Spring 2010

A Novel Digital Audio Watermarking Approach by Embedding Coefficients in Discrete Cosine Transform Domain

Erol Duymaz
Old Dominion University

A NOVEL DIGITAL AUDIO WATERMARKING APPROACH BY EMBEDDING COEFFICIENTS IN DISCRETE COSINE TRANSFORM DOMAIN

by

Erol Duymaz
B.S. July 2002, 9th September University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2010

Approved by:

Zia-ur Rahman (Director)

Dimitrie Popescu (Member)

Jiang Li (Member)

ABSTRACT

A NOVEL DIGITAL AUDIO WATERMARKING APPROACH BY EMBEDDING COEFFICIENTS IN DISCRETE COSINE TRANSFORM DOMAIN

Erol Duymaz
Old Dominion University, 2010
Director: Dr. Zia-ur Rahman

Watermarking is a basic secure communication method. It is used for embedding a recognizable pattern in media in such a manner that modification of the media also modifies the pattern, thus making it easy to detect the modification. This technique and its variants have many practical applications pertaining to secure communications, media verification, etc. *Digital audio watermarking* is a technique for embedding data within an audio signal in such a way that the original and the modified audio signals are essentially identical. The embedded data can be used for various purposes such as secure communication in military applications, owner identification and verification, content authentication, etc.

In this thesis, a watermark audio signal is hidden in a message audio signal by using a discrete cosine transform (DCT) domain approach. The tests of fidelity between the original and the watermarked signal and robustness applied to the watermarked signal. The results with both the Human Auditory System (HAS) and numeric/graphic aspects are presented. The results show that an embedded watermark is not easily detectable using either the HAS or other techniques. Additionally, it can be detected successfully in the simulation domain, but it may be susceptible to some noise and channel limitations in the real world.

Keywords: Digital audio watermarking, Discrete Cosine Transform (DCT), Discrete Cosine Transform (DCT) coefficients.

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake.

…and to the Turkish Air Force.

ACKNOWLEDGMENTS

I am heartily thankful to my supervisor, Dr. Zia-ur Rahman who guided my study with his precious comments and to all my instructors, both at Aeronautics and Space Technologies Institute in Istanbul, Turkey and at ODU in Norfolk VA, USA, who provided my background with their teachings.

I offer my regards and blessings to my family and all of my friends who supported me in all respects throughout this project.

And to my honey Burcu Aksu Duymaz who did not let me walk alone during my whole study.

TABLE OF CONTENTS

Page

LIST OF TABLES.....viii

LIST OF FIGURES.....ix

Chapter

I. INTRODUCTION.....1

1. PROBLEM STATEMENT.....2
1.1 Related Works.....2

2.SPECIFIC OBJECTIVES.....5

II. BACKGROUND OF THE STUDY.....7

1.WATERMARKING.....7
1.1 Watermarking History.....8

2.DIGITAL WATERMARK AND APPLICATIONS.....8
2.1 Signature.....9
2.2 Copyright Protection.....9
2.3 Fingerprinting.....9
2.4 Content Authentication.....10
2.5 Copy Protection.....10
2.6 Broadcast Monitoring.....10

3.PROPERTIES OF DIGITAL WATERMARK.....11
3.1 Embedding Efficiency.....11
3.2 Fidelity.....11
3.3 Fragility.....12
3.4 Data Payload.....12
3.5 Blind or Informed Detector.....12
3.6 False Positive Rate.....13
3.7 Robustness, Security and Cost.....13

4.DIGITAL AUDIO WATERMARK TECHNIQUES.....13
4.1 Quantization Based Schemes.....16
4.2 Spread Spectrum.....17
4.3 Two Set.....18
4.4 Self Marking.....18
4.5 Replica.....19

TABLE OF CONTENTS

Page

Chapter

5.TRANFORM DOMAINS.....19
5.1 Fourier Transform.....20
5.2 Short Time Fourier Transform.....24
5.3 Wavelet Transform.....25
5.4 Discrete Cosine Transform.....27

6.A GOOD SIMULATION TOOL FOR ALGORITM : MATLAB....28

III. APPLICATION AND RESULTS.....30

IV. TRANSMISSION CONSIDERATIONS.....41

V. CONCLUSIONS AND FUTURE WORK.....44

REFERENCES.....46

APPENDIX
MATLAB CODE OF ALGORITHM.....48

VITA.....52

LIST OF TABLES

Table Page

1. The precision of the DCT coefficient as a function of the SNR and the type of audio signal 43

LIST OF FIGURES

Figure Page

1. Annual number of papers on watermark by IEEE.....2
2. A general watermark system.....7
3. A typical audio watermarking schemes.....14
4. A simple quantization scheme.....16
5. Kernels for echo hiding.....19
6(a). Time domain representation of a stationary signal.....23
6(b). Frequency domain representation of a stationary signal.....23
7(a). Time domain representation of a nonstationary signal.....23
7(b). Frequency domain representation of a nonstationary signal.....23
8(a). Overlapping window parameters used in the STFT analysis.....25
8(b). Illustration of how analysis windows are moved during analysis.....25
8(c). Spectral window used during STFT analysis propose.....25
9. Matlab interface.....29
10. Message signal in time domain.....30
11. Watermark signal in time domain.....31
12. Magnitude of the DCT coefficients of the message signal.....31
13. Magnitude of the DCT coefficients of the watermark signal.....32
14. The watermark embedding module (at sender side).....34
15. The watermark detection module (at receiver side).....35
16. The message and the watermark signal DCT coefficients respectively.....37
17. The detected watermark signal in time domain.....38
18. The DCT coefficients of the original signal and the watermarked signal.....39
19. The watermark and the detected watermark signal DCT coefficients respectively.....40
20. The detected watermark signal DCT coefficients.....41
21. Future work algorithm for watermark embedding at sender side.....45
22. Future work algorithm for watermark detection at receiver side.....45

CHAPTER 1

INTRODUCTION

Communication between members of a species has existed since the advent of the species. Leaping ahead to the human species, communication started when some human needed to convey some information to another and understand them in return. Since those early forms of communications in gestures and sounds, much has changed in both the style and methods of communication. Nowadays humans communicate with each other using multiple techniques and methods not limited to face-to-face verbal communication. Today it is possible, for instance, to communicate with people a long distance away over a wire or through a small device held in your hand.

Sometimes people need to hide their communication from everyone except the one intended to receive the communication. For instance, a prisoner who wants to plan an escape may try to hide his communications with his conspirator using some kind of code, or a battle commander needs to hide his message to officers in the field using some other technique. So the need for ways to perform secure or unshared communication arose.

Watermarking is a basic secure communication method. It embeds a recognizable pattern in transmission media to provide authentication. Digital audio watermark is a technique for embedding additional data along with the audio signal. Embedded data is used for various purposes such as copyright protection, owner identification, or security. A number of audio watermarking techniques are in existence today. They use different methods to embed a robust watermark and keep original signal fidelity. In this thesis, a watermark audio signal will be hidden in a message signal in the discrete cosine

---

The reference model for this thesis is IEEE Transactions.

transform (DCT) domain, and tests will be performed on the watermarked signal to verify robustness and fidelity.

We will describe related works and the purpose of the study in the first section. The brief history of watermarking, digital audio watermarking techniques, signal fundamentals and transform methods, and simulation domain acknowledge are presented in Chapter 2. A description of our algorithm and results are given in Chapter 3. Chapter 4 describes the impact of transmission considerations on performance, and Chapter 5 provides some future research directions and conclusions.

1.1 PROBLEM STATEMENT

In the following section we discuss the state-of the art of audio watermarking.

1.1.1 Related Works

The number of works on watermarking has increased sharply in recent years. The number of studies published in IEEE since the birth of digital watermarking in the mid 1990s are given in Figure 1 for ten years [1]. Digital audio watermark has also see an increased interest in these years.

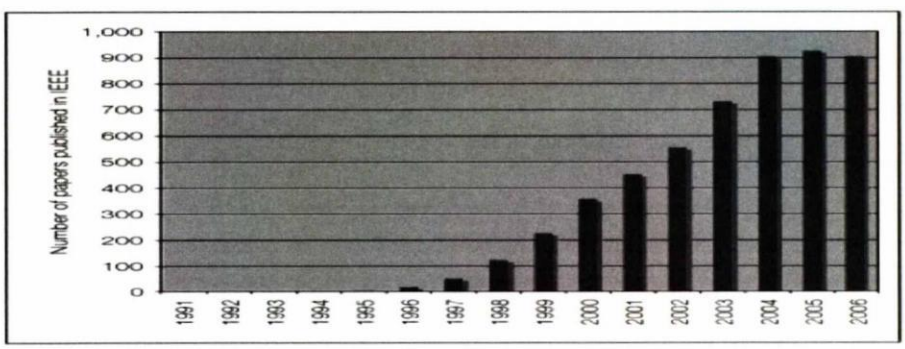


Figure 1: Annual number of papers on watermarking by IEEE [1].

Digital audio watermarking can be performed using many different techniques. Alsalamai and Al-Akaidi [2] present a good survey of digital audio watermarking principles, and Kim [3] examined digital audio watermarking techniques in his study.

Among the researchers, Wang and Zhao [4] used a combination of the discrete wavelet transform (DWT) and the DCT for synchronization invariant audio watermarking. In their study, they used a blind digital audio watermarking scheme which can extract the watermark without the help of the original signal against synchronization attack, which uses adaptive quantization not to eliminate the watermark information from the watermarked signal but to change the embedding position so that the detector cannot detect the right watermark. They proposed a new approach to resist the synchronization attack more effectively and combined the DWT and DCT to improve the transparency of the digital watermark. They embedded the watermark into the low frequency components of the signal. Their experimental results show that their proposed watermarking scheme produces watermarks that are inaudible and robust against various signal processing issues such as noise, resampling, requantization, random cropping or compression [4].

Bhat et al. [5] used "Mean Quantization" in the Cepstrum domain. They proposed a novel audio watermarking algorithm based on the Cepstrum transform, which is a common transform used to derive information from a speech signal that can be used to separate the excitation signal (which contains the words and the pitch) and the transfer function (which contains the voice quality) for audio copyright protection. That blind algorithm embeds the watermark data into the original audio signal using mean quantization of Cepstrum coefficients. Experimental results in the study show that the

audio watermarking scheme is not only imperceptible but also robust against various common signal processing attacks such as noise adding, resampling, low-pass filtering, requantization, compression and cropping. In addition, the performance is better than Cepstrum-based audio watermarking schemes based on statistical mean manipulation [5].

Ramalingam and Krishnan [6] used Gaussian Mixture Models (GMMs) of short-time Fourier Transform (STFT) features for audio fingerprinting. Regarding their states in audio fingerprinting, an audio clip must be recognized by matching an extracted fingerprint to a database of previously computed fingerprints. The fingerprints should reduce the dimensionality of the input significantly, provide discrimination among different audio clips, and, at the same time, be invariant to distorted versions of the same audio clip [6].

They designed the audio fingerprints by modeling an audio clip on a GMM and evaluated the performance of many easy-to-compute STFT features, such as Shannon entropy, spectral centroid, spectral bandwidth and spectral flatness measure. These features were further used to test the robustness of the fingerprints under a large number of distortions. To make the system robust, they used some of the distorted versions of the audio for training. However, they showed that the audio fingerprints modeled using GMM were not only robust to the distortions used in training but also to distortions not used in training [6].

As for this thesis, two signals—a message and a watermark—will be combined using the DCT domain. The watermark audio signal will be hidden in the message signal in such a way that no knowledge of the original signal is needed to retrieve the watermark from the transmitted signal. Tests will be performed on the watermarked signal to verify

robustness and fidelity and results will be derived from mathematical analysis using the MATLAB environment. Additionally, results from the HAS will be used to evaluate the performance.

1.2 SPECIFIC OBJECTIVES

With the spread of the Internet in recent years, digital multimedia works like video, audio and images have become increasingly available for electronic transmission, production, and publishing. Connected to this increase in the use of digital media is the strong desire for protection against unauthorized copy and propagation to protect financial and proprietary rights. These concerns triggered research for finding ways to deter copyright trespassing. One of the best solutions for this challenging problem looks to lie in information hiding techniques. Information hiding is the process of embedding a message into the digital signal. The embedded message needs to be audibly imperceptible.

There are many digital watermarking algorithms in the literature today. Generally speaking, each has some drawbacks even while it is sufficient in other aspects. This is the reason why researchers continue to look for better algorithms. This thesis proposes a novel method for digital audio watermarking that is applicable to many applications ranging from multimedia to military uses. We evaluate the algorithm thoroughly in a simulation environment and discuss its pros and cons.

The new method is simple to implement. It works on the representation of the audio signals in the DCT where the signal is represented by its coefficients. The algorithm is based on hiding the watermark signal transform coefficients in the message signal

coefficients. To evaluate system performance, different host signals from a human voice to music are used, and the effect of noise, bandwidth limitations and cropping on the algorithm are investigated.

The results of this study will provide some idea of the performance of the proposed novel digital watermark algorithm and its pros and cons versus other approaches existent in the literature.

CHAPTER 2

BACKGROUND OF THE STUDY

2.1 WATERMARKING

A watermarking system consists of three modules: watermark signal generation module, watermark embedding module and watermark detection module [3]. The watermark signal is generated using a non-invertible function that takes as an input a watermark key. In some systems the host signal (cover-object) is taken into account when the watermark is generated. This will help the watermark generator in producing an imperceptible signal-dependent watermark [3].

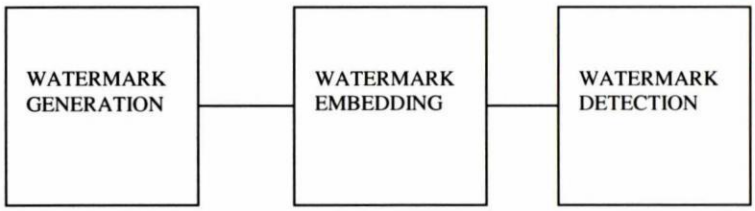


Figure 2: A general watermark system.

Watermark embedding can be performed either in the time domain or in the transform domain (DFT, DCT, DWT, etc.) using a suitable embedding rule (e.g., addition or multiplication). Finally, watermark detection is performed by some sort of correlation detector or statistical hypothesis testing, with or without resorting to the original signal [3].

2.1.1 The History of Watermarking

Probably the first known watermark was used in Italy more than 700 years ago to indicate the paper brand and the mill that produced it [7]. By the 18th century watermarks began to be used as anti-counterfeiting measures on money and other documents. The term "watermark" was introduced near the end of the 18th century. It was probably given because the marks resemble the effects of water on paper. The first example of a technology similar to digital watermarking is a patent filed in 1954 by Emil Hembrooke [8] for identifying music works. In 1988, Komatsu and Tominaga [8] appear to be the first to use the term "digital watermarking." Around 1995, interest in digital watermarking began to mushroom, and today it is very popular in both science and the entertainment world [8].

2.2 DIGITAL WATERMARKING AND APPLICATIONS

Digital watermarking is a technology that allows a secret message to be hidden in media, without detection by the user. That watermark is not apparent to the user and does not affect in any way use of the original data. Watermark information is predominantly used to identify the creator of a digital file, i.e. a picture, a song, or text. The requirements that a watermarking system has to comply with are always based on the application. Thus, before we review the requirements and design considerations, we will present some applications of watermarking [1].

2.2.1 Signature

The watermark identifies the owner of the content. This information can be used by a potential user to obtain the legal rights to copy or publish the document from the owner. It might also be used to help settle ownership disputes [8].

2.2.2 Copyright Protection

Copyright protection is the most important application of watermarking. The objective is to embed information that identifies the copyright owner of the digital media, in order to prevent other parties from claiming the copyright. This application requires a high level of robustness to ensure that the embedded watermark cannot be removed without causing a significant distortion in the digital media. Additional requirements besides the robustness also have to be considered. For example, the watermark must be unambiguous and still resolve rightful ownership if other parties embed additional watermarks [2].

2.2.3 Fingerprinting

The objective of this application is to convey information about the legal recipient rather than the source of the digital media, in order to identify single distributed copies of the digital work. It is very similar in idea to the serial number of software products. In this application a different watermark is embedded into each distributed copy, in contrast to the first application where only a single watermark is embedded into all copies of digital media. Just as with the copyright protection application of watermarking, fingerprinting requires high robustness [2].

2.2.4 Content Authentication

The objective of this application is to detect any modification of the digital data. This can be achieved with the so-called fragile watermarks that have a low robustness to certain modification (e.g. compression). The idea here is that any small manipulation would destroy the watermark, thus indicating that the digital media has been tampered with [2].

2.2.5 Copy Protection

This application tries to find a mechanism to disallow unauthorized copying of digital media. Copy protection is very difficult in open systems; in a closed system, however, it is feasible. In such systems it is possible to use watermarks to indicate the copy status of the digital media (e.g. copy once or never copy). On the other hand, the copy software or device must be able to detect the watermark and allow or disallow the requested operation according to the copy status of the digital media being copied [2].

2.2.6 Broadcast Monitoring

Producers of advertisements or audio and video works want to make sure that their works are broadcast during the time they purchase from broadcasters. The low-tech method of broadcast monitoring is to have human observers watch the broadcasting channels and record what they see or hear. This method is costly and error prone. The solution is to replace human monitoring with automated monitoring. One method of automated broadcast monitoring is to use watermarking. With watermarking, an identification code can be embedded in the work being broadcast. A computer-based monitoring system can detect the embedded watermark and determine whether the

broadcast was aired at the correct time and for the duration of airtime purchased from the broadcasters [2].

2.3 PROPERTIES OF DIGITAL WATERMARKING

Watermarking systems can be characterized by a number of properties. The relative importance of each property depends on the requirements of the application. The properties being discussed in this section are associated with the watermark embedder, the watermark detector, or both [2].

2.3.1 Embedding Effectiveness

The effectiveness of a watermarking system is the probability that the output of the embedder will be watermarked. The cover work is said to be watermarked when the input to a detector results in a positive detection. The effectiveness of a watermarking system may be determined analytically or empirically by embedding a watermark in a large number of cover works and detecting the watermark. The percentage of cover works that result in positive detection will be the probability of effectiveness [2].

2.3.2 Fidelity

In general, the fidelity of a watermark system refers to the perceptual similarity between the original and the watermarked version of the cover work. However, since the watermarked version may be degraded in the transmission process prior to being perceived by a person, a different definition of fidelity may be more appropriate. We may define watermarking system fidelity as a perceptual similarity between the un-watermarked and watermarked works at the point at which they are presented to a viewer [2].

2.3.3 Fragility

Some application fields require exactly the opposite of robustness. Consider, for example, the use of paper watermarks in bank notes. The point of these watermarks is that they do not survive any kind of copying and therefore can be used to indicate the bill's authenticity. In some applications, the watermark is required to survive certain transformations and be destroyed by others, and that makes the design of fragile watermarking difficult [8].

2.3.4 Data Payload

Data payload refers to the number of bits a watermark embeds in a unit of time or work. For audio, data payload refers to the number of embedded bits per second that are transmitted. Different applications require different data payloads. For example, copy control applications may require just a few bits embedded in cover works while fingerprinting applications may require substantially more data [2].

2.3.5 Blind or Informed Detector

We refer to the detector that requires the original, un-watermarked work as an informed detector. Informed detectors may require information derived from the original work rather than original work itself. Conversely, detectors that do not require the original work are referred to as blind detectors. Informed detectors generally have good performance in watermark extraction. However, this results in a huge number of original works that have to be stored [2].

2.3.6 False Positive Rate

A "false positive" is the detection of a watermark in a cover work that does not actually contain one. When we talk of a false positive rate, we refer to the number of false positives we expect to occur in a given number of runs of the detector [2].

2.3.7 Robustness, Security and Cost

Robustness refers to the ability to detect the watermark after common signal processing operations. Audio watermarking needs to be robust to temporal filtering, A/D conversion, time scaling, etc. Not all applications of watermarking require robustness to all the forms of processing operations. This depends on the nature of application of the watermarking system. The security of a watermark refers to its ability to resist hostile attacks. A hostile attack is a process that is specifically used to thwart the watermark's purpose. Hostile attacks can fall into three categories: unauthorized removal, unauthorized embedding, and unauthorized detection. The cost of a watermarking system refers to the speed with which embedding and detection must be performed and the number of embedders and detectors that must be deployed. Other issues include whether the detector and the embedder are to be implemented as a hardware device or as a software application or plug-in [2].

2.4 DIGITAL AUDIO WATERMARK TECHNIQUES

Audio watermarks are special signals embedded into digital audio. These signals are extracted by detection mechanisms and decoded to provide the original watermark. Audio watermarking schemes rely on the imperfection of the human auditory system. However, the human ear is much more sensitive than other sensory systems. Thus, good

audio watermarking schemes are difficult to design. Even though the current watermarking techniques are far from perfect, audio watermarking schemes have been applied widely during the last decade. These schemes are very sophisticated in terms of robustness and imperceptibility. While robustness and imperceptibility are both important requirements of watermarking, they are often in conflict with each other because robustness often entails embedding more data that often leads to perceptible distortion in the cover signal [3]. Figure 3 shows several audio watermarking schemes that belong to the class of blind watermarking. Non-blind watermarking schemes are theoretically interesting but not so useful practically since they require double storage capacity and double communication bandwidth for watermark detection. Of course, non-blind schemes may be useful as a copyright verification mechanism in a copyright dispute [3].

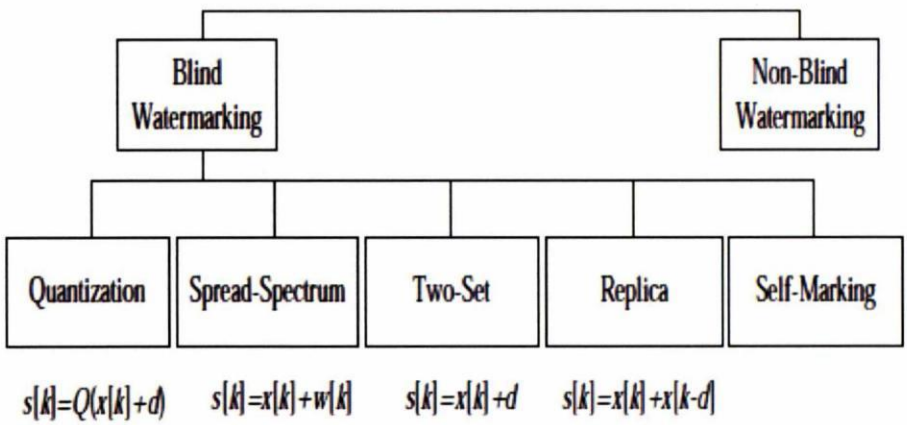


Figure 3: A typical audio watermarking schemes [3].

On the other hand, blind watermarking schemes can detect and extract watermarks without use of the un-watermarked audio. Therefore, it requires only a half storage

capacity and half bandwidth compared with the non-blind watermarking scheme. Hence, only blind audio watermarking schemes are considered in this thesis. Needless to say, the blind watermarking methods need self-detection mechanisms for detecting watermarks without un-watermarked audio [3]. A partial list of blind watermarking techniques is given below:

1. Quantization based watermarking which quantizes the sample values of the signal to make valid and invalid sample values.
2. Spread-spectrum method based on the similarity between watermarked audio and pseudo-random sequence.
3. Two-set methods based on differences between two or more sets, which include the patchwork scheme.
4. Replica method that uses a close copy of the original audio, including the replica modulation scheme [3].
5. Self-marking scheme [3].

Of course, many more schemes and their variants exist. For example, time-base modulation is theoretically interesting. However, this mechanism is a non-blind watermarking scheme. The audio watermarking scheme that encodes compressed audio data does not embed real watermarking signal into raw audio. Furthermore, no psychoacoustic model is available in the compressed domain to enable the adjustment of the watermark to ensure inaudibility [3].

Synchronization is important for detecting watermarks especially when the embedded audio signal has been attacked. Most of the audio watermarking schemes are position-based, i.e., watermarks are embedded into specific positions and detected from

that position. Thus, an attack that causes a shift in the signal, leading to a shift in positions of the embedded data, will cause position-based detection schemes to fail. The main purpose of synchronization schemes is to find the shifted positions [3].

Several synchronization schemes are surveyed in the literature [3]. In audio watermarking, time-scaling or pitch-scaling attack is one of the most difficult attacks to manage [3].

2.4.1 Quantization Based Schemes

A scalar quantization scheme quantizes a sample value x and assigns a new value to the sample x based on the quantized sample value. In other words, the watermarked sample value y is represented as follows [3]:

$$y = \begin{cases} q(x, D) + D/4 & \text{if } b = 1 \\ q(x, D) - D/4 & \text{otherwise} \end{cases} . \tag{1}$$

x is quantized to $q(x,D)$ or to the black circle (•) in Figure 4, where $q(x,D)$ denotes the anchor. If the watermarking bit b is 1, the anchor is moved to the white circle (◦). Otherwise, the cross (×) stands for the watermarking bit 0 [3]. For example, for D = 8, and x = 81, $q(81, 8)$ = 80. If b = 1, then y = 82; otherwise, y = 78. As shown in Figure 4, the distance between anchors is D [3].

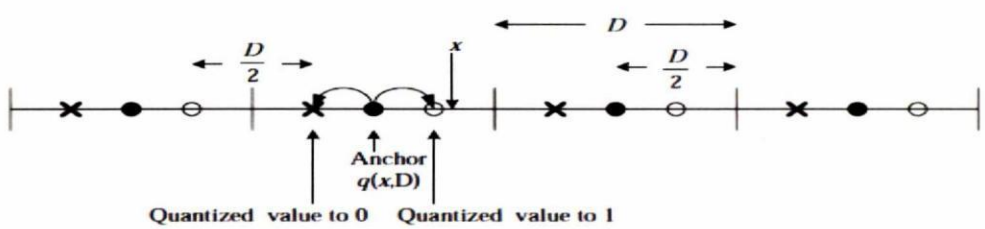


Figure 4: A simple quantization scheme [3].

Detection is the inverse process of embedding. The detection process is summarized as follows:

$$b = \begin{cases} 1 & \text{if } 0 < y - q(x, D) < D/4 \\ 0 & \text{if } -D/4 < y - q(x, D) < 0 \end{cases} \quad . \tag{2}$$

The detected bit b is derived from the received signal y based on the relationship defined in Equation 3. This scheme is simple to implement and is robust against noise attack so long as the noise margin is below $D/4$ [3].

2.4.2 Spread Spectrum

The spread-spectrum watermarking scheme is an example of correlation-based methods that embed a pseudorandom sequence and detect the watermark by calculating the correlation between the pseudorandom noise sequence and the watermarked audio signal. Spread-spectrum is one of the most popular schemes and has been studied well in the literature. This method is easy to implement but has some serious disadvantages: it requires time-consuming, psycho-acoustic shaping to reduce audible noise, and is susceptible to time-scale modification attack [3].

The spread-spectrum scheme spreads a pseudorandom sequence across the audio signal. The wideband noise can be spread into either the time-domain or the transform-domain signal, no matter what transform is used. Frequently used transforms include the DCT, the Discrete Fourier Transform (DFT), and the DWT. The binary watermark message $v = \{0,1\}$ or its equivalent bipolar variable $b = \{-1, +1\}$ is modulated by a pseudorandom sequence $r(n)$ generated by means of a secret key. Then the modulated watermark $w(n) = br(n)$ is scaled according to the required energy of the audio signal $s(n)$ [3].

A scaling factor α controls the trade-off between robustness and inaudibility of the watermark. The modulated watermark $w(n)$ is equal to either $r(n)$ or $-r(n)$ depending on whether $v = 1$ or $v = 0$. The modulated signal is then added to the original audio to produce the watermarked audio $x(n)$ as [3]:

$$x(n) = s(n) + \alpha w(n) \tag{3}$$

The detection scheme uses linear correlation. Because the pseudorandom sequence, $r(n)$ is known and can be regenerated by means of a secret key; watermarks are detected by using correlation between $x(n)$ and $r(n)$ as

$$c = \frac{1}{N}\sum_{i=1}^{N} x(i)r(i), \tag{4}$$

where N denotes the length of signal [3].

2.4.3 Two Set

A blind watermarking scheme can be devised by making two sets different. For example, if two sets are different, then we can conclude that the watermark is present. Such decisions are made by hypothesis tests typically based on the difference of means between two sets. Making two sets of audio blocks have different energies can also be a good solution for blind watermarking. Patchwork also belongs to this category. Of course, depending on the applications we can exploit the differences between two sets or more [3].

2.4.4 Self-Marking

The self-marking method embeds watermarks by leaving self-evident marks in the signal. This method either embeds a special signal into the audio or changes the signal

shape in the time domain or the frequency domain. The time-scale modification method and many schemes based on the salient features belong to this category. A clumsy self-marking method (for example, embedding a peak into the frequency domain) is prone to attack since it is easily noticeable [3].

2.4.5 Replica

In replica watermarking, the original signal can be used as an audio watermark. Echo hiding is a good example of this technique [3].

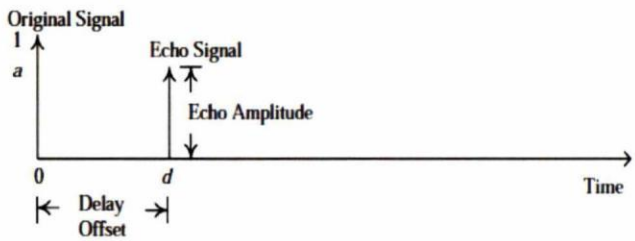


Figure 5: Kernels for echo hiding [3].

Replica modulation also embeds part of the original signal in the frequency domain as a watermark. Thus, replica modulation embeds a properly modulated original signal as a watermark. The detector can also generate the replica from the watermarked audio and calculate the correlation. The most significant advantage of this method is its high immunity to synchronization attack [3].

2.5 TRANSFORM DOMAINS

As is known, it is possible to represent a signal as a function of other signals. For example in Fourier transforms, a signal is represented by the sum of the *sine* and the

cosine functions. Many other transform forms also exist. We can examine the various transform domains and determine which is the most suitable for the application.

2.5.1 Fourier Transform

The Fourier transform is a systematic way to decompose "generic" functions into a superposition of "symmetric" functions. These symmetric functions are usually quite explicit (such as a trigonometric function $sin(nx)$ or $cos(nx)$) and are often associated with physical concepts such as frequency or energy [9].

Indeed, the Fourier transform is a fundamental tool in the study of groups (and, more precisely, in the representation theory of groups, which roughly speaking show that a group can define a notion of symmetry). The Fourier transform is also related to topics in linear algebra, such as the representation of a vector as linear combinations of an orthonormal basis or as linear combinations of eigenvectors of a matrix (or a linear operator) [9].

To give a very simple prototype of the Fourier transform, consider a real-valued function $f: R \rightarrow R$. Recall that such a function $f(x)$ is even if $f(-x) = f(x)$ for all $x \in R$ and is odd if $f(-x) = -f(x)$ for all $x \in R$. A typical function f, such as $f(x) = x^3+3x^2+3x+1$, will be neither even nor odd. However, one can always write f as the superposition $f = f_e + f_o$ of an even function f_e and an odd function f_o by the formula [9]:

$$f_e(x) := \frac{f(x)+f(-x)}{2}; f_o(x) := \frac{f(x)-f(-x)}{2} \quad . \tag{5}$$

For instance, when $f(x) = x^3+3x^2+3x+1$, then $f_e(x) = 3x^2+1$ and $f_o(x) = x^3+3x$. Note also that this decomposition is unique; there are no other even functions $\tilde{f}_e$ and odd functions $\tilde{f}_o$ such that $f = \tilde{f}_e + \tilde{f}_o$ [9].

This rudimentary Fourier transform is associated with the two-element multiplicative group $\{-1, +1\}$, with the identity element +1 associated to the identity map $x \rightarrow x$ on the real line, and the other element -1 associated to the reflection map $x \rightarrow -x$ [9].

For a more complicated example, let $n \geq 1$ be an integer, and consider a complex valued function f: $C \rightarrow C$. If $0 \leq j \leq n-1$ is an integer, let us say that such a function $f(z)$ is a harmonic of order j if we have $f(e^{2\pi i/n}z) = e^{2\pi ij/n} f(z)$ for all $z \in C$; note that even and odd functions correspond to the cases $j = 0$, $n = 2$ and $j = 1$, n = 2 respectively. As another example, the functions z^j , z^{j+n}, z^{j+2n}, etc. are harmonics of order j. Then we can split any function uniquely as a superposition $f = \sum_{j=0}^{n-1} f_j$ of harmonics of order j, by means of formula [9]:

$$f_j(x) := \frac{1}{n}\sum_{k=0}^{n-1} f(e^{2\pi ik/n}x)e^{-2\pi ijk/n} \quad . \tag{6}$$

In some signal processing operations, one may need to have both time and frequency information. When the signal at hand is a time domain signal, a conversion from time amplitude representation to frequency domain representation may be obtained by using the Fourier Transform (FT) as defined in equation [10]:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \quad . \tag{7}$$

The FT decomposes a signal into its frequency components by multiplying with a complex exponential that has sines and cosines of frequency ω, and integrates over all times, so if the signal has a component of ω, that component and the sinusoidal term will

coincide and give a relatively large value. Because of the integration term that runs over all time ranges, there is no time information in the Fourier transformed signal [10]. This is why the FT is a translation between two extreme representations of a signal, namely between $x(t)$, which is perfectly localized in time and $X(\omega)$, which is perfectly localized in frequency [10].

However, a frequency domain signal can be transformed into the time domain by using the inverse FT (IFT) given by [10]:

$$x(t) = \int_0^{2\pi} X(\omega) e^{j\omega t} dt \quad . \tag{8}$$

It also follows that no matter where in time a frequency component occurs it will have the same effect on the integration in equation 8. However, if we have a non-stationary signal, where the frequency content changes over time, we may need time information in addition to frequency information. Thus, it may be inferred that FT is not suitable for nonstationary signals [10]. On the other hand, as frequency content does not change in time for stationary signals, all frequency components exist at all times. Since there is no need for the time information for a stationary signal, FT can work well for those. Both of the signals in Figures 6 and 7 contain the same four frequency components. However, the stationary signal in Figure 6 contains them at all times, while the nonstationary signal in Figure 7 contains them successively. Except the disturbance like components, the two FTs are alike. However, one cannot argue about the time localization of the four dominant frequency components in Figure 7 [10]. Here it should be noted that the graphs in Figure 7 have different scales from those in Figure 6.

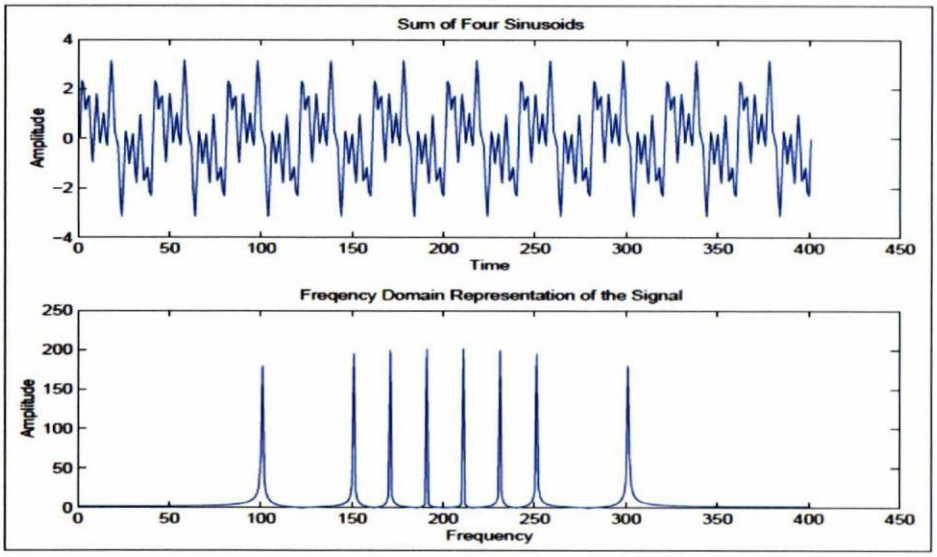


Figure 6: (a) Time domain representation and (b) frequency domain representation of a stationary signal [10].

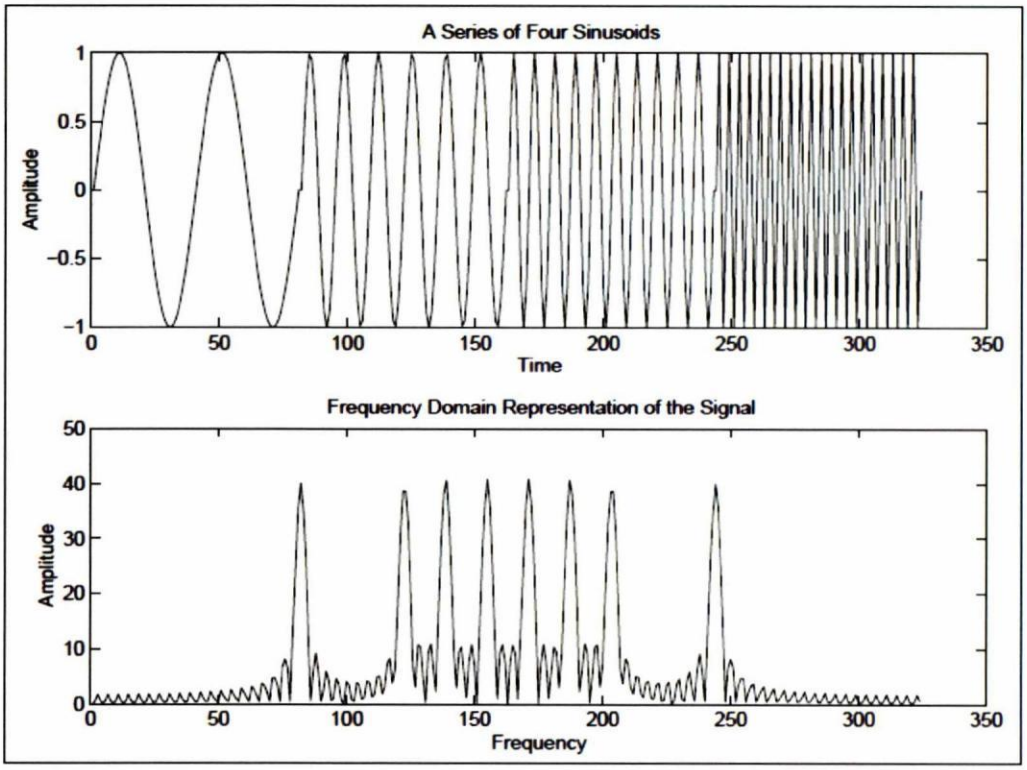


Figure 7: (a) Time domain representation and (b) frequency domain representation of a nonstationary signal [10].

2.5.2 Short-time Fourier Transform (STFT)

To obtain information both on time localization and frequency content of a signal one may use the short time Fourier transform (STFT). The motivation of STFT is in assuming the signal to be stationary for a while. It is mostly used for 2D signals, i.e. a 2D image signal. Due to the nonstationary nature of an image, traditional Fourier analysis is not adequate to analyze the image completely [11].

The STFT is needed to resolve the properties of the image both in space and also in frequency. It can extended the traditional one-dimensional time-frequency analysis to two-dimensional image signals to perform short (time/space)-frequency analysis. Here let us recapitulate some of the principles of 1D STFT analysis and show how it is extended to 2D [11].

When analyzing non-stationary 1D signal $x(t)$, it is assumed that it is approximately stationary in the span of a temporal window $w(t)$ with finite support. The STFT of $x(t)$ is now represented by time frequency atoms $X(\tau,\omega)$ and is given by [11]:

$$X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)\omega^*(t-\tau)e^{-j\omega t}dt \quad . \tag{9}$$

In the case of 2D signals, the space-frequency atoms is given by [11]:

$$X(\tau_1,\tau_2,\omega_1,\omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x,y)W^*(x-\tau_1,y-\tau_2)e^{-j(\omega_1 x+\omega_2 y)}dxdy \quad . \tag{10}$$

Here τ_1, τ_2 represent the spatial position of the two-dimensional window $W(x,y)$; ω_1, ω_2 represent the spatial frequency parameters. Unlike the regular FT, the result of the STFT is dependent on the choice of the window $\omega(t)$ [11].

For the sake of analysis any smooth spectral window such as Hanning, Hamming or even a Gaussian window may be utilized. However, when one is also interested in enhancing and reconstructing an image directly from the Fourier domain, one's choice of a window is fairly restricted [11]. Figure 8 illustrates how the spectral window is parameterized. At each position of the window, it overlaps (OVRLP) pixels with the previous position. This preserves the ridge continuity and eliminates 'blocking' effects common with other block processing image operations. Each such analysis frame yields a single value of the dominant orientation and frequency in the region centered on ω_1, ω_2 [11].

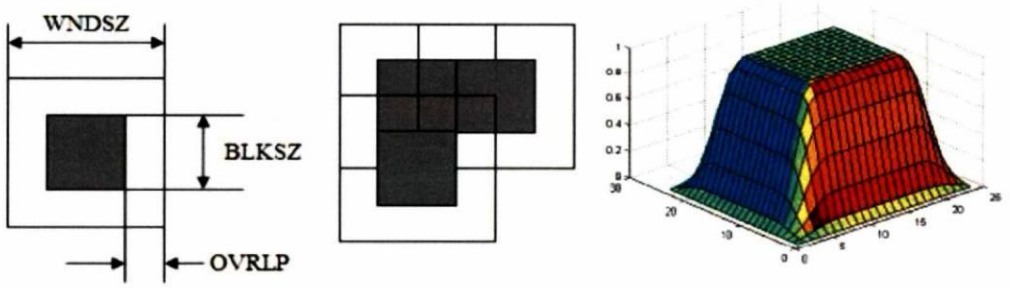


Figure 8: Overlapping window parameters used in the STFT analysis (left); illustration of how analysis windows are moved during analysis (middle); spectral window used during STFT analysis phase (right) [11] (WNDSZ:windowsize and BLKSZ:blocksize).

2.5.3 Wavelet Transform

The Wavelet transform (WT) gives us the ability to compute the frequency content of the input signal at variable resolutions [10]. It provides a representation, in terms of a set of wavelet functions that are the translated and scaled versions of a single mother

wavelet function. Say $\psi(t)$ is the mother wavelet function. In this case the set of window functions are [10].

$$\psi_{s,t}(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-\tau}{s}\right) \tag{11}$$

where τ is the translation parameter, and s is the scale (dilation) parameter [10]. They are chosen to have a unit norm so that [10]:

$$\int_{-\infty}^{\infty}\left|\frac{1}{\sqrt{|s|}}\psi\left(\frac{t-\tau}{s}\right)\right|^2 dt = 1 \tag{12}$$

Equation 14 summarizes the idea of wavelet transform in continuous time:

$$CWT(x;\psi(\tau,s)) = \frac{1}{\sqrt{|s|}}\int x(t)\psi^*\left(\frac{t-\tau}{s}\right)dt \tag{13}$$

By taking the inner products of the input signal $x(t)$ and the translated and scaled versions of the mother wavelet function, one can express $x(t)$ in terms of the set of wavelet functions. When the windowing function is of finite length, the transform is said to be compactly supported [10].

In order to implement the idea of CWT in a digital environment, one needs to convert continuous time operations into discrete time domain. With a special choice of dilation and translation parameters, one can switch from continuous wavelet transform to discrete time wavelet transform. Usually the parameters are chosen according to equations:

$$\begin{aligned} s &= s_0^{-m} \\ \tau &= n\tau_o s_0^{-m} \end{aligned} \tag{14}$$

where m and n are integers [10].

In this case, the discrete time wavelet transform equation becomes [10]

$$X(m,n) = s_0^{m/2} \int_{-\infty}^{\infty} x(t)\psi(s_0^m t - n\tau_0)dt \quad . \tag{15}$$

In digital signal processing operations everything is in discrete time. Here the function $\psi(t)$ can be said to be discretized at the values of $\psi(t)$ at instants $t = s_0^m t - n\tau_0$. On the other hand, sampling in time domain will make $\psi(t)$ a discrete function which results in the discrete wavelet transform (DWT) [10].

2.5.4 Discrete Cosine Transform (DCT)

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation, each transform coefficient can be encoded independently without losing compression efficiency. This section describes the DCT [12]. The most common DCT definition of a 1-D sequence of length N is

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left(\frac{\pi(2x+1)u}{2N}\right) \tag{16}$$

for $u = 0,1,2,\ldots,N-1$ [12].

Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u)C(u)\cos\left(\frac{\pi(2x+1)u}{2N}\right) \tag{17}$$

for $x = 0,1,2,\ldots,N-1$ [12]. In both equations $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0 \end{cases} \qquad (18)$$

It is clear that $C(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)$. Thus, the first transform coefficient is proportional to the average value of the sample sequence. In literature, this value is referred to as the DC coefficient. All other transform coefficients are called the AC coefficients [12].

To avoid dealing with complex numbers of other transform methods, we are going to use the DCT for its simplicity in audio signal processing.

2.6 A GOOD SIMULATION TOOL FOR ALGORITHM : MATLAB

While there are several higher level programming environments like Mathematica or Maple that can be used for simulation of signal processing algorithms, we use the Matlab environment. Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [13].

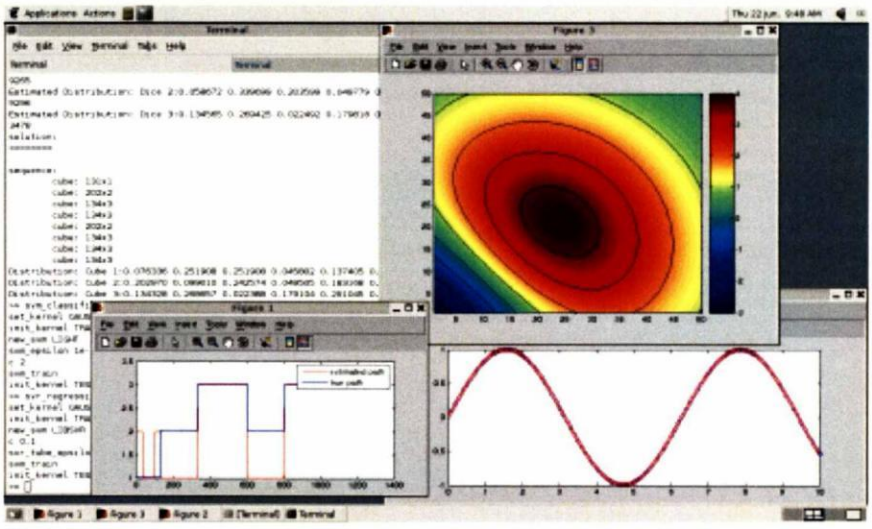


Figure 9: Matlab interface [13].

Typical uses of Matlab include [13]:

- Math and computation;
- Algorithm development;
- Modeling, simulation, and prototyping;
- Data analysis, exploration, and visualization;
- Scientific and engineering graphics;
- Application development, including graphical user interface building.

Matlab is an interactive system whose basic data element is an array that does not require dimensioning. This allows one to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran. Matlab features a family of application-specific solutions called toolboxes. We use the communications and the digital signal processing toolboxes in our research [13].

CHAPTER 3

APPLICATION AND RESULTS

The main issue in watermarking schemes is to decide how to embed the watermark into the message. Today many ways exist in the literature. While the nonblind algorithms tend to be very robust, their requirement that the original signal is needed to detect the watermark is not always practical. For this reason, blind approaches, which do not require the original signal to detect the watermark signal, are preferred among researchers.

In the study we start with two audio signals: the original message which is to be transmitted and the watermark signal. The watermark signal is be embedded into the original signal. In a computer, the two signals are recorded and transformed to the DCT domain using MATLAB. Figure 10 shows the time domain representation of the original signal, and Figure 11 shows the time domain respresentation of the watermark.

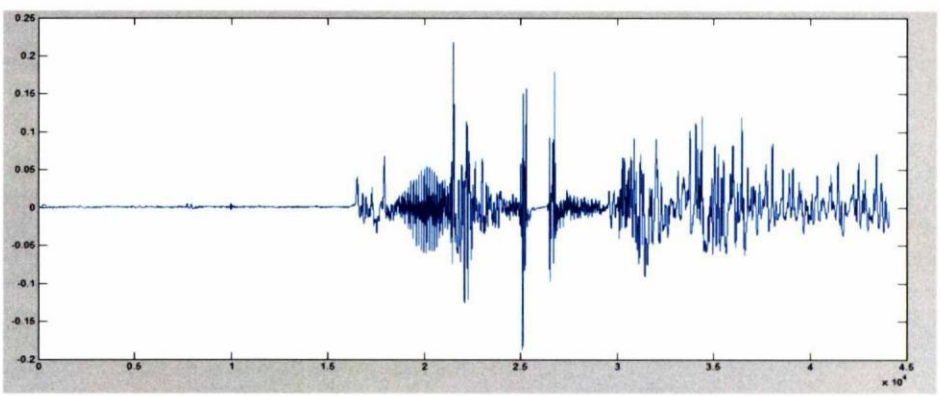


Figure 10: Message signal in time domain (merhaba).

The recording of the Turkish word "merhaba" which means hello in English has been chosen as the original message, and the watermark signal is the recording of another Turkish word "gunaydin" which means good morning in English.

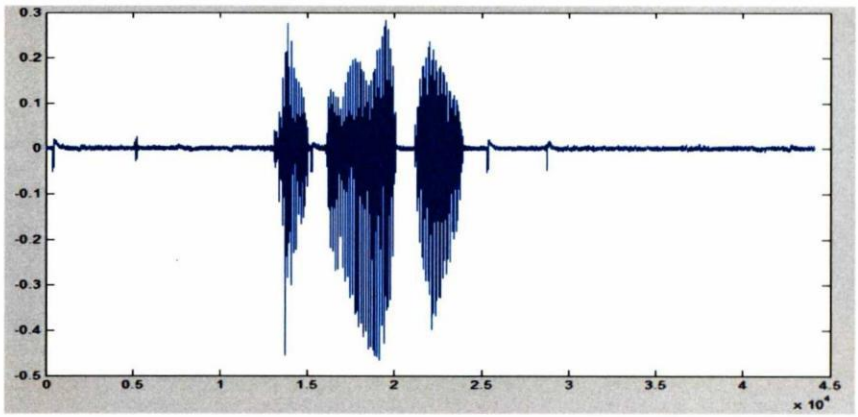


Figure 11: Watermark signal in time domain (gunaydin).

By using the DCT, the two signals are transformed to the frequency. Here the signals can be represented by the coefficients of DCT as shown in Figures 12 and 13.

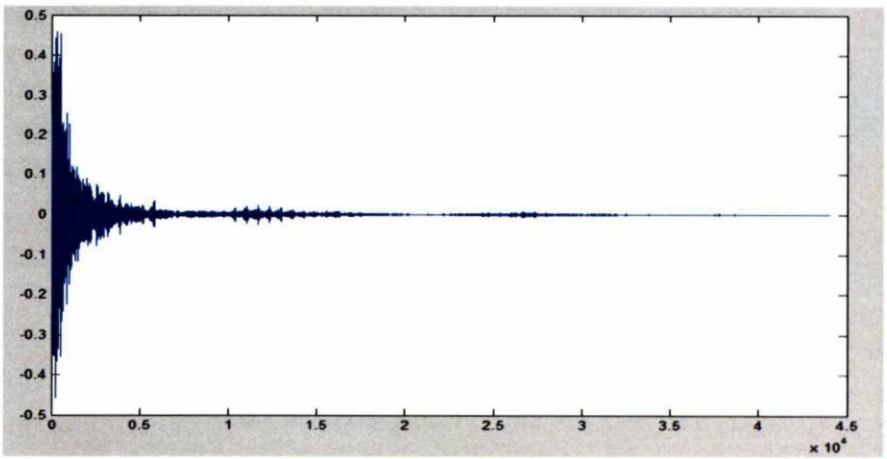


Figure 12: Magnitude of the DCT coefficients of the message signal.

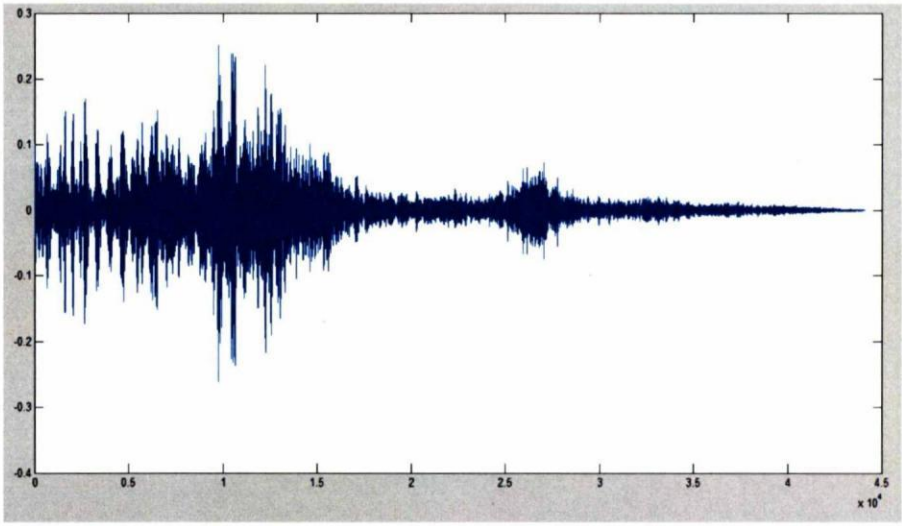


Figure 13: Magnitude of the DCT coefficients of the watermark signal.

The main idea in this thesis is to embed the DCT coefficients of the watermark signal into the coefficient of the message signal.

The process for embedding the watermark into the signal is fairly straightforward and relies on quantizing the DCT coefficients of the two signals. The floating-point representation of a DCT coefficient is represented by:

$$c = d_n d_{n-1} \mathrm{K}\ d_0 \cdot m_1 m_2 \mathrm{K}\ m_p \quad . \tag{19}$$

where c is the DCT coefficient, d_i, $i = 1,...n$ are the digits to the left of the decimal and m_k, $k = 1,...,p$ are the mantissa, or the digits to the right of the decimal sign. The algorithm for data embedding is as follows:

1. Truncate the DCT coefficient of the message to 4 decimal places, i.e., $p = 4$.

2. Truncate the DCT coefficient of the watermark to 4 significant digits, i.e., p = 4.

3. Multiply the truncated watermark coefficient by 10^{-5} to shift it 5 places to the left. After the multiplication, the watermark DCT coefficient will take the form: 0.0000.....

4. Concatenate the truncated signal coefficient with the shifted truncated watermark coefficient to form the DCT coefficient of the embedded message.

5. Take the inverse DCT of the concatenated signals, and transmit.

This procedure forms a DCT combined DCT coefficient that has 10 digits in the mantissa. An 11th digit needs to be added to the representation to indicate the sign of the coefficient of the watermark signal. If the watermark signal coefficient and the message signal coefficients have the same sign, then the sign digit is set to zero. If they differ then the sign digit is set to 1 and the sign of the watermark coefficient—the signal to be retrieved—is obtained from the sign of the DCT coefficient of the transmitted data. If the coefficient of the transmitted signal is negative and the sign digit is a 1, the DCT coefficient of the watermark is positive and vice versa. The block diagrams in Figures 14 and 15 show this approach for the embedding and detection processes.

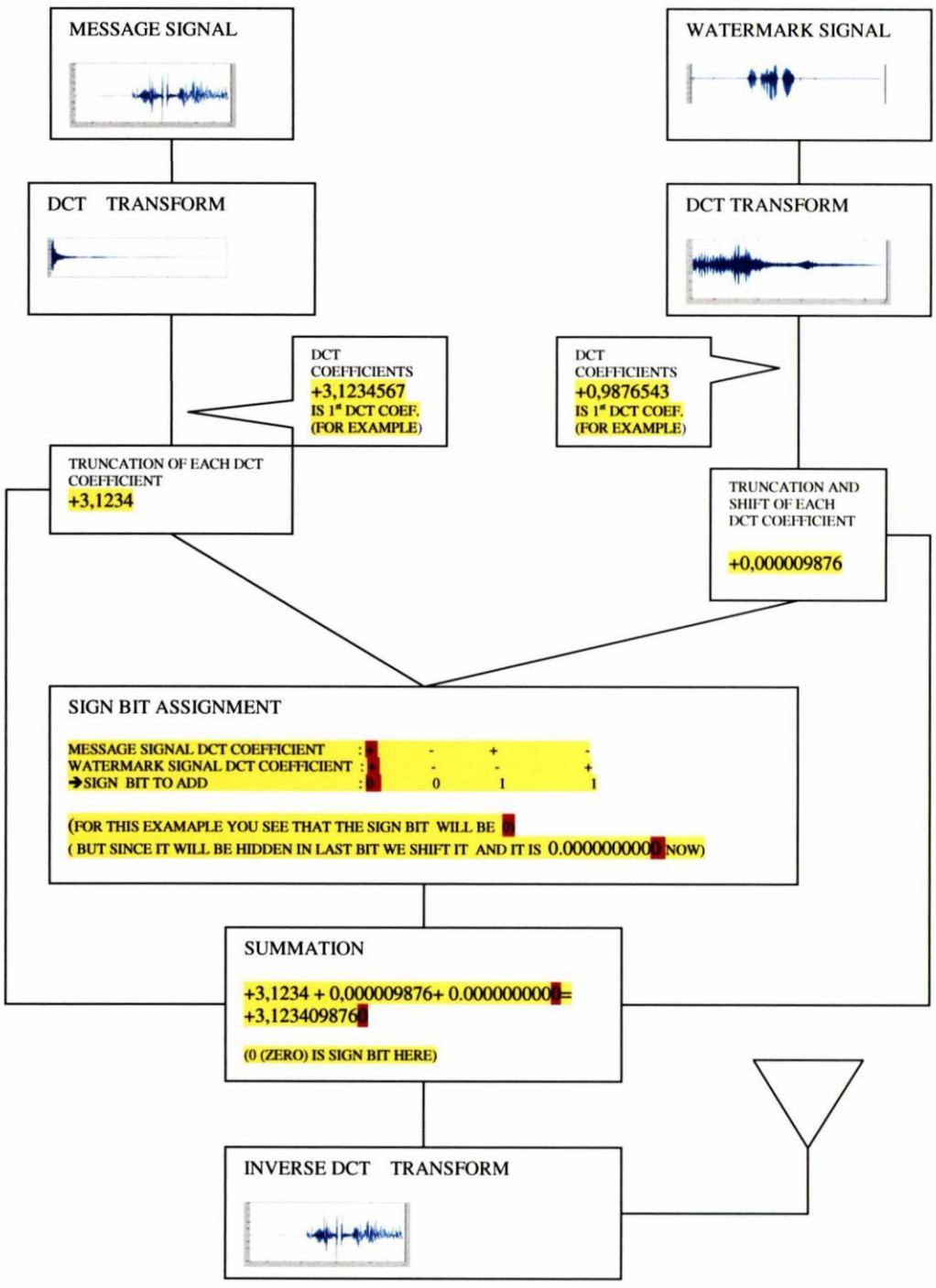


Figure 14: The watermark embedding module (at sender side).

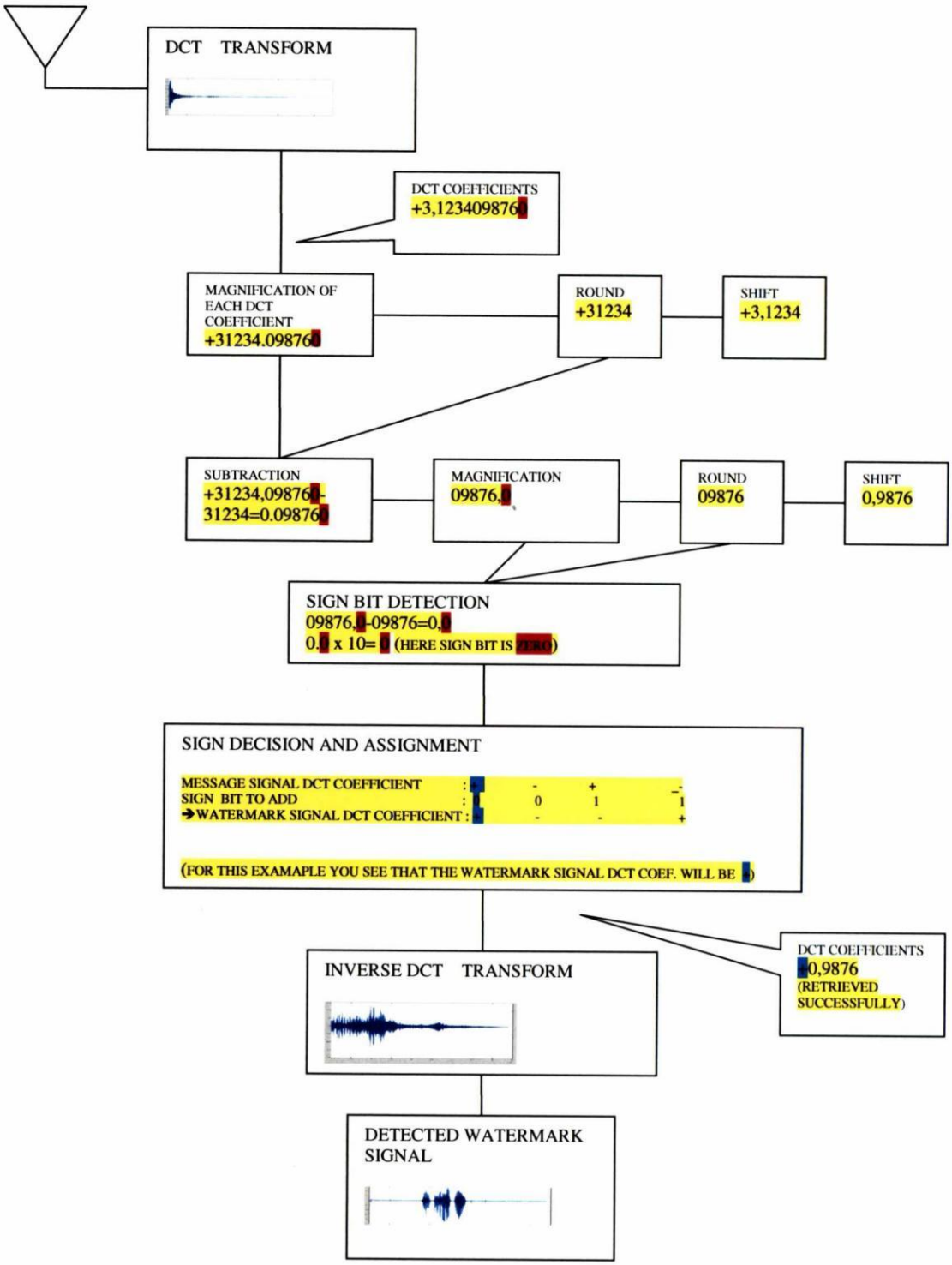


Figure 15: The watermark detection module (at receiver side).

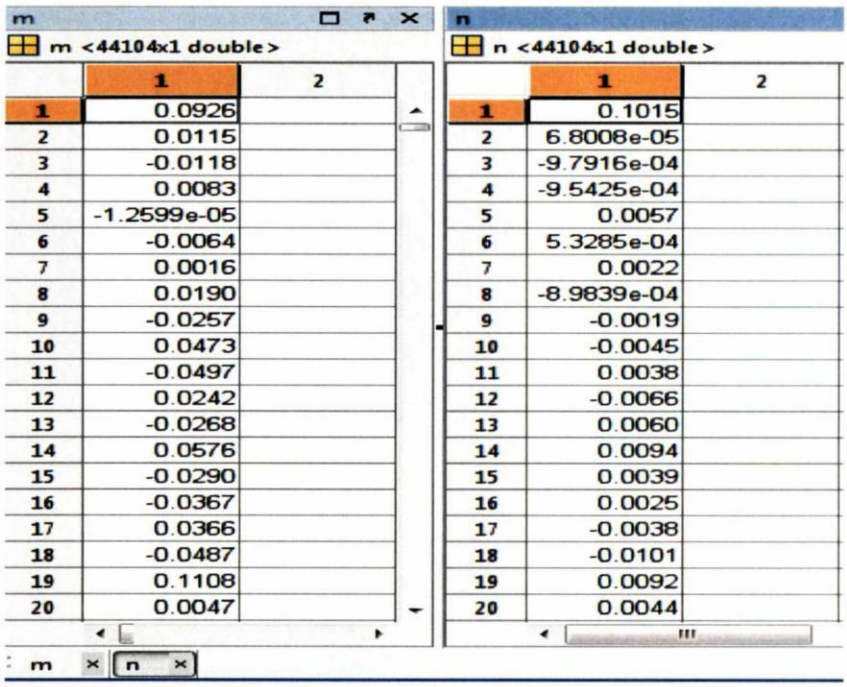
Let us use an example to clarify the process. The first DCT coefficient of the message signal is 3.1234567, and the first DCT coefficient of the watermark signal is 0.9876543. Since the coefficient does not change significantly after the 4th digit in the decimal fraction let us truncate digits after 4. Then, the first DCT coefficients of the message and watermark signals are, respectively, 3.1234 and 0.9876. Now the DCT coefficient of the watermark signal is shifted to the right by 5 digits which makes it 0.000009876. Concatenation is then a simple matter of adding the two coefficients, giving the new coefficient as: 3.1234000000 + 0.000009876 = 3.123409876. Here it is obvious that 3.1234 is derived from the first DCT coefficient of the message signal, and the rest, 09876, is derived from the first DCT coefficient of the watermark signal.

The last step is to assign the last digit associated with the sign of the DCT coefficient of the watermark. There are 4 possible cases for the DCT coefficients of the message and watermark signals: + + , - - , + - , and - + , where the '+' and '-' indicate the sign of the coefficients of the two signals. Since the watermark signal coefficients are concatenated with the message signal coefficients by shifting and addition, the sign of the watermark coefficient is needed to perform the correct addition operation. If the signs of the two coefficients are different, the addition operation would change the magnitude of the coefficient of the DCT coefficient of the original signal rather than just concatenating it. Hence, the sign bit needs to be appended to the transmission coefficent to ensure that the correct information is decoded at the receiver.

For example, when the DCT coefficient of the message signal is 3.1234567 and the DCT coefficient of the watermark signal is -0.9876543, the truncation, shift and add processes would produce: 3.1234000000 + (-0.000009876) = 3.123390124 which would

be interpreted as the DCT coefficient of the original signal, 3.1233, and the DCT coefficient of the watermark signal is interpreted as 9.0124 which is incorrect. Hence, the addition is performed as: 3.1234000000 + abs(-0.000009876), which produces the correct result, and a sign digit of 1 is appended in the leftmost position to indicate the change in sign between the two coefficients.

The experimental results are shown in Figure 16. These are the numeric values of the first 20 DCT coefficients of the original (left) and the watermark data (right).



	1	2
1	0.0926	
2	0.0115	
3	-0.0118	
4	0.0083	
5	-1.2599e-05	
6	-0.0064	
7	0.0016	
8	0.0190	
9	-0.0257	
10	0.0473	
11	-0.0497	
12	0.0242	
13	-0.0268	
14	0.0576	
15	-0.0290	
16	-0.0367	
17	0.0366	
18	-0.0487	
19	0.1108	
20	0.0047	

	1	2
1	0.1015	
2	6.8008e-05	
3	-9.7916e-04	
4	-9.5425e-04	
5	0.0057	
6	5.3285e-04	
7	0.0022	
8	-8.9839e-04	
9	-0.0019	
10	-0.0045	
11	0.0038	
12	-0.0066	
13	0.0060	
14	0.0094	
15	0.0039	
16	0.0025	
17	-0.0038	
18	-0.0101	
19	0.0092	
20	0.0044	

Figure 16: The message and the watermark signal DCT coefficients respectively.

At the decoder, the last digit determines the sign of the watermark coefficient. Look at the example and again let the DCT coefficient of the message signal be 3.1234567 and the DCT coefficient of the watermark signal be -0.9876543. The respective DCT

coefficient of the combined signal at the receiver is 3.1234098761. Here the last digit of 1 indicates that the sign of the watermark coefficient is opposite to the sign of the message coefficient, so it is negative. This means that during the detection operation, the value 09876 is to be read as -0.9876.

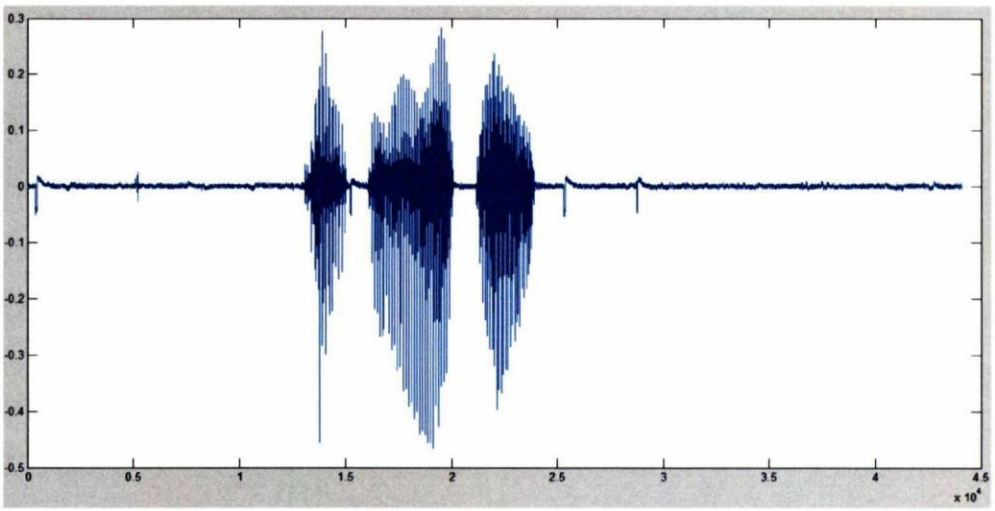


Figure 17: The detected watermark signal in time domain.

This procedure was used in the algorithm. After conducting tests it is observed that the watermark signal was detected and retrieved successfully. Figure 17 shows the retrieved watermark. Compare this with the original watermark signal shown in Figure 11. It is very difficult to visually compare the two figures and see any significant differences.

m

m <44104x1 double>

	1	2
1	0.0926	
2	0.0115	
3	-0.0118	
4	0.0083	
5	-1.2599e-05	
6	-0.0064	
7	0.0016	
8	0.0190	
9	-0.0257	
10	0.0473	
11	-0.0497	
12	0.0242	
13	-0.0268	
14	0.0576	
15	-0.0290	
16	-0.0367	
17	0.0366	
18	-0.0487	
19	0.1108	
20	0.0047	

a

a <44104x1 double>

	1	2
1	0.0925	
2	0.0114	
3	-0.0117	
4	0.0082	
5	-5.6100e-08	
6	-0.0064	
7	0.0016	
8	0.0189	
9	-0.0257	
10	0.0472	
11	-0.0496	
12	0.0242	
13	-0.0267	
14	0.0575	
15	-0.0289	
16	-0.0366	
17	0.0365	
18	-0.0487	
19	0.1107	
20	0.0046	

a m

Figure 18: The DCT coefficients of the original signal (left) and the watermarked signal (right).

The DCT coefficients of the original and detected watermark signals and their difference are shown in Figure 19, and the magnitude of DCT coefficients of the detected watermark signal is shown in Figure 20. The results in Figure 20 should be compared with the results in Figure 13. Errors due to truncation are evident. However, the key comparison between the original signal and the watermarked signal does not show perceptible visual or audible changes as shown in Figure 18.

n <44104x1 double>

	1	2
1	0.1015	
2	6.8008e-05	
3	-9.7916e-04	
4	-9.5425e-04	
5	0.0057	
6	5.3285e-04	
7	0.0022	
8	-8.9839e-04	
9	-0.0019	
10	-0.0045	
11	0.0038	
12	-0.0066	
13	0.0060	
14	0.0094	
15	0.0039	
16	0.0025	
17	-0.0038	
18	-0.0101	
19	0.0092	
20	0.0044	

d <44104x1 double>

	1	2
1	0.1014	
2	0	
3	-9.0000e-04	
4	-9.0000e-04	
5	0.0056	
6	5.0000e-04	
7	0.0022	
8	-8.0000e-04	
9	-0.0018	
10	-0.0044	
11	0.0037	
12	-0.0066	
13	0.0060	
14	0.0093	
15	0.0039	
16	0.0025	
17	-0.0038	
18	-0.0101	
19	0.0091	
20	0.0043	

dd <44104x1 double>

	1	2
1	6.0678e-05	
2	6.8008e-05	
3	-7.9158e-05	
4	-0.0019	
5	0.0113	
6	0.0010	
7	2.3150e-05	
8	-0.0017	
9	-8.6812e-05	
10	-0.0089	
11	0.0075	
12	-0.0132	
13	0.0120	
14	5.8237e-05	
15	0.0078	
16	0.0050	
17	-0.0076	
18	-1.4958e-05	
19	5.6532e-05	
20	7.2720e-05	

n × d × dd ×

Figure 19: The DCT coefficients of the original watermark signal (left) , the detected watermark signal (middle) and their difference (right).

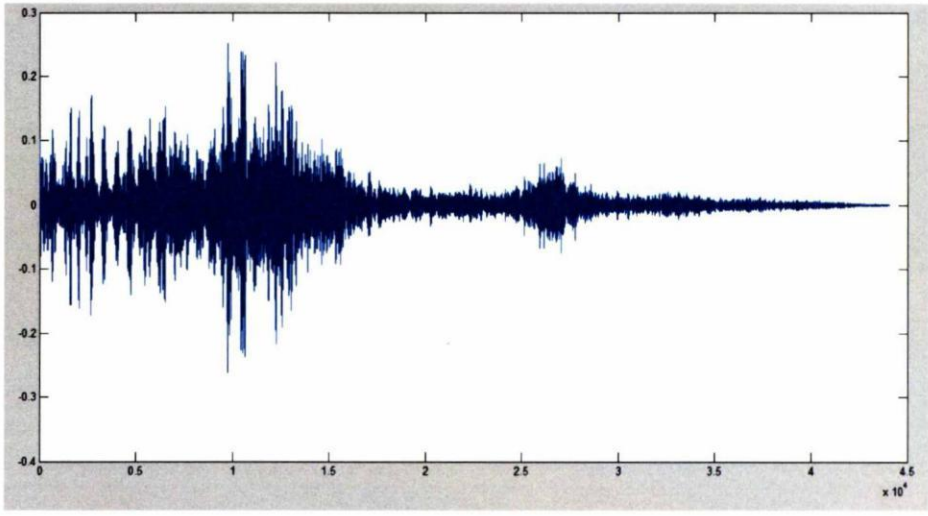


Figure 20: Magnitude of the detected watermark signal DCT coefficients.

CHAPTER 4

TRANSMISSION CONSIDERATIONS

The results in Chapter 3 show numerically and graphically that a watermark signal can be embedded in a message signal using the proposed method and be detected and retrieved succesfully. These results were also tested by listening to the original watermark and the retrieved watermark. However, the original algorithm testing was performed in a clean simulation environment. There was no signal fading or channel noise that could impact the watermarked signal and, hence, its DCT coefficients. In this chapter, we examine some of the issues related to real-world channels. We use the communication toolbox in Matlab for these tests.

The main idea is that in a communication system, when the transmitter side sends its message to the receiver side through a wired medium, the signal experiences some losses due to the length and the tranmission quality of the cable, or, if it is wireless transmission, some losses due to weather effects. Since the high frequency components dominantly characterize an audio signal, these components should be examined for both cropping and bandwidth effects.

In our study, the channel limitations are one of the major considerations and are simulated by a low pass filter. When the watermarked signal is filtered by a Butterworth lowpass filter,

$$H(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \ldots + a(n+1)z^{-n}} \quad . \tag{20}$$

where b and a are length n+1 row vectors that represent the filter coefficients in descending powers of z. It is observed that the watermark is detectable only if the cut-off frequency ω_c of the filter is $0.01F_s$, where F_s is the sampling frequency of the signal. This cut-off frequency corresponds to about 200 Hz if a first order filter is used or about 400 Hz for a second order Butterworth filter and about 600 Hz for a third order filter. Thus, the cut-off frequency can be computed as a function of the order of the filter using Equation 22:

$$\omega_c = 0.01nF_s \quad . \tag{21}$$

In order to test crop attacks, the combined signal is cropped at different rates. It is observed that the watermark signal is retrievable only if the rate is very close to 1 which is almost no cropping. With these results it is hard to say that the system is robust to crop attacks.

In this algorithm, we initially selected 4 digits to represent the DCT coefficients of the original and the watermark signals. We can change this selection to a lower resolution to match the low bandwidth of the transmission channel. However, each choice impacts the performance of the algorithm differently. If we use one or two digits of the DCT coefficients, like 0.0 or 0.9 for the previous example when the watermark signal DCT coefficient was 0.9876543, then the watermark signal is not retrievable. On the other hand, if we use three or more digits, i.e 0.98 or 0.987, the watermark signal is detectable by simulaton tool; however, it is more susceptible to channel constraints.

In the presence of white Gaussian noise (AWGN) which is simulated in Matlab, the watermark signal is retrievable at SNR greater than 135 dB for the case of the three digit DCT coefficients usage. It is detectable at 175 dB SNR for 4 digit representation of the

DCT coefficients and at 210 dB SNR for the 5 digit representation. Here we may conclude that each digit corresponds to requiring an increase in the SNR of approximately 35 dB SNR. This response can be explained by looking at the number of digits needed to represent the DCT coefficient as the resolution of the signal. Higher resolutions are more susceptible to channel errors and, hence, require a higher SNR for error-free signal reconstruction.

When music is used as the message and the watermark signal, approximately 5 dB lower SNR gives successful watermark retrieval, as shown in Table 1. It means that speech is more fragile to noise than music.

Table 1: The precision of the DCT coefficient as a function of the SNR and the type of audio signal.

SAMPLE SNR VALUES AT WHICH THE WATERMARK SIGNAL IS RETRIEVABLE IN THE PRESENCE OF AN AWGN		
DIGITS USED FOR DCT COEFFICENTS	**SNR (SPEECH)**	**SNR (MUSIC)**
3 (i.e., 3.12)	135 dB	129 dB
4 (i.e., 3.123)	175 dB	170 dB
5 (i.e., 3.1234)	210 dB	211 dB

Finally, it can be stated that the algorithm proposed in this study has very good theoretical performance but because of being susceptible especially to noise may not be easily implementable for real world applications.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis the main concern was to find a new solution to the problem of secure communications using digital audio watermarking. The novel approach presented in this thesis quantizes the DCT coefficients of the watermark signal that is to be hidden and embeds them into the DCT coefficients of the message, or host, signal. The results show that although the proposed algorithm has very good theoretical performance, it may not be easily implementable for real world applications since it is quite susceptible to noise.

As future work, we propose another approach based on attenuating, and then mixing, the DCT coefficients of the watermark and the original signal. The embedding algorithm will sum the host signal DCT coefficients with the watermark signal DCT coefficients as shown in Figure 21. With this method one may construct a non-blind detection algorithm which uses a host signal at the receiver to recover the watermark signal but which will require more bandwidth, or one may construct a blind algorithm which does not use host signal, as shown in Figure 22, for retrieval of the watermark signal. Although the latter technique will be more complicated to implement, both approaches would be less susceptible to noise and likely to be more robust than our proposed algorithm.

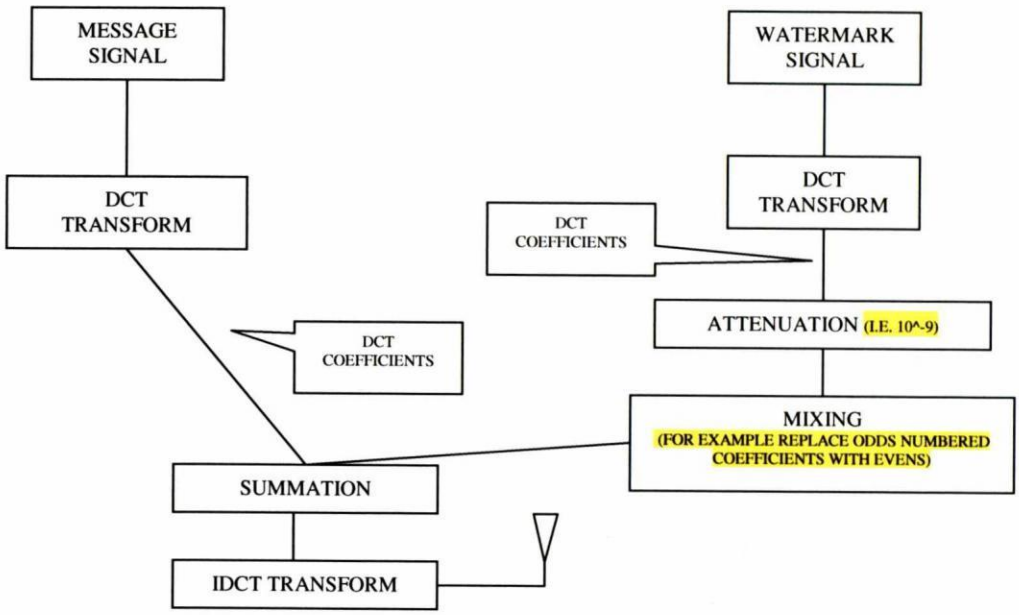


Figure 21: Future work algorithm for watermark embedding at sender side.

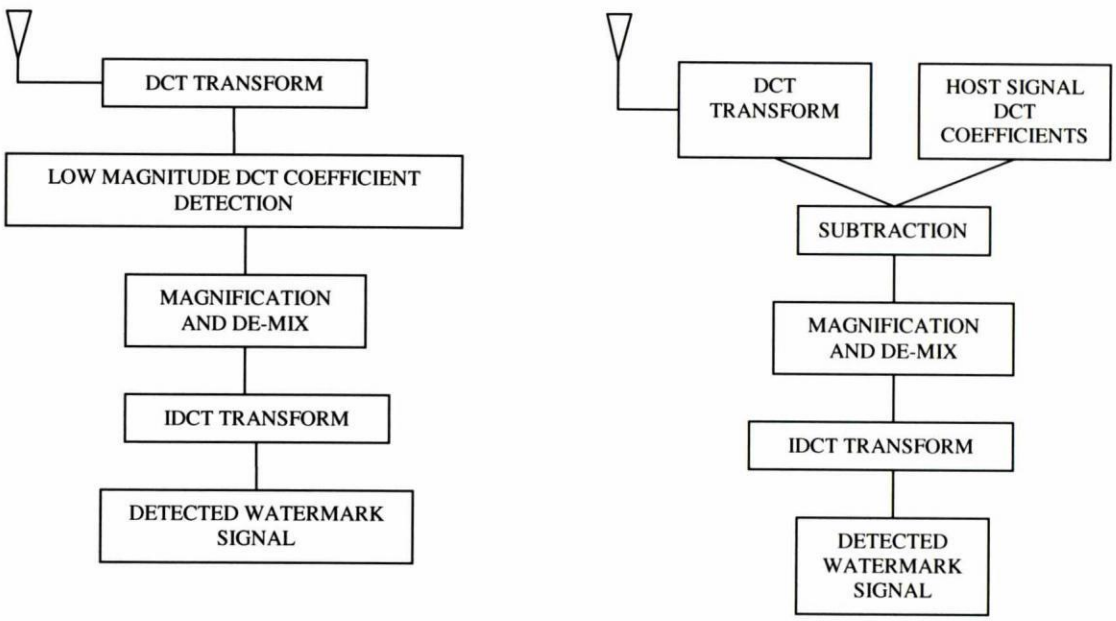


Figure 22: Future work algorithm for blind watermark detection (left) and non-blind detection (right) at receiver side.

REFERENCES

[1] Cox I. J., Miller M., Bloom J., Friedrich J., and Kalker T., *Digital Watermarking and Stenography*, Morgan Kaufmann Publishers, 2008.

[2] Alsalami M. A. T., and Al-Akaidi M. M., "Digital Audio Watermarking: Survey," *Proceedings of the 17th European Simulation Multiconference*, 2003.

[3] Kim H. J., Choi, Y. H., Seok, J. W., and Hong, J. W., "Audio Watermarking Techniques," *Intelligent Water Marking Techniques*, Vol.7, No:1, pp. 185-218, 2004.

[4] Wang X. Y., and Zhao H., "A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT," *IEEE Transactions on Signal Processing,* Vol. 54, No. 12, pp. 4835-4840, 2006.

[5] Bhat V. K., Sengupta I., and Das A., "Audio Watermarking Based on Mean Quantization in Cepstrum Domain," *Proceedings of the 16th International ADCOM Conference*, 2008.

[6] Ramalingam A., and Krishnan S, "Gaussian Mixture Modeling of Short-Time Fourier Transform Features for Audio Fingerprinting', *IEEE Transactions On Information Forensics And Security,* Vol. 1, No. 4, pp.457-463, 2006.

[7] Etisalat College of Engineering, 2004. Retrieved January 20, 2010 from Etisalat College website: http://www.emirates.org/ieee/events.html# Digital_Watermarking.

[8] Nguépi, N. S. A., *Digital Watermarking*, Seminar Series Selected Topics of IT Security, Summer Term 2007, Faculty of Security in Information Technology, Technical University of Darmstadt, 2007.

[9] Tao, T., "Fourier Transform," Department of Mathematics, UCLA, 1999.

[10] Yucel Z., "Watermarking Via Zero Assigned Filter Banks," MS Thesis, Bilkent University, Ankara/TURKEY, August 2005.

[11] Chikkerur S. S., Cartwright A. N., and Govindaraju V., "Fingerprint Image Enhancement Using STFT Analysis," *Pattern Recognition*, Vol. 40, No. 1, pp. 198-211, 2007.

[12] Khayam S.A., 'The Discrete Cosine Transform (DCT): Theory and Application', Seminar Series of Information Theory and Coding, Spring Term 2003, Department of ECE, Michigan State University, 2003.

[13] *MATLAB The Language of Technical Computing-MATLAB Graphics Reference Version 5*, December 1996, The MathWorks Inc., MA, USA.

APPENDIX

MATLAB CODE OF ALGORITHM

```
% SIGNALS

[x,Fs,Nbits] = wavread('mes.wav');
[y,Fs,Nbits] = wavread('water.wav');

%soundsc(y,Fs);
%plot(y);

%-------------------------------------------------

% FILTER

% [bf,af] = butter(1,0.01);    % n.degree LPF coeff. Wc=0.1*n*Fs
% Af1 = filter(bf,af,x);
% Af2 = filter(bf,af,y);
%
% %soundsc(Af1,Fs)
%
% Y1 = fft(x);
% Pyy1 = abs(Y1)/length(Y1) ;
% f1 = Fs * (0:length(Y1)/2) / length(Y1) ;
% lf1 = length(f1);
%
% Y2 = fft(Af1);
% Pyy2 = abs(Y2)/length(Y2) ;
% f2 = Fs * (0:length(Y2)/2) / length(Y2) ;
% lf2 = length(f2);
%
% subplot(221)
% plot(x)
% title('Original Signal')
% xlabel('time domain')
%
% t=0:44103;
% subplot(223)
% plot(f1,Pyy1(1:lf1))
% title('Original signal')
% xlabel('frequency Hz.')
%
% subplot(222)
% plot(Af1)
% title('signal after filter')
% xlabel('time domain')
%
% subplot(224)
% plot(f2,Pyy2(1:lf2))
% title('signal after filter')
% xlabel('frequency Hz.')
 %-------------------------------------------------
```

```
 % DCT

m=dct(x);
n=dct(y);

%--------------------------------------

% USING COMPRESSED DATA (ORIGINAL SIGNAL OR WATERMARK SIGNAL)

%N=max(size(n0));
%cr=1/3;
%n= n0(1:round(cr*N));
%n=[n;zeros(N-round(cr*N),1)];

%--------------------------------------

% WATERMARK EMBEDDING

m1=m*1000;
m2=fix(m1);
m3=m2/1000;

n1=n*1000;
n2=fix(n1);
n3=n2/1000;

n4=n3/10000;

for t=1:44104

    if m(t)<0 && n(t)>0
    n4(t)=-n4(t)-0.00000001;

    elseif m(t)>0 && n(t)<0
    n4(t)=-n4(t)+0.00000001;

    end

end

a=m3+n4;
% for t=1:441
% fprintf('%18.10f %18.10f %18.10f %18.10f %18.10f\n', a(t), m(t),
n(t), m3(t), n4(t))
% end

ai = idct(a);
%soundsc(ai,Fs);


%----------------------------------------
```

```
 % CROPPING

%N=max(size(ai));
%cr=1/20; % Crop rate

%Ac= ai(1:round(cr*N));
%Ac=[Ac;zeros(N-round(cr*N),1)];

%figure;
%subplot(2,1,1);
%plot(ai);
%title('Original data');
%subplot(2,1,2);
%title('Cropped data');

%-----------------------------------------------


% NOISE

An=awgn(ai,170);

%subplot(211)
%plot(dct(ai))
%title('Original Signal')
%xlabel('freq domain')

%subplot(212)
%plot(dct(An))
%title('signal after noýse')
%xlabel('freq domain')


%-----------------------------------------------

% WATERMARK DETECTION

b = dct(An); %use ai:for regular case
             %    Ac:for cropped signal
             %    Af:for filtered signal
             %    An:for noisy signal

a1 = (b*10.0^7);
a2=round(a1);
br=round(10*(abs(a1-a2)));

c=double(a2*10.0^-4);
c1=fix(c);
c2=c-c1;
d=double(c2*10.0);

dd=n-d;
max(dd);
min(dd);
```

```
for t=1:44104
    if (br(t)==1 && a(t)>0)
        d(t)=-d(t);
    elseif (br(t)==1 && a(t)<0)
        d(t)=-d(t);
    end

end


P=idct(d);
% for t=1:441
% fprintf('%18.10f %18.10f %18.10f %18.10f\n',  y(t), k(t), n(t), d(t))
% end

figure, plot(P);
soundsc(P,Fs)
```

VITA

Erol Duymaz

Phone: (757) 401-1892
erolduymaz@hotmail.com

Education:

B.S. Electrical and Electronics Engineering, 9th September University, Izmir, Turkey, 2002

Experience:

Officer, 2004 - 2010
Turkish Air Force

- Air Technical Schools Command, Izmir, Turkey, Officer Training, March 2004-August 2004.
- 1st Air Supply and Maintenance Center Command, Eskisehir, Turkey, Precision Measurement Equipment Laboratory Manager, September 2004-August 2008.
- Turkish Air Force Academy, Electronics Engineering Department, Istanbul, Turkey, MS Student, September 2008-September 2009.
- Old Dominion University ECE Department, Norfolk VA, USA, MS Student September 2009-May 2010.