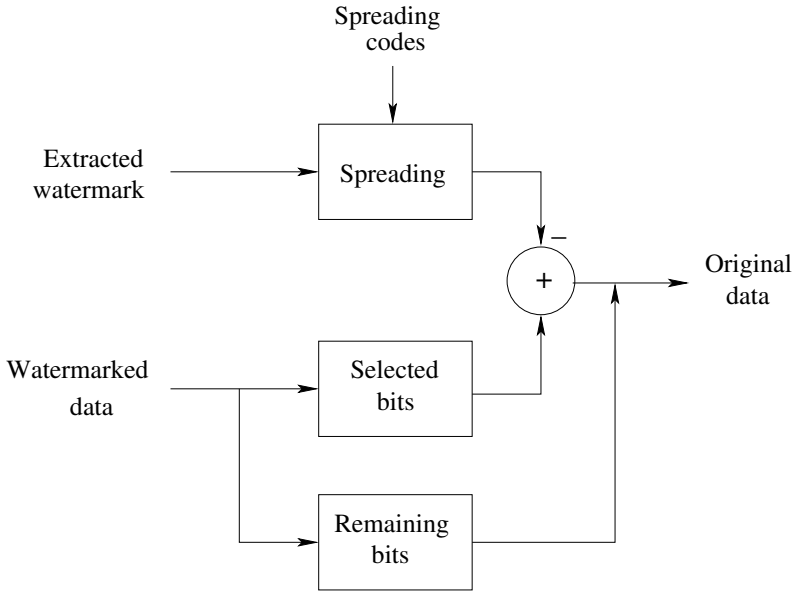


Figure 6.3: *Removal of watermark.*

6.2.1 Watermark Insertion

Let $\boldsymbol{w_{in}}$ be a binary watermark which is changed to antipodal bits (simply replace zeros with minus ones) to form a watermark representation $\boldsymbol{w} = [b_1, b_2, \ldots, b_n]$, where, $b_i \epsilon \{-1, 1\}$. Select "k" mutually orthogonal spreading codes $\boldsymbol{s_i} = [s_1, s_2, \ldots, s_l]$ each of length "l". These spreading codes should have zero mean and follow the necessary conditions (2.1), (2.2) and (2.3).

Now $\boldsymbol{X}$ be the modulated data, which is subject to watermark. $\boldsymbol{X} = [\boldsymbol{i_1}, \boldsymbol{i_2}, \ldots, \boldsymbol{i_{n/k}}]^T$ is arranged in the form "n/k" numbers of vectors. $\boldsymbol{i_j}$ represent any vector from $\boldsymbol{X}$ and is described as $\boldsymbol{i_j} = [i_1, i_2, \ldots, i_l]$, length of each $\boldsymbol{i_j}$ vector is equal to the length of spreading codes $\boldsymbol{s_i}$. Elements of $\boldsymbol{i_j}$ are modulated signals, e.g. in case of Binary Phase Shift

Keying (BPSK) signal constellation is $\{-1, 1\}$ i.e. $i_i \epsilon \{-1, 1\}$, and in case of QPSK $i_i \epsilon \{\pi/4 + j\pi/4, \pi/4 - j\pi/4, -\pi/4 + j\pi/4, -\pi/4 - j\pi/4\}$ and so on for 8-PSK and 16-PSK. Modify these $\boldsymbol{i_j}$ vectors according to:

$$\boldsymbol{i'_j} = \boldsymbol{i_j} + \alpha[b_1 \boldsymbol{s_1} + b_2 \boldsymbol{s_2} + \ldots + b_k \boldsymbol{s_k}] \tag{6.1}$$

In each $\boldsymbol{i_j}$ vector "k" bits are added, here "k" is the number of spreading codes used. "α" is the gain factor, which controls the intensity of watermark. Higher the value of α, stronger be the watermark and noisier be the watermarked data. Replace all $\boldsymbol{i_j}$ vectors by $\boldsymbol{i'_j}$ to form watermarked data $\boldsymbol{X'}$.

6.2.2 Watermark Extraction

Let $\hat{\boldsymbol{X}}$ be the received data. Select the same $\boldsymbol{i_j}$ vectors as before, which are now $\hat{\boldsymbol{i}}_j$. Every bit is extracted by calculating cross correlation between $\hat{\boldsymbol{i}}_j$ and spreading codes $\boldsymbol{s_i}$.

$$\hat{b}_i = sign < \boldsymbol{i'_j}, \boldsymbol{s_i} > \quad if\ |\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}| < |\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}| \tag{6.2}$$

All the watermark bits are extracted, using (6.2), to form extracted watermark $\hat{\boldsymbol{w}} = [\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_n]$. Fig. 6.4 shows a simulation, how $|\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}|$ and $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$ looks at different spreading code lengths. Every point in this plot is an average of thousand tests. Here, $\boldsymbol{i_j}$ are random sequences, such that $i_i \epsilon \{-1, 1\}$. It can be seen that $|\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}|$ does not grow linearly but $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$ grows linearly. As the length of spreading codes grows a point is reached where $|\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}| < |\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$. So higher the value of α and longer the spreading codes larger be the magnitude of $(\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T})$, hence more accurate is the watermark extraction. Note that, if value of α is increased, signal energy required for watermark is increased also. Fig. 6.4 shows that, to use very low value of α, very long spreading code is required, e.g. if $\alpha = 0.025$ to detect watermark bits correctly a spreading code longer than 1100 or 1200 bits is needed.

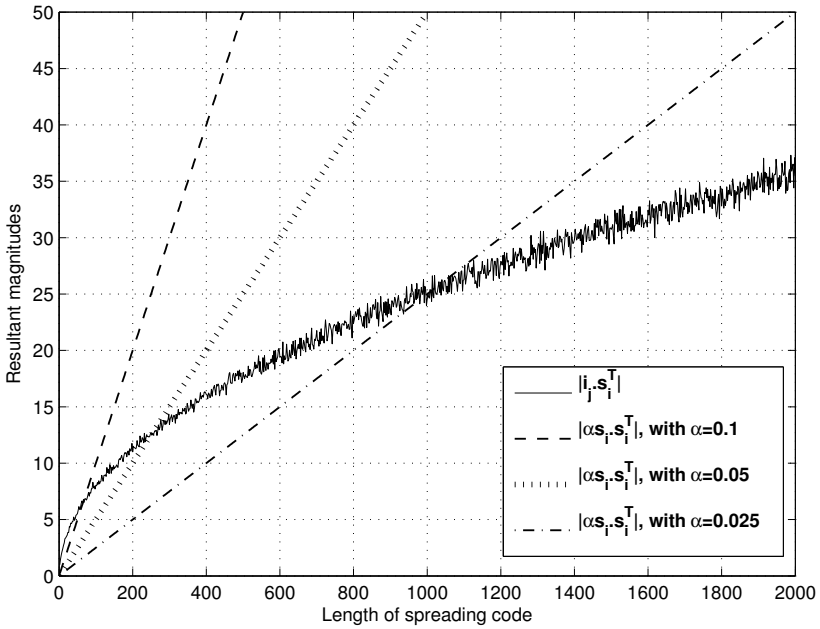


Figure 6.4: *Length of spreading code versus* $|\boldsymbol{i_j} \cdot \boldsymbol{s_i^T}|$ *and* $|\alpha \boldsymbol{s_i} \cdot \boldsymbol{s_i^T}|$.

6.2.3 Removal of Watermark

If the intensity of the watermark α is known, watermark can be removed after extraction using:

$$\boldsymbol{i_{j2}} = \hat{\boldsymbol{i}}_{\boldsymbol{j}} - \alpha[\hat{b}_1 \boldsymbol{s_1} + \hat{b}_2 \boldsymbol{s_2} + \ldots + \hat{b}_k \boldsymbol{s_k}] \tag{6.3}$$

Now all $\hat{\boldsymbol{i}}_{\boldsymbol{j}}$ are replaced by $\boldsymbol{i_{j2}}$ again to get original data back. If original watermark is known to the receiver it should be used in watermark removal process.

$$\boldsymbol{i_{j2}} = \hat{\boldsymbol{i}}_{\boldsymbol{j}} - \alpha[b_1 \boldsymbol{s_1} + b_2 \boldsymbol{s_2} + \ldots + b_k \boldsymbol{s_k}] \tag{6.4}$$

6.2.4 Scrambling using Watermarking

Scrambler is a device, which encode the data in such a way that it become unintelligible for the receiver that is not equipped with proper descrambling device. If the intensity of watermark "α" is increased,

watermarked signal becomes more noisy. If this noisy signal is demodulated it results in wrong data. Data can only be demodulated correctly, if watermark is detected and later removed using the reversibility of the proposed scheme. So, high intensity watermarked signal can be considered as scrambled signal. Watermark extractor and remover is similar to descrambling device. If unauthorized users receive the watermarked data, it is useless for them. Authorized users receive the watermarked data detect the watermark and later remove it to use the data.

6.2.5 Multiple Spreading Codes

By using multiple orthogonal spreading codes, authentication becomes more secure. First of all evedropper has to break more spreading codes. Secondly, authentication and scrambling become more tricky. For example, in case of BPSK if one spreading code and $\alpha = 0.1$, 1 become 1.1 or 0.9 and -1 become -1.1 or -0.9. If four spreading codes are used and $\alpha = 0.1$, 1 can be 1.4, 1.3, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6 and -1 can be -1.4, -1.3, -1.2, -1.1, -1.0, -0.9, -0.8, -0.7, -0.6.

6.3 Experimental Results

In all the experiments, 128000 bits long random sequences are watermarked with 500 bits. These random sequences are modulated using BPSK, QPSK, 8-PSK and 16-PSK before watermark insertion. Experiments are done using one ($l = 256,\ n = 500,\ n/k = 500$), two ($l = 512,\ n = 500,\ n/k = 250$) and four ($l = 1024,\ n = 500,\ n/k = 125$) spreading codes. These watermarked signals ($\alpha = 0.1$) are transmitted over Additive White Gaussian Noise (AWGN) channel. At receiver watermarks are extracted and then signals are demodulated. Fig. 6.5 shows percentage BER in extracted watermarks at different SNR when one spreading code is used. Here maximum BER is close to 8% at SNR = $1db$. Fig. 6.6, when two spreading codes are used maximum BER is less than 3%. There is no bit error using four spreading codes. If unwatermarked data is transmitted over the same channel BER is quite high Fig. 6.7. Although because of additive white gaussian noise

data is distorted but watermark can withstand distortions introduced by the channel. This shows the robustness of proposed watermarking scheme. Fig. 6.8, 6.9 and 6.10 shows BER in extracted watermarks at different α values using one, two and four spreading codes respectively.

Following experiments show how scrambling works using digital watermarking. Table 6.1 (one spreading code), table 6.2 (two spreading codes) and table 6.3 (four spreading codes) show the percentage of incorrectly detected data bits at different α values. Here watermarked data is demodulated before watermark removal. It can be seen, from the amount of wrong bits, that watermarked data is useless for unauthorized users. In all the cases, if watermark is extracted before demodulation and later removed, there is no bit error in demodulated data.

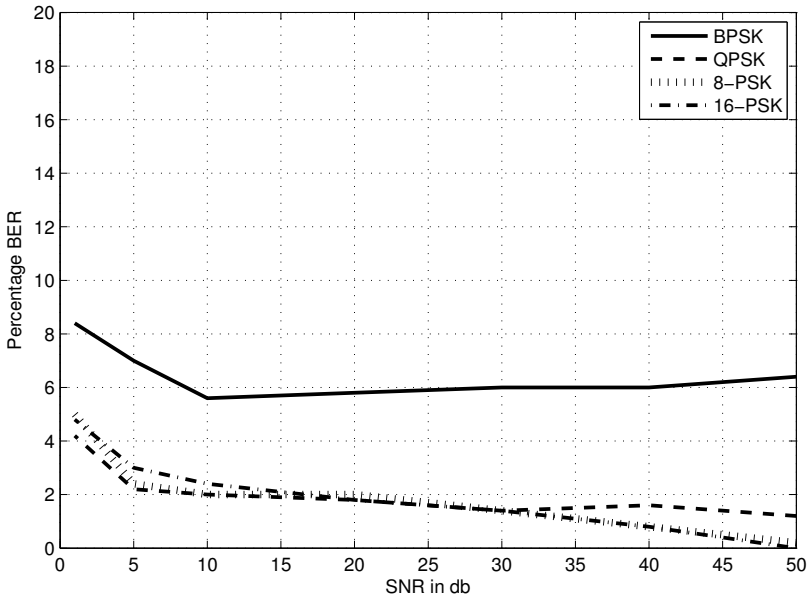


Figure 6.5: *BER in watermark because of additive white gaussian noise (one spreading code).*

6.4 Conclusion

A blind *and* reversible watermarking algorithm for radio frequency signals is presented in this chapter. This scheme can be used for authenti-

Table 6.1: Scrambling with one spreading code

α	BPSK	QPSK	8-PSK	16-PSK
2	49.98%	74.92%	87.61%	93.77%
1.5	49.89%	74.80%	87.55%	93.83%
1	0%	37.08%	68.86%	84.56%
0.75	0%	0%	49.79%	75.26%
0.5	0%	0%	50.07%	74.96%
0.25	0%	0%	0%	50.16%
0.1	0%	0%	0%	0%

Table 6.2: Scrambling with two spreading code

α	BPSK	QPSK	8-PSK	16-PSK
2	25.04%	37.51%	43.77%	46.78%
1.5	25.02%	37.30%	43.55%	46.89%
1	25.07%	37.54%	43.75%	46.89%
0.75	24.99%	37.58%	43.62%	46.82%
0.5	0%	18.80%	34.26%	42.18%
0.25	0%	0%	24.95%	37.59%
0.1	0%	0%	0%	12.61%

Table 6.3: Scrambling with four spreading code

α	BPSK	QPSK	8-PSK	16-PSK
2	30.20%	49.91%	55.90%	60.67%
1.5	30.20%	49.41%	53.37%	57.10%
1	30.78%	49.32%	53.83%	58.86%
0.75	30.85%	46.51%	52.83%	55.66%
0.5	6.69%	28.93%	49.19%	53.14%
0.25	0%	4.55%	34.53%	46.12%
0.1	0%	0%	0%	21.86%

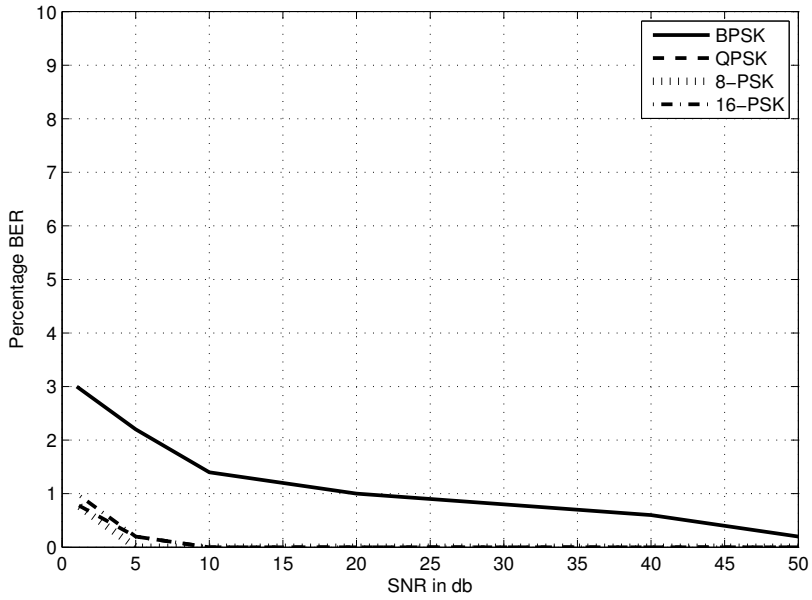


Figure 6.6: *BER in watermark because of additive white gaussian noise (two spreading codes).*

cation. Perceptually transparent watermark can be added in the form of low level noise. Reversible version of proposed algorithm can extract the watermark and later recover the original data back. A high magnitude watermark noise can serve as a scrambler. Data can not be demodulated correctly without watermark removal. Further research can be done to send some useful information beyond a simple security tag as a watermark. This information can be extracted after receiving watermarked data, processed and later removed to get original data back.

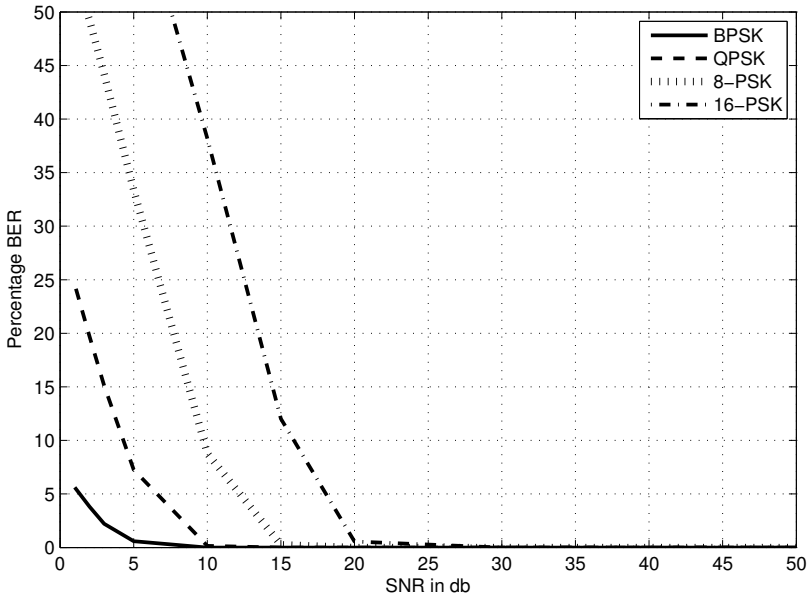


Figure 6.7: *BER in transmitted signals because of additive white gaussian noise (no watermark).*

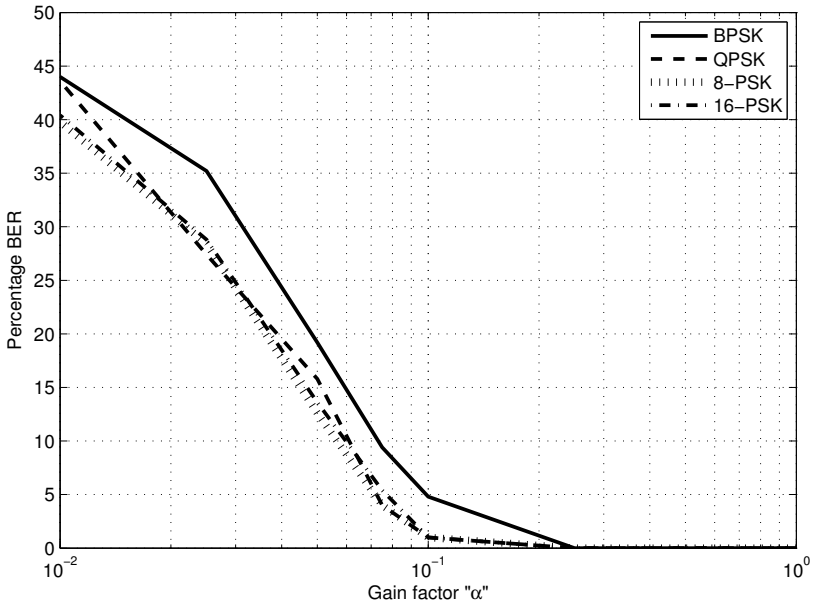


Figure 6.8: *BER in watermark at different α values (one spreading code).*

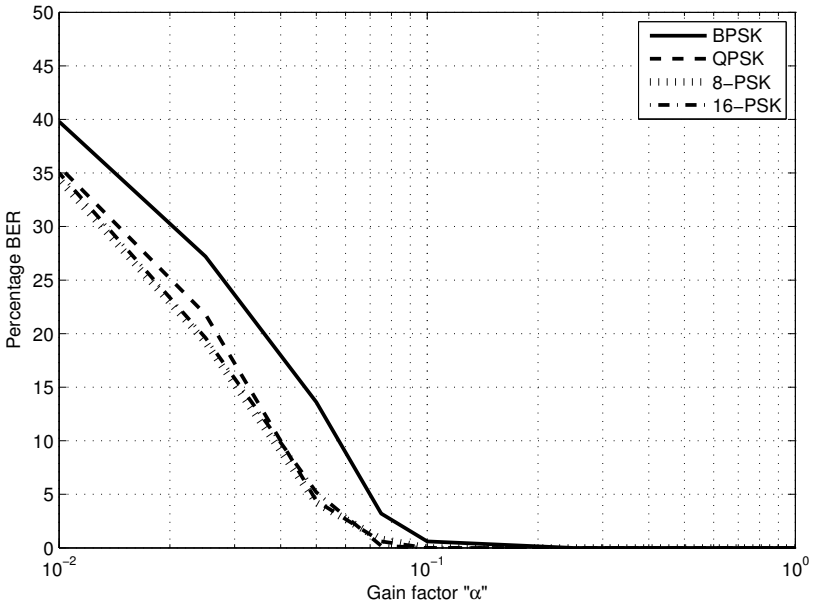


Figure 6.9: *BER in watermark at different α values (two spreading codes).*

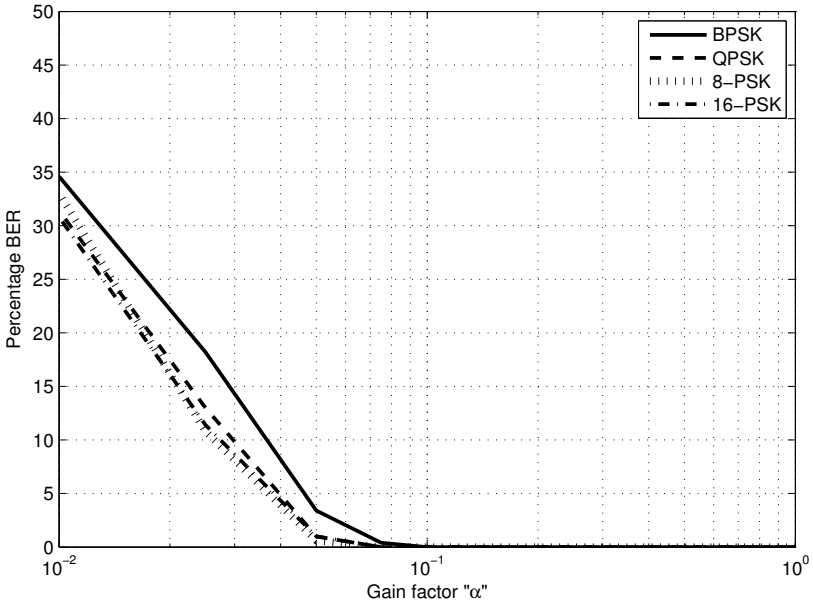


Figure 6.10: *BER in watermark at different α values (four spreading codes).*

7 Conclusions and Future Work

In this thesis, an oblivious, robust, secure and reversible watermarking scheme is proposed. As this scheme is robust it can be used for general watermarking applications, where watermark should stay in the image, after watermark detection. Usually robust watermarking schemes are not reversible and reversible watermarking schemes are not robust. So, interestingly the proposed watermarking scheme is special because it is robust and reversible.

In proposed watermarking scheme, watermark can be inserted to spatial or frequency domain. Watermarking scheme based on spatial domain is not as robust as frequency domain based scheme. However, spatial domain watermarking scheme is computationally more efficient than its frequency domain counterpart. In frequency domain watermarking, image is transformed to DCT or Wavelet domain which requires some computation. In case of spatial domain watermarking, image is not transformed, watermark is directly added to pixels. Furthermore, watermark is inserted, extracted and removed only using addition operation, even multiplication is not required. So, on light weight machines where computational complexity is more important than robustness spatial domain scheme can be implemented. However, on the machines where computational complexity is not a big issue, frequency domain watermarking should be the first choice.

Usually, reversible watermarking algorithms use some properties of the host image to embed the watermark. Later these properties are used for blind extraction and removal of watermark. In proposed watermarking algorithm, spreaded watermark is arithmetically added to host image. Insertion, extraction and removal of watermark are not dependent on host image. Therefore, proposed watermarking algorithm is independent of host image format, hence can be applied to any format of images. The fact, that proposed watermarking algorithm is not

host data dependent, gives the flexibility that proposed CDMA based watermarking algorithm can be extended to audio, video or some other multimedia data.

A novel idea of self reference watermarking (watermark is part of the original image) is presented. A part of an image is selected and this selected part is embedded in the image as watermark. Later watermarked image is attacked (transmitted, compressed or any other image processing operation performed). From that attacked image same part is selected (which was used as watermark). Later, the watermark is extracted from the attacked watermarked image. If the watermarking scheme is robust, extracted watermark is the same as the original selected part. Now one has reference to original image and attacked version of that reference. With the help of these two references, JPEG image quantization ratio is estimated and the quality of JPEG compressed images is assessed. Proposed algorithms are based on the reference to original image and attacked version of that reference. So, further research can be done to implement some more algorithms based on self reference watermarking. For example, image quality assessment algorithm can also be implemented for Gaussian blur or additive noise using self reference watermarking.

An important application based on self reference watermarking is that the transmitted images can be equalized blindly. In proposed algorithm, watermark acts as a training sequence. The received selected part can be considered as received training sequence and the extracted watermark can be considered as reference training sequence. With the help of received selected part and extracted watermark, images are equalized blindly. Simulations have shown that proposed scheme can be used for blind equalization of different channels.

It is seen that CDMA based watermarking algorithms are weak against copy attack. Copy attack means that the watermark, of the watermarked image, can be estimated and copied to some other unwatermarked images. Copy attack is dangerous in applications where watermark is used for authentication. If watermark is easily copied to some unwatermarked images, then some illicit images can be authenticated. One reason of this failure is that checkmark 1.2 [38] only uses JPEG compressed images as input. In CDMA based watermarking, spreaded watermark is added to host image in the form of noise. As JPEG

compressed images are quantized images, so it is very easy to estimate noise from the quantized images and then copy it to some other images. Even JPEG compressed CDMA based watermarked images become robust against copy attack is self reference watermarking is used. If a part is selected from an image and added in the from of watermark to same image. To authenticate that image watermark is extracted from the image. This extracted watermark is compared to received selected part. If both are same, image is authenticated. On the other hand, if that watermark, is copied to some other image, watermark extractor will not authenticate that image. Because when the watermark is extracted and later compared to received selected part both are not same. Hence, self reference watermarking is also an answer to copy attack in JPEG compressed images.

Last but not the least, CDMA based watermarking scheme for radio frequency signals is proposed. This scheme can be used to authenticate radio frequency signals. Spreaded watermark is added to the modulated signal just before transmission. Spreaded watermark is just low level added noise to the signal. Evedropper would just mistake watermark for noise. At receiver, the watermark is extracted before demodulation. If watermark is present or correctly detected, signal is authenticated. If high intensity watermark is used, the signal becomes very noisy. If this noisy signal is received and demodulated it leads to wrong data. Data can only be demodulated correctly, if watermark is extracted and removed from watermarked signal before demodulation. So, high intensity watermarking serves as automatic scrambling.

Further research can be done to develop self reference watermarking scheme for radio frequency signals. Self reference watermarking can be used for channel estimation/detection and blind equalization of wireless signals. It is seen in OFDM networks cyclic prefix is used to reduce complexity and counter intersymbol interference. Cyclic prefix is simply sending some data bits twice. The disadvantage of cyclic prefix is that it consumes extra bandwidth. In self reference watermarking we also send some data twice, first simply over the channel and second in the form of watermark. Watermark is hidden in the data in the form of noise and does not consume extra bandwidth. It is interesting to investigate if self reference watermarking can be used for cyclic extensions.

A Appendix

Figure A.1: Original images.

Figure A.2: Original images.

Figure A.3: Original images.

Figure A.4: Watermarked images at 40db using proposed watermarking scheme, watermark is spreaded over whole image.

Figure A.5: Watermarked images at 40db using proposed watermarking scheme, watermark is spreaded over whole image.

Figure A.6: Watermarked images at 40db using proposed watermarking scheme, watermark is spreaded over whole image.

Bibliography

[1] F.A.P. Petitcolas: *Introduction to Information Hiding*, Information Techniques for Steganography and Digital Watermarking, S.C. Katzenbeisser et al., Eds. Northwood, MA: Artec House, pp 1-11, Dec. 1999.

[2] I.J. Cox, J. Kilian, T. Leighton and T.Shamoon: *Secure Spread Spectrum Watermarking for Multimedia*, IEEE Transactions on Image Processing, Volume 6, Number 12, Page(s): 1673-1687, Dec. 1997.

[3] I.J. Cox, M. Miller and J.Boom: *Watermarking Application and Their Properties*, International Conference on Information Technology: Coding and Computing, Page(s): 6-10, Las Vegas 2000.

[4] A. Nikolaidis, S. Tsekeridou, A. Tefas and V. Solachidis: *A Survey on Watermarking Application Scenarios and Related Attacks*, IEEE International Conference on Image Processing (ICIP), Thessaloniki, Greece, Oct. 2001.

[5] M. Barni and F. Bartolini: *Data Hiding for Fighting Piracy*, IEEE Signal Processing Magzine, Volume 21, Number 2, Page(s): 28-39, 2004.

[6] Al Bovik: *The Essential Guide to Image Processing*, Second Edition, Academic Press, Elsevier Inc., 2009.

[7] J.A. Bloom, I.J. Cox, T. Kalker, M.I. Miller and C.B.S. Traw: *Copy Protection for dvd Video*, Proc. of IEEE, vol. 87, no. 7, pp. 1267-1276, 1999.

[8] C.-T. Hsu and J.-L. Wu: *Hidden Digital Watermarks in Images*, IEEE Trans. on Image Processing, Volume 8, Number 1, Page(s):

58-68, Jan. 1999.

[9] Y.-T. Pai, S.-J. Ruan and Jürgen Götze: *A High Quality Robust Digital Watermarking Scheme*, Pacific-Rim Conference on Multimedia, Hangzhou, P.R. China, Nov. 2006.

[10] I.J. Cox, M.L. Miller and A.L. McKellips: *Watermarking as communications with side information*, Proc. of IEEE, vol. 87, no. 7, pp. 1127-1141, July 1999.

[11] R.L. Pickholtz, D.L. Schilling and L.B. Milstein: *Theory of Spread Spectrum Communications- A Tutorial*, IEEE Transactions on Communications COM-30, Page(s): 855-884, May 1982.

[12] B. Vassaux, P. Bas and J.-M. Chassery: *A New CDMA Technique for Digital Image Watermarking, Enhancing Capacity of Insertion and Robustness*, Proceedings of the International Conference on Image Processing, Volume 3, Page(s): 983-986, Oct. 2001.

[13] Y. Xin and M. Pawlak: *Multibit Data Hiding Based on CDMA*, Canadian Conference on Electrical and Computational Engineering, Volume 2, Page(s): 935-938, May 2004.

[14] M.K. Samee and J. Götze: *Increased Robustness and Security of Digital Watermarking Using DS-CDMA*, Proceedings of the 7th IEEE International Symposium on Signal Processing and Information Technology, Cairo, Egypt, Page(s):189-193, Dec. 2007.

[15] H.C.S. Rughooputh and R. Bangaleea: *Effect of Channel Coding on the Performance of Spatial Watermarking for Copyright Protection*, IEEE 6th Africon Conference in Africa, Volume 1, Page(s):149 - 153, Oct. 2002.

[16] J. Chou and K. Ramchandran: *Robust Turbo-Based Data Hiding for Image and Video Sources*, International Conference on Image Processing Proceedings, Volume 2, Page(s): II-133 - II-136, Sept. 2002.

[17] M.K. Samee, J. Götze, S.-J. Ruan and Y.-T. Pai: *Digital Watermarking: Spreading Code versus Channel Coding*, Proceedings of the 3rd IEEE International Symposium on Communications, Control and

Signal Processing, Malta, Page(s):1409-1412, March 2008.

[18] H. Golpîra, and H. Danyali, *Reversible blind watermarking for medical images based on wavelet histogram shifting*, IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 31-36, 2009.

[19] D. Coltuc, *Improved Capacity Reversible Watermarking*, IEEE International Conference on Image Processing (ICIP), 2007, Volume: 3 , pp. III-249-III-252, 2007 .

[20] M.K. Samee and J. Götze, *CDMA Based Reversible and Blind Watermarking Scheme for Images*, International Conference on Intelligence and Information Technology (ICIIT 2010), Lahore, Pakistan, 2010 (published in Procedia Engineering, (ISSN: 1877-7058, ELSEVIER), 2012).

[21] M.K. Samee and J. Götze, *CDMA Based Blind and Reversible Watermarking Scheme for Images in Wavelet Domain*, Proc. of 19th International Conference on Systems, Signals and Image Processing (IWSSIP-2012), Vienna, Austria, April 2012.

[22] S. Altous, M. K. Samee and J. Götze, *Reduced Reference Image Quality Assessment for JPEG Distortion*, Proc. of 53rd International Symposium ELMAR-2011, Zadar, Croatia, September 2011.

[23] M. K. Samee and J. Götze, *Reduced Reference Image Quality Assessment for Transmitted Images Using Digital Watermarking*, Proc. of 7th International Symposium on Image and Signal Processing and Analysis (ISPA 2011), Dubrovnik, Croatia, September 2011.

[24] S. Altous, M. K. Samee and J. Götze, *Blind Quantization Ratio Estimation for JPEG Images Using Digital Watermarking*, 53rd International Symposium ELMAR-2011, Zadar, Croatia, September 2011.

[25] John G. Proakis, *Digital Communications*, fourth edition, McGraw Hill, 2000.

[26] M.K. Samee and J. Götze, *Blind equalization using Digital Water-*

marking, Proc. of 4th International Symposium on Communications, Control and Signal Processing, pp. 1-4, 2010.

[27] M.K. Samee and J. Götze, *Computationally Efficient Blind Equalization Based on Digital Watermarking*, Proc. of 18th European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark, 2010.

[28] M.K. Samee and J. Götze, *Channel Equalization Using Watermark as a Training Sequence*, URSI Kleinheubacher Tagung 2010, Miltenberg, Germany, 2010 (published in Advances in Radio Science, 2011).

[29] M.K. Samee and J. Götze, *Blind Equalization Using Reversible Watermarking*, Proc. of 52nd International Symposium ELMAR-2010, Zadar, Croatia, September 2010.

[30] M.K. Samee and J. Götze, *Authentication and Scrambling of Radio Frequency Signals Using Reversible Watermarking*, Proc. of 5th International Symposium on Communications, Control and Signal Processing (ISCCSP-2012), Rome, Italy, May 2012.

[31] S. P. Maity and M. K. Kundu: *A Blind CDMA Image Watermarking Scheme in Wavelet Domain*, IEEE International Conference on Image Processing, Volume 4, Page(s): 2633-2636, Oct. 2004.

[32] Y. Fang, J. Huang and Y. Q. Shi: *Image Watermarking Algorithm Applying CDMA*, IEEE International Symposium System and Circuts, Volume 2, Page(s): 948-951, May 2003.

[33] N. Mandhani and S. Kak: *Watermarking Using Decimal Sequences*, Cryptologia, Volume 29, Page(s): 50-58, Jan. 2005.

[34] J. J. K. Ó Ruanaidh and S. Pereira: *A Secure Robust Digital Image Watermark*, Electronic Imaging, SPIE Proceedings, Zrich, Switzerland, May 1998.

[35] T. Kohda, Y. Ookubo and K. Shinokura: *Digital Watermarking Through CDMA Channels Using Spread Spectrum Techniques*, IEEE

6th International Symposium on Spread-Spectrum Techniques and Applications, Sep. 2000.

[36] G.C.M. Silvestre and W.J. Dowling: *Embedding Data in Digital Images Using CDMA Techniques*, IEEE International Conference on Image Processing, Volume 1, Page(s): 589-592, April 2000.

[37] C. Shoemaker: *Hidden Bits: A Survey of Techniques for Digital Watermarking*, Independent Study, EER-290, Prof Rudko, Spring 2002.

[38] S. Pereira, S. Voloshynovskiy, M. Manneño, S. M. Maillet and T. Pun, *Second Generation Benchmarking and Application Evaluation*, Information Hiding Workshop III, Pittsburgh, PA, USA, April 2001.

[39] I. J. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker: *Digital Watermarking and Steganography*, Second Edition, The Morgan Kaufmann Series in Multimedia Information and Systems, Elsevier Inc., chapter 11, 2008.

[40] Y. Hu and B. Jeon, *Reversible Visible Watermarking Technique for Images*, IEEE International Conference on Image Processing, 2006, pp. 2577-2580.

[41] S. Emmanuel, H. C. Kiang, and A. Das, *A Reversible Watermarking Scheme for JPEG-2000 Compressed Images*, IEEE International Conference on Image Processing, 2006, pp. 69-72.

[42] Y. Du and T. Zhang, *A Reversible and Fragile Watermarking Algorithm Based on DCT*, International Conference on Artificial Intelligence and Computational Intelligence (AICI), 2009, pp. 301-304.

[43] R. Baušys and A. Kriukovas, *Reversible watermarking scheme for image authentication in frequency domain*, 48th International Symposium focused on Multimedia Signal Processing and Communications (ELMAR) 2006, pp. 53-56.

[44] J. Lee, H. Kim and J. Lee; *Information Extraction Method without Original Image using Turbo Code*, Proceedings of International Conference on Image Processing, Volume 3, Page(s):880 - 883, Oct.

2001.

[45] M. Kesal, M.K. Mihcak, R. Koetter and P. Moulin; *Iteratively Decodable Codes for Watermarking Applications*, Proc. 2nd Symposium on Turbo Codes and Their Applications, Brest, France, Sep. 2000.

[46] J. Huang and Y.Q. Shi; *Reliable information bit hiding*, IEEE Transactions on Circuits and Systems for Video Technology, Volume 12, Issue 10, Page(s):916 - 920, Oct. 2002.

[47] A. Bastug and B. Sankur: *Improving the Payload of Watermarking Channels via LDPC Coding*, IEEE Signal Processing Letters, Volume 11, Issue 2, Part 1, Page(s): 90-92, Feb. 2004.

[48] S. Baudry, J.-F. Delaigle, B. Sankur, B. Macq, H. Maître: *Analyses of the error correction strategies for typical communication channels in watermarking*, Signal Processing 81, Elsevier 2001, Page(s):1239-1250.

[49] Y.Q. Shi, X.M. Zhang, Z.-C. Ni and N. Ansari: *Interleaving for Combating Bursts of Errors*, IEEE Circuits and Systems Magazine, Volume 4, Issue 1, Page(s):29 - 42, First Quarter 2004.

[50] P. Gastaldo, G. Parodi, J. Redi and R. Zunino, "No-reference quality assessment of JPEG image by using CBP neural networks," IEEE International Symposium on Circuits and System, 5, pp.772- 775, 2004.

[51] H.R. Sheikh, A.C. Bovik and L. Cormack, "Image quality assessment using natural scene statistics," Image Processing, IEEE Transactions, pp. 1918 - 1927, 2004.

[52] W. Gao, C. Mermer and Y. Kim, "A de-blocking algorithm and a blockiness metric for highly compressed images," IEEE Transactions on Circuits and Systems for Video Technology 12, pp. 1150ï¿½ 1159, 2002.

[53] S.A. Karunasekera and N.G. Kingsbury, "A distortion measure for blocking artifacts in images based on human visual sensitivity," IEEE Transactions on Image Processing 4, pp. 713-724 , 1995.

[54] H.R. Wu and M. Yuen, "A generalized block-edge impairment metric for video coding," IEEE Signal Processing Letters 70, pp. 247 - 278, 1998.

[55] R.V. Babu and S. Suresh, "GAP-RBF based NR image quality measurement for JPEG coded images," Springer-Verlag Berlin Heidelberg, ICVGIP, LNCS 4338, pp. 718- 727, 2006.

[56] Z. Wang, G. Wu, E. Yang and A.C. Bovik, "Quality-aware images," IEEE Transactions on Image Processing, volume (15), pp. 1680 - 1689, 2006.

[57] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," IEEE Transactions on Image Processing, vol. 12 , issue 2, pp. 230-235, 2003.

[58] Z. Fan and R. L. de Queiroz, "Maximum likelihood estimation of JPEG quantization table in the identification of Bitmap compressed history," Proceedings of 2000 International Conference on Image Processing, vol. 1, pp. 948-951, 2000.

[59] S. Hamdy, H. El-Messiry, M. Roushdy and E. Kahlifa, "Quantization table estimation in JPEG images," International Journal of Advanced Computer Science and Applications, vol. 1, No. 6, Dec. 2010.

[60] G.-S. Lin, M.-K. Chang and Y.-L. Chen, "A passive-blind forgery detection scheme based on content-adaptive quantization table estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21 , issue 4, pp. 421-434, 2011.

[61] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," IEEE Trans. Circuits Sys. Video Technol. vol. 2, pp. 91-94, Mar. 1992.

[62] Z. Fan and R. Eschbach, "JPEG decompression with reduced artifacts," Pro. IS&T/SPIE Symposium on ELectronic Imaging: Image and Video Compression, San Jose, CA, 1994.

[63] C.I. Podilchuk, and E.J. Delp, "Digital watermarking: algorithms

and applications ," IEEE Signal Processing Magazine, vol. 18, issue 4, pp. 33-46, 2001.

[64] G.K. Wallace, "The JPEG still picture compression standard," IEEE Transactions on Consumer Electronics, 1991.

[65] B. Chen and G. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," IEEE Transactions Information Theory, volume (47), pp. 1423-1443, 2001.

[66] H.R. Sheikh, Z. Wang, A.C. Bovik and L.K. Cormack, "Image and video quality assessment research, LIVE database," Http://live.ece.utexas.edu./research/quality/, 2003.

[67] Simon Haykin: *Adaptive Filter Theory*, third edition, Prentice Hall, 1996.

[68] L.R. Litwin, Jr.: *Blind Channel Equalization*, IEEE Potentials, Volume 18, Issue 4, Page(s):9-12, October-November 1999.

[69] P. Campisi, M. Carli, G. Giunta and A. Neri: *Blind Quality Assessment System for Multimedia Communications Using Tracing Watermarking*, IEEE Transactions on Signal Processing [see also IEEE Transactions on Acoustics, Speech, and Signal Processing], Volume 51, Issue 4, Page(s):996-1002, April 2003.

[70] Y.-W. Ding, Z. Lin and L. Wang: *A Multipurpose Public-Key Cryptosystem Based Image Watermarking*, 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM '08., Page(s):1-4, 12-14 Oct. 2008.

[71] F. Benedetto, G. Giunta and A. Neri: *End-to-End QoS Provision and Control in Wireless Communication Systems by Means of Digital Watermarking Signal Processing*, 3rd International Symposium on Wireless Communication Systems. ISWCS '06. Page(s):655-659, 6-8 Sept. 2006.

[72] S. P. Maity, M. K. Kundu and S. Maity: *An Efficient Digital Watermarking Scheme for Dynamic Estimation of Wireless Chan-*

nel Condition, International Conference on Computing: Theory and Applications. ICCTA '07, Page(s):671-675, 5-7 March 2007.

[73] M.U. Neto, L.C.T. Gomes, J.M.T. Romano and M. Bonnet: *Adaptive Equalization based on Watermarking*, 2006 International Telecommunications Symposium, Page(s):743-748, 3-6 Sept. 2006.

[74] V. Zarzoso and P. Comon: *Blind and Semi-Blind Equalization Based on the Constant Power Criterion*, IEEE Transactions on Signal Processing, Volume 53, Issue 11, Page(s):4363-4375, November 2005.

[75] L. Sun and C. Zhao: *Optimizing Blind Equalization Intelligent Algorithm for Wireless Communication Systems*, 3rd International Conference on Natural Computation, ICNC 2007, Volume 3, Page(s):146-149, 24-27 August 2007.

[76] O. Dabeer and E. Masry: *Convergence Analysis of the Constant Modulus Algorithm*, IEEE Transactions on Information Theory, Volume 49, Issue 6, Page(s):1447-1464, June 2003.

[77] B. Bogdan and J. Lopatka, *A Real Time Generator of Watermarking Signal for FM Radios*, Proc. of the 9th European Conference on Wireless Technology, pp. 269-272, 2006.

[78] M. Li, Y. Lei, X. Zhang, J. Liu and Y. Yan, *Authentication and Quality Monitoring Based on Audio Watermark for Analog AM Shortwave Broadcasting*, Proc. of 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, vol. 2, pp. 263-266, 2007.

[79] J.E. Kleider, S. Gifford, S. Chuprun, and B. Fette, *Radio frequency watermarking for OFDM wireless networks*, Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.5, pp. V-397-400, 2004.

[80] G. Zhou, P. Fan; L. Hao and N. Suehiro, *DFT Scrambling Vector OFDM and Its Performance Analysis*, Proc. of 3rd International Conference on Communications and Mobile Computing, pp. 381-384, 2011.

Personal Information

Name:	Muhammad Kashif Samee
Born:	06.09.1978
Born in:	Lahore / Pakistan
Marital status:	married, two children.

Curriculum Vitae

1998 – 2002	Study electrical engineering at the University of Engineering and Technology, Lahore, Pakistan.
2003 – 2006	MS Automation and Robotics Dortmund University of Technology, Dortmund, Germany.
2007 – 2012	Ph.D. student at the Information Processing Lab Dortmund University of Technology.