Université d'Ottawa • University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Ronghui TU

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Master of Computer Science

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Semi-fragile Digital Audio Watermarking

J. Zhao

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

P. Liu

A. El Sadik

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

SEMI-FRAGILE DIGITAL AUDIO WATERMARKING

by

Ronghui Tu

A thesis submitted to
the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of
the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute of Computer Science
School of Information Technology and Engineering
University of Ottawa

Ottawa, Ontario, Canada

Your file Votre référence
ISBN: 0-612-89913-6
Our file Notre référence
ISBN: 0-612-89913-6

Acknowledgement

I would like to thank my supervisor, Dr. Jiying Zhao, for bring me into this interesting topic "digital watermarking", for his support and encouragement during my graduate study.

I would also like to extend my thanks to the members of the Multimedia Communications Research Laboratory for their friendship and suggestions.

Finally, I would like to thank my family for their continuous support in these years.

Abstract

The digital information has brought many changes to our life in recent decades. With the rapid growth of digital information, a lot of new techniques are coming into being. Digital watermarking is one of them. This thesis presents a semi-fragile audio watermarking scheme which can be applied to content authentication or copyright verification. The major contribution of this thesis is the introduction and implementation of a unified copyright verification and content authentication algorithm for audio signal.

In our approach, we embed the watermark in the discrete wavelet domain of an audio by using quantization technique. The discrete wavelet domain has both the spatial and frequency information which make it possible to detect various modifications. The advantage of using quantization technique is that we could extract the watermark without using original audio.

In the procedure of watermark extraction, some signal processing operations may change the length of testing audio. We employ a matching filter to locate the start point of the watermark. This filter can be generated during the embedding procedure. The cost of the filter is very small.

We conduct several experiments for evaluating the performance of this new watermarking technique on both of two applications - content authentication and copyright verification. Our experimental results show that this scheme is robust to mp3 compres-

sion, additive noise, and filtering attacks. At the same time, the embedded watermark can also be used to check whether the audio content has been modified or not. If the audio is determined to be modified, an assistant program can be invoked to find out where the modification is.

Contents

List of Figures

List of Tables

Acronyms

DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FIR	Finite Impulse Response
HAS	Human Auditory System
HVS	Human Visual System
IDWT	Inverse Discrete Wavelet Transform
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bits
MP3	MPEG Audio Layer 3
MPEG	Moving Picture Experts Group
PDA	Personal Digital Assistant
PN	Pseudorandom Noise
PSNR	Peak Signal-to-Noise Ratio
SDMI	Secure Digital Music Initiative
STFT	Short-Time Fourier Transform

Chapter 1

Introduction

1.1 Background

With the rapid growth of the Internet and personal computers, the digital format of media becomes more and more popular. Along with powerful software, new devices, such as digital camera, mp3 player and PDA (Personal Digital Assistant), have made consumers convenient to create, manipulate and store the digital multimedia data. Internet and wireless network provide a channel to transmit and to exchange these multimedia information. In the recent decades, these new formats of data have brought many changes to our life. However, they also pose the danger of illegal copy, redistribution and various malicious attacks. Therefore the protection of ownership and the prevention of unauthorized tampering of digital multimedia data become important concerns.

The most common method for security issue is to use cryptographic techniques. In cryptography, the information is encrypted before its transmission and it can be viewed after decryption. Once the data are decrypted, the digital signature is removed and there is no more proof of ownership. In other words, cryptography can protect digital

media only in transit, but once decrypted, the media has no further protection.

Watermarking is a new technique that has potential to protect digital data even after they are decrypted. A watermark is a data stream embedded into original signal imperceptibly. Once a watermark is embedded into the host signal, it is never removed during normal usage. A watermark can be used to identify copyright holder, to prevent illegal copy and to verify whether the content is modified. The host signal for a watermark can be in various formats, such as audio, image or video.

The concept of digital watermarking is derived from steganography, which is a term from Greek language and means "covered writing". During thousands of years, people have hidden information by various methods[1]. For example, ancient Greeks wrote text on wax-covered tablets. Invisible inks also offered a common form of invisible writing.

Paper watermarks are designs or patterns put into paper during its production, by making the layer of pulp thinner or thicker when it is still wet, and hence, the name watermark. Paper watermark can be seen when holding the paper against the light or, in some cases, over a black surface. Usually, they show the manufacturer's name, and geometric designs. The objective of watermarks in paper is, essentially, identifying the paper, as a signature of the manufacturer, or as a security measure to avoid forgery of important documents as bank notes, passports, etc. A good example for paper watermark is a bill in some countries.

The idea of using digital watermarking for copyright protection arose in 1994 by Brassil *et al.*[2]. During the following years, digital watermarking has gained a lot of attention and has evolved quickly. A lot of practical working methods and systems have been developed.

Both steganography and watermarking describe technologies that are used for hiding secret information in cover data. However, steganography stands for techniques in general that allow secrete communication, usually by embedding or hiding the secret

information in other, unsuspected data. Steganographic methods generally do rely on the assumption that the existence of the covert communication is unknown to third parties and they are mainly used in secret point-to-point communication between trusting parties. As a result, steganographic methods are in general not robust.

Watermarking, as opposed to steganography, has the additional notion of robustness against attacks. Even if the existence of the hidden information is known, it is difficult for an attacker to destroy the embedded watermark, even if the algorithmic principle of the watermarking method is public.

There are many ways to categorize the watermarking techniques. The straightforward way is according to the type of the host signal. The host multimedia could be plain text, audio, image, video and 3-D graphics.

With the concern of embedding techniques, watermarks could be classified as spatial watermarks and spectral watermarks. The spatial watermarking techniques embed an invisible watermark into the host signal in its spatial domain. This technique is easy to implement, but the watermark is not robust enough against attacks. In spectral watermarking, the watermark is embedded in a certain transform domain. The transform can be discrete Fourier transform (DFT), discrete cosine transform (DCT) or wavelet transform.

From the view of different applications, watermarks could be divided into two groups: robust watermarks and fragile watermarks. Robust watermarks cannot be removed by common signal processing operations. The watermark is used to prove ownership and to detect unauthorized copies of media. Fragile watermarks are used for content authentication[3]. In this case, watermarks are changed if the host media is modified. The authentication will be failed even if there is a small change of the watermark.

In our research work, we focus on audio watermarking and develop a multipurpose

scheme for both copyright protection and content authentication.

1.2 Contributions

This thesis presents a semi-fragile audio watermarking technique which embeds the watermark in the discrete wavelet domain of an audio by quantizing the selected coefficients. This approach could be applied for both content authentication and copyright verification.

In image authentication, quite a number of methods have been proposed. However, to our best knowledge, only one of them is oriented to audio authentication [4]. The major contribution of this thesis is that we introduce the novel authentication idea into audio field and present a scheme for both audio copyrighting and authentication. In our scheme, the watermark is placed in the discrete wavelet domain, which has both spatial and frequency information. With this characteristic, it is possible for our approach to detect various modifications such as substitution of data, filtering and mp3 compression. In addition, as a semi-fragile watermarking, this algorithm is also designed robust against some attacks. Therefore it can be used in copyright verification as well.

A major advantage of using quantization technique in this approach is that the watermark extraction could be performed without using original audio. The quantization parameters used in the algorithm are user-defined. Different value of the quantization parameters affects the robustness of the watermark.We make a discussion on how to choose a proper parameter in this thesis.

In watermark extraction procedure, a matching filter is employed to find watermark start point in watermarked audio.

We implement the algorithm and show that it is robust against additive noise attack,

filtering and mp3 attacks. At the same time, it shows that the watermark can be used for authentication as well. Another advantage of our scheme is that we can embed a large amount of data into the audio.

For authentication purpose, if the audio is failed in authentication, an assistant program is invoked to find out where the modified content is.

Publications resulting from this research:

1. Ronghui Tu and Jiying Zhao, A Novel Semi-Fragile Audio Watermarking Scheme, IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications, Ottawa, Ontario, Canada, 20-21 September 2003.

2. Ronghui Tu and Jiying Zhao, A Semi-Fragile Audio Watermarking Scheme Based on Wavelet Transform and Quantization. Submitted to Signal Processing.

1.3 Thesis Organization

In Chapter 2, we give a brief introduction on digital watermarking techniques and overview the various existing approaches. In Chapter 3, we make an introduction to several important techniques which are used in our scheme. Chapter 4 is organized to describe our watermarking scheme and some implementation strategies. Chapter 5 and Chapter 6 give experimental results and conclusions, respectively.

Chapter 2

Overview of Digital Watermarking

2.1 Digital Watermarking Framework

In general, a digital watermarking system consists of an embedder, a detector and a watermark, as shown in Fig.2.1. The embedder takes two inputs. One is a watermark which is only known by author, and the other is an original signal in which we want to embed the watermark. In most cases, the watermark is a collection of bits, which may come from an encoded character string, from a pattern, from some executable agents, or others [5]. The watermark embedder incorporates the watermark W into the signal A and generates a watermarked signal A'. The embedding algorithm is defined as Equ.2.1[6]:

$$A' = E(A, W) \tag{2.1}$$

The watermarked signal A' is then delivered to the communication network where the signal may be distorted by some malicious attacks such as compression, filtering and additive noise. Later, the watermarked signal is presented as the input to the detector.

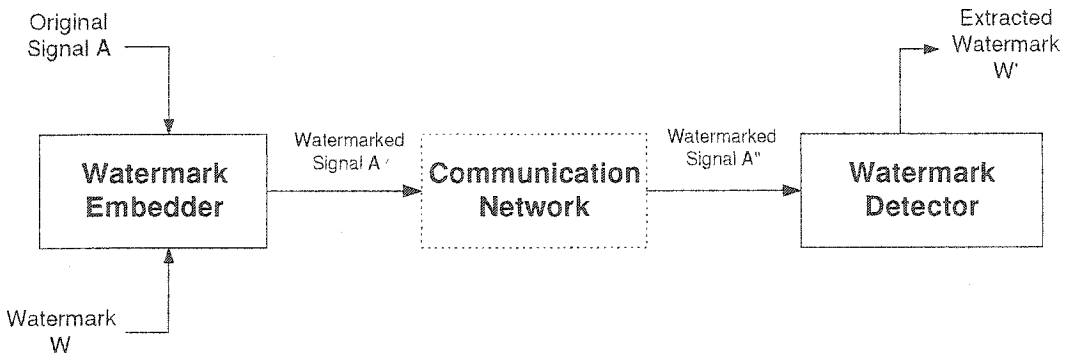


Figure 2.1: A generic watermarking system.

The watermark decoder extracts the watermark from the watermarked signal which may or may not have undergone attacks. The input of the decoder A'' may differ from the original signal A and the watermarked signal A'. The difference is referred to as noise. The decoder can be defined as Equ.2.2[6]:

$$W' = D(A'', A) \tag{2.2}$$

The extracted watermark W' will be compared with the original watermark W to check whether the signal is encoded with a watermark or not.

2.2 Properties of Digital Watermarking Systems

There are a number of properties that a watermarking system should achieve [7][8]. But different applications will have different properties. Therefore, there is no unique set of properties that all watermarking techniques must satisfy. In this section, we review some important properties.

2.2.1 Robustness

Robustness refers to the ability of detecting the watermark after common signal processing such as filtering, lossy compression, distortions and additive noise. Not all watermarking applications require watermark to be robust to all possible signal processing operations. It is application dependent.

2.2.2 Imperceptibility

The embedded watermark should not be noticed by users, and it should not destroy the quality of the original signal. In other words, the data embedding process should not introduce any perceptible artifacts into the host data.

2.2.3 Data Payload

Data payload is the number of bits that a watermark encodes within a host signal. Different applications may require different data payload. Copyright verification, inserting a serial number or author identification, may require only a small amount of information to be incorporated in the signal. In contract, television broadcast monitoring will require a watermark segment in each second.

2.2.4 Blind Detection

In some applications, the watermark detector does not require any information related to the original signal, which is referred to as blind detection. In the watermarking literature, systems that use blind detection are called public watermarking systems, whereas those that require access to the original signal are called private watermarking systems.

2.2.5 Security

The security of a watermark means that an unauthorized user can neither embed a watermark nor can he detect if a given signal contains a watermark. If the security is required in a watermarking system, at least one secret key has to be used for the embedding and extracting process. For example, in many schemes, the embedded watermarks are pseudorandom signals. In this case, the seed of the pseudorandom number generator may be used as secret key.

2.2.6 Computational Cost

Different applications require the watermark to be done at different speeds and complexities. In broadcast monitoring, both embedders and detectors are required to work in real time. The embedders must not slow down the media production schedule, and the detector must keep up with the real time broadcasting. On the other hand, speed is not an issue when tracking illegal copies, watermark retrieval is needed only when copyright violations have to be investigated. Here the watermark insertion should be of low complexity, but retrieval operation should be more complex in order to counter all possible kinds of attacks on the watermark.

2.2.7 Tradeoffs between Properties

The importance of each property is dependent on the requirements of applications. In fact, it is not possible to achieve all of these properties in one watermarking system at the same time. There are some tradeoffs among them.

In order to make a watermark difficult to be removed, the watermark must be placed in the perceptually "important" parts of a host signal. For example, the audio watermark should be placed in the portions of an audio that most affect human

hearing. However, placing the watermark in the "important" parts goes against the goal of reducing the perceptual effect of the watermark. Thus, the robustness and the imperceptibility of the watermark cannot be maximized at the same time. The similar conflict occurs between the data payload and the imperceptibility of the watermark. The more bits we embed into the signal, the more likely people will notice the presence of the watermark. Therefore, we should optimize these properties according to the specified applications.

2.3 Applications

There are quite a number of watermarking systems developed based on different applications [9].

2.3.1 Copyright Protection

One of the main reasons of introducing digital watermarking is for copyright protection. The idea is to embed information about the copyright owner into the data to prevent other parties from claiming to be the rightful owners of the data. The watermark used for this purpose is known only by the authors of the digital source and is supposed to be very robust against various attacks intended to remove the watermark. They also have to be unambiguous and still resolve rightful ownership after other parties embedding additional watermarks. The data payload for this application does not have to be high.

2.3.2 Content Authentication

Signature information is embedded to source, and later is used to verify whether the content has been tampered or not. In this application, the robustness of the watermark

is not a concern. If the source is modified, the watermark along with it is also modified. This kind of watermark is referred to as fragile watermark.

2.3.3 Copy Control

The embedded watermark in this case is used to tell recording equipment what content may not be recorded. If the recording devices were fitted with a watermarking detector, the devices could be made to prohibit recording whenever a never-copy watermark is detected at its input. An example of such a system in audio is SDMI system [10].

2.3.4 Fingerprinting

In this application, the watermark is used to trace the originator or recipients of a particular copy of multimedia data. For example, the owner of a multimedia product could place a different watermark in each copy. If the product is subsequently redistributed illegally, the owner could find out who should be responsible. Therefore, the watermark could identify people who obtain content legally but illegally redistribute it.

2.4 Attacks

One categorization of the wide class of existing attacks contains four types of attacks: removal attacks, geometric attacks, cryptographic attacks, and protocol attacks [11][12]. Fig.2.2 shows the classification of attacks on digital watermarks.

2.4.1 Removal Attacks

The goal of removal attacks is to remove the watermark from the watermarked signal completely without cracking the security of the watermarking system. This category

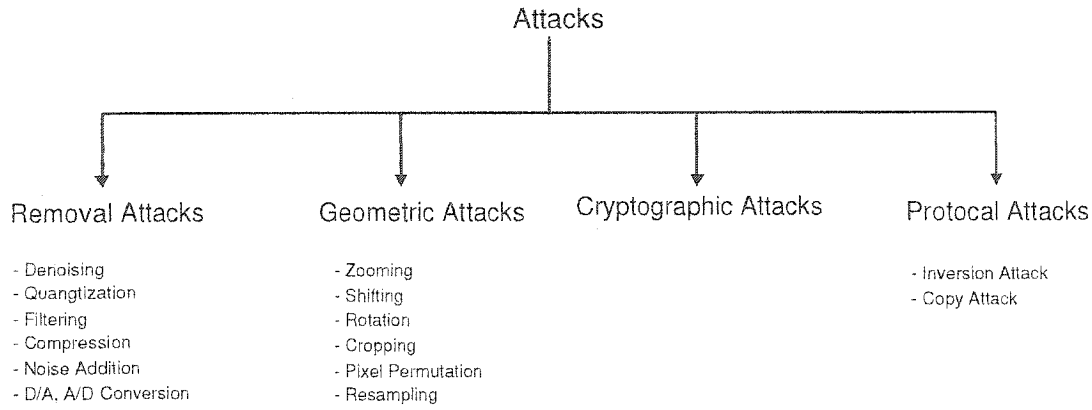


Figure 2.2: Classification of attacks on digital watermarks.

includes denoising, quantization, remodulation, and collusion attacks. Not all of these attacks could remove the watermark completely, but they may damage the watermark information significantly.

2.4.2 Geometric Attacks

In contrast to removal attacks, geometric attacks do not actually remove the watermark itself, but to distort the synchronization between detector and the watermark. The detector could recover the embedded watermark information when perfect synchronization is achieved. However, the complexity of achieving synchronization is too great to be practical. For Audio watermarking, the best known benchmark tool, Stirmark [13], integrates a variety of geometric attacks such as time stretch, cut samples, zero-cross-insertion and so on.

2.4.3 Cryptographic Attacks

Cryptographic attacks aim at cracking the security methods of the watermarking schemes and thus finding a way to remove the embedded watermark or to embed

misleading watermarks. One such technique is exhaustive search for the embedded information. Another attack is to create a non-watermarked signal when a watermark detector is available. However, these attacks have some restrictions due to their high computational complexities.

2.4.4 Protocol Attacks

In this category, there are two types of attacks. One is inversion attack. The idea behind inversion attack is that the attacker embeds his own watermark from the watermarked data and claims to be the owner of the data. This can create ambiguity with respect to true ownership of the data. Another protocol attack is copy attack. The goal is to estimate a watermark from watermarked data and copy it to some other data.

2.5 A Benchmark Tool: Stirmark

In the previous section, we reviewed various attacks against the watermark system. With the growing number of attacks, it is necessary to develop a benchmark for evaluating different watermarking algorithms. For image, there are several benchmark tools such as Checkmark [14] benchmark and Certimark [15]. For audio signals, the well-known benchmark is Stirmark [16].

Stirmark was the first benchmark tool developed at the University of Cambridge for digital watermarking technologies. Given a watermarked input audio, Stirmark generates a number of audio modifications which can then be used to verify if the embedded watermark can still be detected or not. Audio attacks implemented in Stirmark 3.1 include: adding noise attack, filtering, inversion, normalization, compression, changing the loudness of an audio, adding samples and cutting samples, time stretch and so on.

2.6 Literature Review

2.6.1 An Introduction to Audio Watermarking Algorithms

The properties and characteristics described above are common to both image and audio watermarking systems. However, audio and image watermarking systems exhibit significant differences. First of all, images are two-dimensional signals which may face more attacks such as rotation and scaling. Another difference lies in the domain where watermark embedding takes place. Audio watermarking methods usually work on time or frequency domain, while image watermarking methods often work on other domains, e.g., DCT domain which is closely related to JPEG compression standard or wavelet transform domain. Finally, due to the difference between human visual system (HVS) and human auditory system (HAS), different masking principles should be taken into account. In general, HAS is more sensitive to distortions than the visual system. Therefore, it is a challenge to make imperceptible audio watermarks.

A review of existing multimedia watermarking techniques can be found in [17]. The review shows that research on audio watermarking is not as mature as image watermarking technique. However, with the invention of efficient audio compression algorithms, the distribution of digital music through the internet is a tendency. The introduction of audio watermarking technique has become a matter of great commercial importance. Some other audio watermarking review papers are [18], [19], [20] and [21].

The simplest method for audio watermarking was presented by Turner [22]. In his method, the watermark is inserted into a digital audio signal by substituting the least significant bits (LSB) of randomly selected audio samples with the watermark bits. A large amount of data can be embedded into audio signal with this approach. However, the major disadvantage is its poor immunity to manipulations. The watermark can be destroyed by many signal-processing attacks.

Gruhl *et al.*[23] proposed a method which embeds watermark into audio by introducing an echo. The value of a hidden datum corresponds to the time delay of the echo and its amplitude. In this algorithm, it adds a repeated version of a component of the audio with small enough offset, initial amplitude and decay rate to make the echo inaudible. The offset between the original and the echo decreases, the two signals blend. At a certain point, the human ear cannot distinguish between the two signals. The extraction is based on the autocorrelation of the ceptrum of the embedded signal. The decision of the time delay can be made by examining the position of a spike that appears in the autocorrelation diagram. Echo hiding can effectively place imperceptible information into an audio data stream. But the watermark embedding process is signal dependent.

Ko *et al.*[24] combined echo hiding and spread spectrum technique. In this approach, an echo is temporally spread in the impulse response domain using a PN sequence, which also acts as a secret key to detect the embedded information from the watermarked signal. By spreading an echo, the amplitude of each component of the echo becomes small. However, It still maintains a high detection ratio. Another advantage is that it is possible to embed multi-information using uncorrelated PN sequences, which is useful for multi-authorities.

Hyen *et al.*[25] proposed multiple echo kernel in watermarking system. This multiple echo kernel comprises both positive and negative pulses but with different offsets. A negative echo has negative initial amplitude and, thus, does not add the copied signal but subtract it from host signal. Subjective evaluation and robustness tests in their research showed that this method is possible to embed echoes whose energies are two times larger than conventional approaches and, thus, to acquire corresponding robustness without degrading host signal quality.

Say *et al.*[26] proposed a scheme which combines echo hiding and masking technique.

Unlike conventional echo hiding techniques, the amplitude of echoes are set below a mask. In this way, it can make sure that the embedded watermark is under the masking threshold of human auditory system. In this approach, the echo kernel could be only one echo, or it can contain several echoes. This method is robust to common signal processing operations such as noise addition, re-sampling, cropping, filtering and MPEG coding.

Spread spectrum is the most popular method in audio watermarking. The main advantages are that the watermark detection does not require original audio, and that it is difficult to extract the hidden data using optimal statistical analysis under certain conditions[27]. The basic spread spectrum technique is designed to encode a stream of information by spreading the encoded data across as much of the frequency spectrum as possible. In the case of audio, the signal will be spread to the entire audible spectrum. Spread spectrum technique could be used either in time domain or in frequency domain.

Cox *et al.*[28] first introduced spread spectrum into the watermarking system. In their approach, the host audio is modulated with the watermark and a PN sequence. As a consequence, the frequency spectrum of the data is spread over the available frequency band. The PN sequence appears as random noise in the frequency domain. The resulting output sequence is attenuated and added back to original audio signal. In this approach, a scalar parameter α is introduced, which determines the extent to which the watermark X alters the sequence V using one of the following three formula:

$$v_i' = v_i + \alpha x_i \tag{2.3}$$

$$v_i' = v_i(1 + \alpha x_i) \tag{2.4}$$

$$v_i' = v_i(e^{\alpha x_i}) \tag{2.5}$$

In spread spectrum technology, Equ.2.4 is mostly used. A single scaling parameter α may not be applicable for perturbing all of the values v_i, since different spectral components may exhibit more or less tolerance to modification. Using multiple scaling parameters $\alpha_1 \ldots \alpha_i$ and using update rules such as $v_i' = v_i(1 + \alpha_i x_i)$ might be more appropriate. In the extraction procedure, the same PN sequence used in embedding will be synchronously multiplied with the embedded signal. The similarity between the extracted watermark X^* and the original watermark X is measured as follows:

$$sim(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X}} \tag{2.6}$$

The basic spread spectrum technique has many deficiencies. First, the marked signal and the watermark have to be perfectly synchronized at watermark detection. Next, to achieve a sufficiently small error probability, the length of the watermark should be sufficient long.

Bassia *et al.* [29] used a similar embedding approach as basic spread spectrum. However, watermark's inaudibility is achieved via noise shaping using a Hamming window. Detection involves a correlation of the original watermark with the watermarked signal evaluated for all possible circular shifts of the watermarked signal. This provides robustness against synchronization attacks but at the cost of much greater computational complexity.

Kirovski *et al.* [30] developed a set of technologies to improve the effectiveness of the embedding and detecting watermarks in audio. Watermark robustness is enabled using block repetition coding for prevention against de-synchronization attacks and psychoacoustic frequency masking. In [30], the authors indicated that the watermark detector should correlate only the audible frequency magnitudes with the watermark, because the inaudible portions of the frequency spectrum are significantly more susceptible

to attack noise. That reduces the effective watermark length, because the inaudible portion often dominates the frequency spectrum of an audio signal. Therefore, they use a simple psycho-acoustic frequency masking model to quantify the audibility of a particular frequency component.

Some other spread spectrum papers in audio watermarking technique are [31], [32], [33], and so on.

Boney *et al.* [34] first used masking phenomenon to embed watermarks into the audio. Audio masking is the effect by which a faint but audible sound becomes inaudible in the presence of another louder audible sound. The masking effect depends on the spectral and temporal characteristics of both the masked signal and the masker. Frequency masking refers to masking between frequency components in the audio signal. If two signals, which occur simultaneously, are close together in frequency, the stronger masking signal may make the weaker signal inaudible. The masking threshold of a masker depends on the frequency, sound pressure level, and tone-like or noise-like characteristics of both the masker and the masked signal. It is easier for a broadband noise to mask a tonal, than for a tonal signal to mask out a broadband noise. Moreover, higher frequency signals are more easily masked. Temporal masking refers to both pre-masking and post-masking. Pre-masking effects make weaker signals inaudible before the stronger masker is turned on, and post-making effects make weaker signals inaudible after the stronger masker is turned off.

In Boney's approach, a PN sequence is first generated. A masking threshold of the audio signal is calculated using MPEG Audio Psychoacoustic Model. The masking threshold is determined on consecutive audio segments of 512 samples. The PN sequence is then filtered with an approximate masking filter in order to ensure that the spectrum of the watermark is below the masking threshold. This approach shows robustness to MP3 compression attacks.

In [35] and [36], temporal masking is employed for embedding an imperceptible watermark.

In the next category of audio watermarking techniques, the watermark is embedded in a certain transform domain. The transforms can be Discrete Fourier Transform (DFT), or Discrete Wavelet Transform (DWT). In these approaches, the amplitude or phase of the transformed coefficients is modified with some specified amount in order to carry watermark information.

[6] presented a form of audio watermarking using phase modulation for proof of ownership applications. The key to maintain phase shift inaudibility is to keep the absolute phase shift small. In this approach, a hidden datum is represented by a particular phase or phase change in the phase spectrum. The watermark embedding works by substituting the phase of an initial audio segment with a reference phase that represents the watermarking data. The phase of subsequent segments is adjusted in order to preserve the relative phase between segments. For the extraction process, the hidden data can be obtained by detecting the phase values from the phase spectrum of the first segment. Phase coding can be used in both analog and digital modes but it is sensitive to most audio compressing algorithms.

Kim *et al.*[37] embedded watermark bits by using the patchwork algorithm on discrete wavelet domain. In this approach, the synchronization bits are used so that it can achieve a fast synchronization between the watermark embedding and detection parts without original audio signals. Several simulation results show that this method is robust against various signal manipulations such as MP3 compression and time scale modification.

Lee *et al.*[38] proposed a digital audio watermarking technique in the cepstrum domain. The authors investigated a spread spectrum technique to insert a watermark into the cepstral component of an audio signal. In the embedding process, the watermark

is hidden under frequency masking threshold. Experiments show that this scheme is robust to MPEG audio coding and additive noise.

In theoretical area, there are quite a numbers of approaches for audio watermarking system, especially in the application of copyright protection. In real world, there are also several developed systems. SDMI is one of the organizations focusing on watermarking techniques.

The Secure Digital Music Initiative (SDMI)[10] is a forum for examining technology which provides some security features for digital music. It also focuses on the techniques of copyright protection for next-generation portable digital music devices. This forum has brought together more than 200 companies and organizations in information technology, consumer electronics, security technology, and the recording industry. SDMI work is defined in two phases. Phase I screening looks for a watermark in the content but allows all music that is compatible with the device to be playable. Phase II will incorporate watermark detection which will allow new releases to play while filtering out pirated copies of music.

These various digital audio watermarking techniques focus on the robustness of watermark. They are used in the application of copyright verification. Content authentication is another important branch for watermarking applications. However, researches on audio content authentication are not well conducted. In the next section, we will review some approaches for image authentication.

2.6.2 Watermarking Systems for Authentication

In this section, we will focus on various watermarking systems for the application of content authentication. However, most of these systems are designed for image.

The main requirement for authentication systems is that modifications which do

modify the content will cause the data to be not authentic.

A number of authentication schemes have been proposed for image. In [39], Friedman described a "trustworthy digital camera" in which a digital camera image is passed through a hash function and then is encrypted using the photographer's private key to produce a piece of authentication data separated from the image. These data are used in conjunction with the image to ensure that no tampering has occurred. Specifically, the photographer's public key is used to decrypt the hashed original image and the result is compared to the hashed version of the received image to ensure authentication. In [40], Walton proposed a technique in which a separate piece of data is not required for authentication. The method requires the calculation of the checksums of the seven most significant bits of the image, so that they may be embedded into randomly selected least significant bits. These two techniques are focused on detecting whether an image was tampered with or not. However, they do not clearly specify how and where the image was changed.

Wu and Liu [41] proposed a watermarking scheme for image authentication which was based on table look-up in frequency domain. The table maps every possible value of JPEG coefficient randomly to 1 or 0 with the constraint that runs of 1 and 0 are limited in length. To embed a 1 in a coefficient, the coefficient is unchanged if the entry of the table corresponding to that coefficient is also a 1. If the entry of the table is a 0, then the coefficient is changed to its nearest neighboring values for which the entry is 1. The embedding of a 0 is similar. This process can be abstracted into the following formula where v_i is the original coefficient, v_i' is the marked one, b_i is the bit to be

embedded in, and $LUT(\cdot)$ is the mapping by look-up table:

$$v_i' = \begin{cases} v_i & if LUT(v_i = b_i) \\ v_i + \delta & if LUT(v_i \neq b_i), and \\ & \delta = min_{|d|}\{d \in Z : LUT(v_i + d) = b_i\} \end{cases} \tag{2.7}$$

The extraction of the signature is simply by looking up the table. That is,

$$\hat{b_i} = LUT(v_i') \tag{2.8}$$

where $\hat{b_i}$ is the extracted bit. This scheme can be used to detect tempering of the marked image and can locate where the tempering has occurred.

Kundur [42] presented a fragile watermarking approach which embeds a watermark in the discrete wavelet domain of an image. In this approach, the discrete wavelet decomposition of a host image is first computed. A user-defined coefficient selection key is then employed. The binary watermark bit is embedded into the selected coefficient through an appropriate quantization procedure. Finally, the corresponding inverse wavelet transform is computed to form the tamper-proved image. In the extraction procedure, a quantization function is applied to each of the selected coefficients to extract the watermark values. For authentication, if it fails, then a tamper assessment is employed to determine the credibility of the modified content. The tamper assessment function is as follow:

$$TAF(w, \tilde{w}) = \frac{1}{N_w} \sum_{i=1}^{N_w} w(i) \oplus \tilde{w}(i) \tag{2.9}$$

where w is the true watermark, $\tilde{w}$ is the extracted mark, N_w is the length of the watermark, and $\oplus$ is the exclusive-OR operator. The value of $TAF(w, \tilde{w})$ ranges between zero and one. The presence of tampering is determined if $TAF(w, \tilde{w}) \geq \tau$, where$0 \leq \tau \leq 1$

is prespecified threshold. If $TAF(w, \tilde{w}) < \tau$, then the modifications on the image are considered to be incidental and negligible. To determine image modifications for specific frequencies or spatial regions the watermark can be extracted from the corresponding marked wavelet coefficients alone.

In Lu and Liao's approach[43], robust and fragile watermarks are simultaneously embedded, for copyright protection and content authentication. By quantizing a host image's wavelet coefficients as masking threshold units, two complementary watermarks are embedded using cocktail watermarking and they can be blindly extracted without access to the host image. For the purpose of image protection, the new scheme guarantees that, no matter what kind of attack is encountered, at least one watermark can survive well. On the other hand, for the purpose of image authentication, this approach can locate the part of the image that has been tampered with and tolerate some incidental processes that have been executed. In addition to images, this method has been extended to audio watermarking[4]. However, the watermark is embedded into the FFT domain of an audio signal while the image approach embeds the watermark in wavelet domain.

In this thesis, we design and implement a semi-fragile audio watermarking scheme for copyrighting and authentication. In our approach, the watermark is embedded into host audio by quantizing the selected wavelet coefficients. This idea is similar to Kunder's approach [42]. However, there are several differences between Kundr's approach and ours. In our approach, we embed the watermark into several wavelet decomposition levels other than one level. In this way, we make our watermarking scheme a multipurpose one. In quantization step, it is difficult to find a proper quantization parameter because the value of audio wavelet coefficients are very small. We use different quantization parameters in different wavelet levels. This is based on the characteristic of audio wavelet coefficients. In Chapter 4, we will introduce our scheme in detail.

Chapter 3

Techniques used in the scheme

3.1 Wavelet Transform

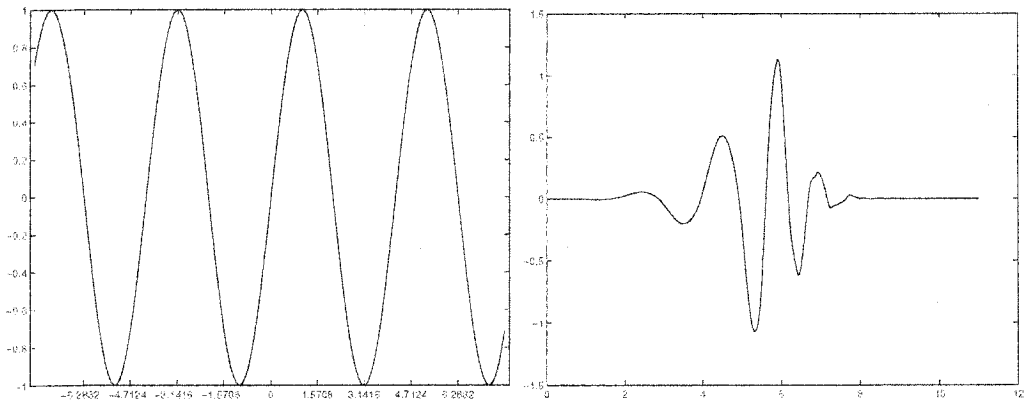
Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that sine and cosine functions could be used to represent other functions.

A wavelet is a waveform of effectively limited duration that has an average value of zero. Compare wavelet with sine waves, which are the basis of Fourier analysis. Sinusoids do not have limited duration, and while sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric. Fig.3.1 shows an example of a sinusoid and a wavelet.

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is to break up a signal into shifted and scaled versions of the original wavelet (or mother wavelet).

The scale in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a global view of the signal, and low scales

correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scale) correspond to a global information of a signal, whereas high frequencies (low scale) correspond to a detailed information.



(a) Sine wave (b) Wavelet(db3)

Figure 3.1: Difference between sine wave and wavelet.

In the next two sections, we will introduce discrete wavelet transform and inverse discrete wavelet transform.

3.1.1 Discrete Wavelet Transform (DWT)

For many signals, the low-frequency content is the most important part. It contains the main features of a signal. The high frequency, on the other hand, imparts flavor of nuance. Consider the human voice. If we remove the high-frequency components, the voice sounds different, but we can still tell what has being said. However, if we remove enough low-frequency components, we hear gibberish.

In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components of a signal, while the details are the low-scale, high-frequency components.

In discrete wavelet transform, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

In order to define the functions of low pass filters and high pass filters, we can start from the scaling function $\phi(t)$:

$$\phi(t) = \sum_{k=0}^{n} \sqrt{2} h_k \phi(2t-k) = \sum_{k=0}^{n} C_k \phi(2t-k) \tag{3.1}$$

From the scaling coefficients h, we can define four FIR (Finite Impulse Response) filters, organized as follows.

Table 3.1: Four FIR filters

Filters	Low Pass	High Pass
Decomposition	h	g
Reconstruction	$\tilde{h}$	$\tilde{g}$

The four filters are computed using the following scheme shown in Fig. 3.2. In this scheme, we first get the low pass filter h from the scaling function (refer to Equ.3.1). Filter h and g known as the low pass filter and the high pass filter are FIR quadrature-mirror filters. For an even length low pass filter h, the two are related by the following formula:

$$g_n = (-1)^n h_{N+1-n}, n = 0, 1, \ldots, N-1 \tag{3.2}$$

where N is the length of filter h. In Fig.3.2, the function qmf indicates this relationship. The function $reverse$ is used to reverse a vector. For example if $x = [x_1, x_2, x_3]$, then $reverse(x) = [x_3, x_2, x_1]$. Therefore, the filters for reconstruction are reversions of the decomposition filters.

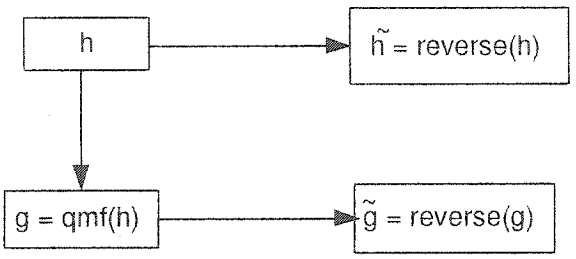


Figure 3.2: The scheme for computing four filters.

After getting the four FIR filters, we can start the discrete wavelet transform. The procedure begins with passing the signal through a half band digital low pass filter and a half band high pass filter. The original signal then emerges as two signals: the low-frequency part A and the high-frequency part D. Unfortunately, if we actually perform this operation on a real digital signal, we end up with twice as much data as we started with. In order to keep the total number of samples to be same as the original signal, we keep only one point out of two in each of the two sub-signals. This is called downsampling. After downsampling, we produce two sequences: cA and cD. Fig.3.3 indicates the whole procedure of wavelet decomposition.

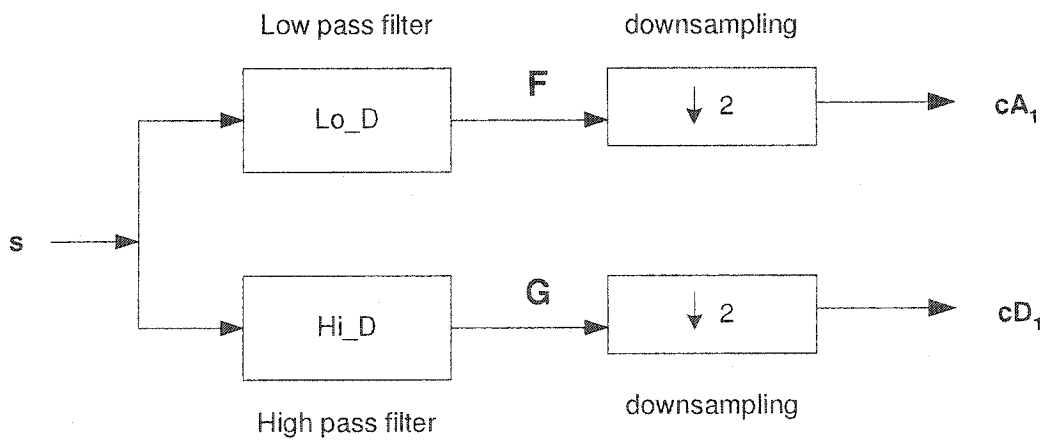


Figure 3.3: The basic step of discrete wavelet transform.

The wavelet decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called the wavelet decomposition tree. Fig.3.4 illustrates the structure of the wavelet decomposition tree.

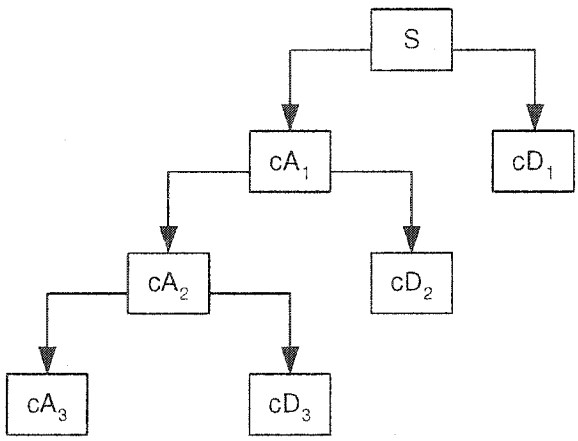


Figure 3.4: Wavelet decomposition tree.

Since the decomposition process is iterative, in theory it can be continued indefinitely. In reality, the decomposition can proceed only until the last approximations consist of a single sample. That is, given a signal s of length N, the DWT consists of $\log_2 N$ levels.

The mathematical representation for calculating cAs and cDs in different resolutions are as follows:

$$cA_k{}^{j+1} = \sum_n h_{n-2k} cA_n{}^j \tag{3.3}$$

$$cD_k{}^{j+1} = \sum_n g_{n-2k} cD_n{}^j \tag{3.4}$$

where j denotes the resolution and k is the index for the samples. The coefficients $cA_k{}^j$ and $cD_k{}^j$ are known respectively as the level j approximation and detail coefficients. The top-level coefficients cA^J represent the original signal.

3.1.2 Inverse Discrete Wavelet Transform (IDWT)

The inverse discrete wavelet transform is to assemble the wavelet coefficients back into the original signal. This is called reconstruction.

While wavelet analysis involves filtering and downsampling, the wavelet reconstruction process consists of upsampling and filtering. Upsampling is the process of lengthening a signal component by inserting zeros between samples. The low pass and high pass filters, $\tilde{h}$ and $\tilde{g}$ respectively, can be calculated by using the algorithm shown in Fig.3.2.

The reconstruction is processed by first upsampling the coefficients and then passing the approximations and details through low pass filter $\tilde{h}$ and high pass filter $\tilde{g}$ separately. Fig.3.5 shows the whole procedure.

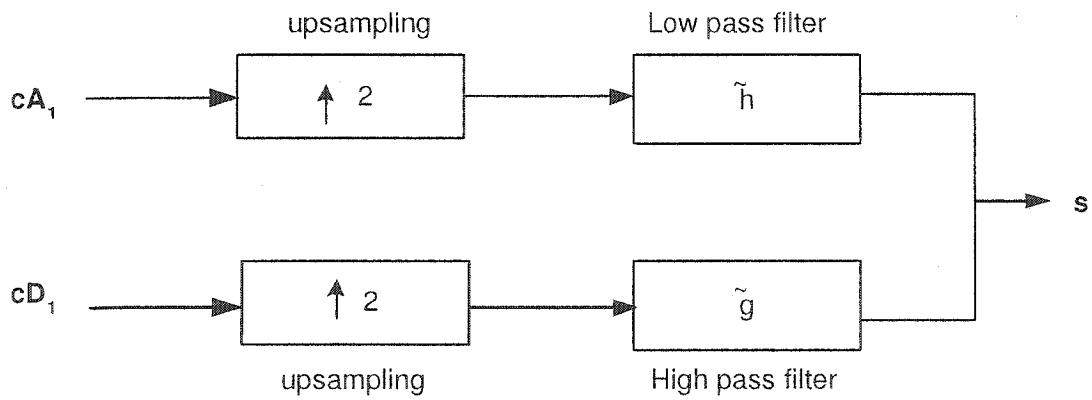


Figure 3.5: Reconstruction of approximations and details.

In multilevel reconstruction, the iteration is employed as we do in the decomposition procedure. We start the wavelet synthesis from the lowest level of a wavelet decomposition tree. The lowest approximation and detail coefficients are first upsampled, filtered and then add together to get their predecessor approximations. The process continues with the calculated approximations and the corresponding details which both

are in the same level. It will stop until we get the original signal. The mathematical representation each iteration is as follow:

$$c_k{}^{j-1} = \sum_n (\tilde{h}_{k-2n} \cdot cA_n{}^j + \tilde{g}_{k-2n} \cdot cD_n{}^j) \tag{3.5}$$

where j denotes the resolution and k is the vector index. The top-level sequence c^J represents the original signal. Take Fig. 3.4 as example, coefficients cA_3 and cD_3 will first get together to produce cA_2. Then we use the computed result cA_2 and cD_2 to calculate cA_1, and then with cD_1, we can get the final result: the original signal S.

3.2 Quantization Technique

The definition of quantization is the division of a quantity into a discrete number of small parts, often assumed to be integral multiples of a common quantity. The input to a quantizer is the original data, and the output is always one among a finite number of levels. The quantizer is a function whose set of output values are discrete, and usually finite.

There are two types of quantization - scalar quantization and vector quantization. In scalar quantization, each input symbol is treated separately in producing the output, while in vector quantization the input symbols are clubbed together in groups called vectors, and processed to give the output.

We can define a quantizer in scalar quantization as consisting of a set of intervals or cells $S = \{S_i; i \in \Gamma\}$, where the index set Γ is ordinarily a collection of consecutive integers beginning with 0 or 1, together with a set of reproduction values or points $C = \{y_i; i \in \Gamma\}$, so that the overall quantizer q is defined by $q(x) = y_i$ for $x \in S_i$, which

can be expressed concisely as

$$q(x) = \sum_i y_i 1S_i(x) \tag{3.6}$$

where the indicator funtion $1s(x)$ is 1 if $x \in S$ and 0 otherwise [45]. For this definition to make sense, we assume that S is a partition of the real line. That is, the cells are disjoint and exhaustive.

A quantizer can be specified by its input partitions and output levels. If the input range is divided into levels of equal spacing, then the quantizer is termed as a uniform quantizer, and if not, it is termed as a non-uniform quantizer. Fig. 3.6 shows an example of uniform quantizer.

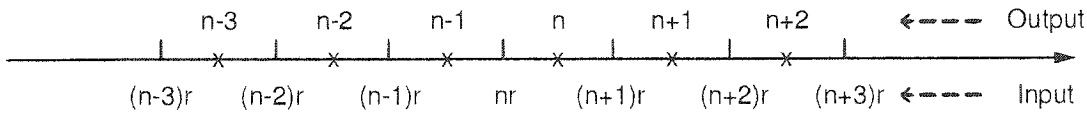


Figure 3.6: A uniform quantizer.

In this figure, the corresponding quantizer is: $q(x) = n$, if $nr \leq x < (n+1) \cdot r$; where n is a real number. This quantizer divides real numbers with a uniform real number r.

In our approach, we employ a uniform scalar quantizer. The uniform real number is called quantization parameter.

3.3 Matching Filters

For template h and the watermarked audio g, where h is shorter than g, the one dimensional normalized cross-correlation function measures the similarity for each translation:

$$r(u) = \frac{\sum_t g(t-u)h(t)}{\sqrt{\sum_t (g(t-u))^2}} \tag{3.7}$$

If the template h matches the audio g exactly, at a translation i, the cross-correlation will have its peak at $r(i)$. The major disadvantage of the cross-correlation method is time-consuming.

According to correlation theorem, the Fourier transform of the correlation of the two signals is the product of the Fourier transform of the correlation of one signal and the complex conjugate of Fourier transform of the other. The Fast Fourier Transform based methods are fast and efficient.

$$R = F^{-1}[G(\omega) \cdot H^*(\omega)] \tag{3.8}$$

where

$$G(\omega) = F[g(t)] = A_G(\omega)e^{-j\Phi_G(\omega)} \tag{3.9}$$

$$H(\omega) = F[h(t)] = A_H(\omega)e^{-j\Phi_H(\omega)} \tag{3.10}$$

and $*$ is the complex conjugate. In Equ.3.8, 3.9 and 3.10, F and F^{-1} are Fourier transform and inverse Fourier transform, respectively.

The following defines three types of traditional filters [46] and a filtering method used by Liu *et al.* (we call it"filtering method 4") [47]:

1. Classical matched filter

   Apply the following filter to $G(\omega)$ in Equ.3.8:

$$H(\omega) = A_H(\omega)e^{-j\Phi_H(\omega)} \tag{3.11}$$

2. Phase-only filter

Apply the following filter to $G(\omega)$ in Equ. 3.8:

$$H_\Phi(\omega) = e^{-j\Phi_H(\omega)} \tag{3.12}$$

3. Amplitude-only filter

   Apply the following filter to $G(\omega)$ in Equ. 3.8:

$$H_A(\omega) = A_H(\omega) \tag{3.13}$$

4. Filtering method 4

   Apply the phase only filter to the phase of the $G(\omega)$ in Equ. 3.8:

$$G(\Phi) = e^{-j\Phi_G(\omega)} \tag{3.14}$$

In image matching, experimental results demonstrated[46] that phase-only filter yields much sharper correlation peaks and better discrimination[48]. It is because that the phase information is considerably more important than the amplitude information in preserving the visual intelligibility of the picture. In case of log-polar mapping, the matching method used by Liu *et al.* is the only one performing well[47], and all others fail when there is a rotation or scaling or both.

However, in our audio case, we found that the classical filter performs much better than others (refer to Section 4.3.4).

Chapter 4

The Proposed Scheme and Implementation Strategies

In this chapter, we propose a semi-fragile audio watermarking scheme. In this scheme, watermark is embedded by quantizing the selected coefficients with a specified quantization parameter Δ in wavelet domain. The watermark's sensitivity against the modification could be controlled by using different values of Δ. This watermarking algorithm is a multipurpose one, which can be used not only for copyright protection but also for content authentication.

We make use of the discrete wavelet domain to embed the watermark because it provides both spatial and the frequency information at the same time. In our scheme, the wavelet coefficients are divided into blocks. One watermark bit is embedded into one block. In this way, the watermark is embedded in all the possible time region of an audio. The localization of the watermark gives the ability to identify distinct regions of watermarked audio which has undergone tampering.

On the other hand, we embed the same watermark string into all of the possible coefficient levels. The watermark in each level is an individual one. This idea is based on

the following point: there is no such an attack which can destroy both the low-frequency and the high-frequency parts at the same time without destroy the perceptual quality of an audio. As we know, the discrete wavelet transform decomposes an audio into one low-frequency level and several high-frequency levels. If we embed watermark into all of these levels, no mater what attack is operated on the audio signal, we can always find out at least one watermark in these levels. The global spread of the watermark makes it robust to all the possible attacks.

In the following part of this chapter, the details of the watermark embedding and the extraction procedure are introduced.

4.1 Watermark Embedding

The flowchart for the whole embedding process is shown in Fig.4.1. The procedure contains the following steps.

Step 1: Watermark Generation The watermark to be embedded is a pseudorandom binary string. It is generated with a pseudorandom noise (PN) code generator. The seed of the generator is used as a secret key of this watermarking system.

Step 2: Discrete Wavelet Transform We choose a mother wavelet function and then compute the Lth-level discrete wavelet decomposition of the original audio. After the decomposition we obtain $L + 1$ sets of coefficients: $\{cA_L, cD_L, cD_{L-1}, \ldots, cD_1\}$, where cA_L is the low-frequency component and $\{cD_L, cD_{L-1}, \ldots, cD_1\}$ are high-frequency components. The value of L is defined by the user.

Step 3: Blocking In each level of wavelet decomposition, the coefficients are first

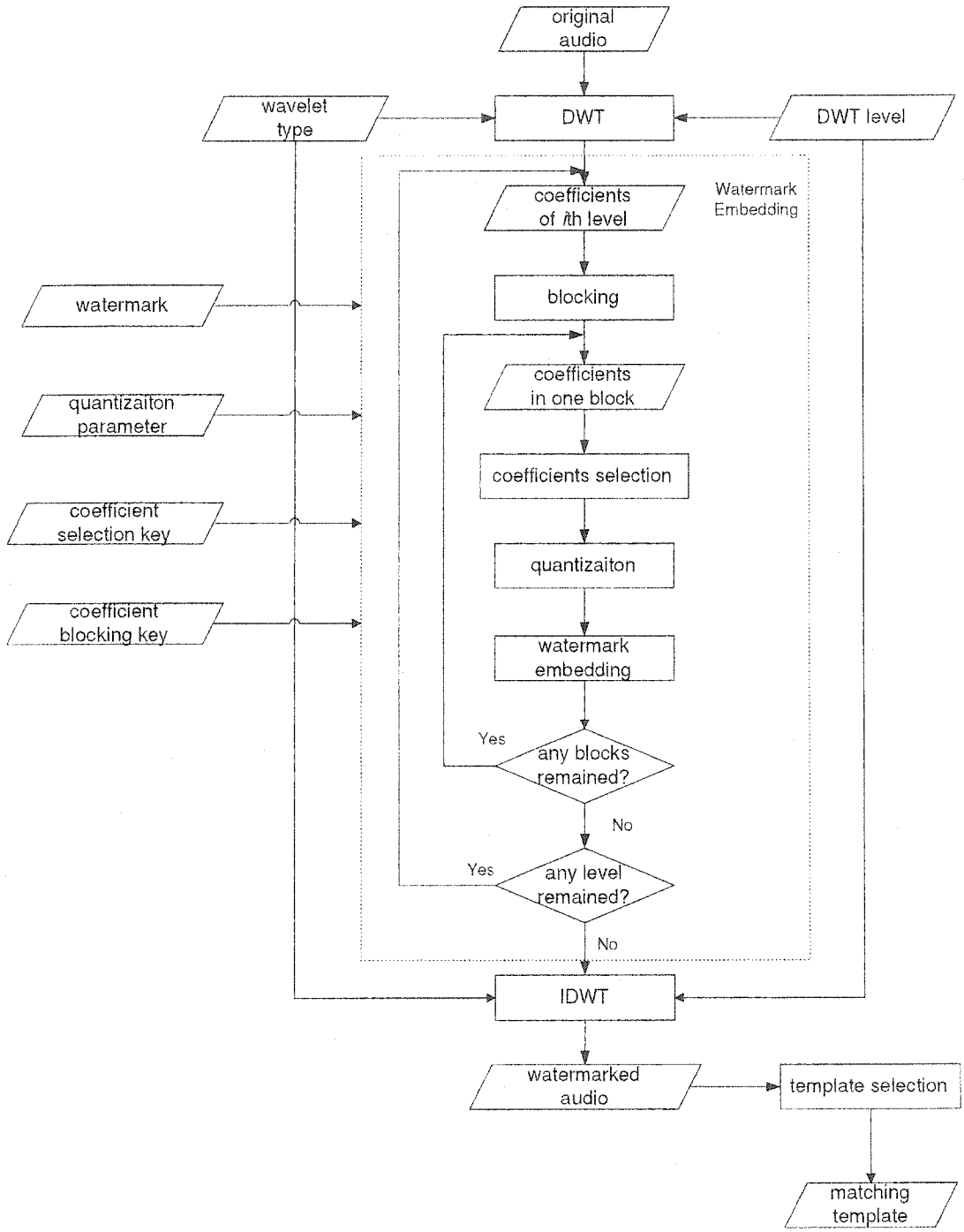


Figure 4.1: Flowchart of watermark embedding procedure.

divided into blocks, each of which is used to embed one watermark bit. The starting point of the first block should be a sufficiently large number so that the watermark bit will not be embedded to a silent region in the beginning part of audio signal. In our approach, the watermark could be embedded into all of the levels, it can also be embedded into some selected levels. The number of the blocks in each level is decided by the coefficient blocking key.

Step 4: Coefficients Selection In each block, we choose the first N largest coefficients to embed the same watermark bit, where N is decided by the coefficient selection key.

Step 5: Quantization For any discrete wavelet transform, the coefficients are real numbers. The quantization procedure could be performed on these coefficients. In quantization step, every real number is assigned a binary number 0 or 1. We could quantize an arbitrary coefficient with the following assignment [42]:

$$Q(e) = \begin{cases} 0 \quad if \quad k \times \Delta \leq e < (k+1) \times \Delta \\ \qquad\quad (k = 0, \pm 2, \pm 4 \ldots) \\ 1 \quad if \quad k \times \Delta \leq e < (k+1) \times \Delta \\ \qquad\quad (k = \pm 1, \pm 3, \pm 5 \ldots) \end{cases} \tag{4.1}$$

where e is the value of the coefficient, while Δ is a positive real number called quantization parameter. Fig.4.2 illustrates this assignment.

Step 6: Watermark Embedding After quantizing the selected coefficients, the watermark is embedded by using the following rules. In these rules, $w(i)$ is the watermark bit to be embedded, $Q(e)$ is the quantization value of the selected coefficient e.

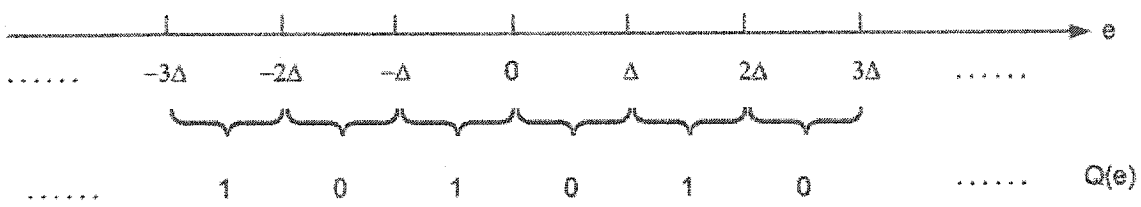


Figure 4.2: The quantization function.

- If $Q(e) = w(i)$, no change will be made to the coefficient.
- If $Q(e) \neq w(i)$, the coefficient e will be forcibly changed so that $Q(e) = w(i)$, using the function $e = e + \Delta$. Δ is the same quantization parameter as in Equ.4.1.

An example of embedding the watermark at one sample point is shown in Fig. 4.3. In this example, m is one of the coefficients selected for embedding the watermark. The watermark bit to be embedded is binary 0. After computing the quantization value of m, we found that $Q(m) \neq w(i)$. Then the second rule is used, which is $m = m + \Delta$. Finally, the new quantization value of m becomes 0, which is equal to the value of the embedded watermark bit. This rule ensures that the quantization value of the selected coefficient is same as the embedded watermark bit.

The parameter Δ is user defined. A smaller value of Δ makes a minor change to coefficients in embedding step and hence makes the watermark sensitive to audio modifications. On the other hand, a large value of Δ makes the watermark robust to many manipulations, but it will achieve a bad imperceptibility. As a semi-fragile watermark, how to decide the value of parameter Δ is a problem. We will discuss it in the following section.

Step7: Iteration Repeat from step 3 to step 6 until we embed the watermark into all

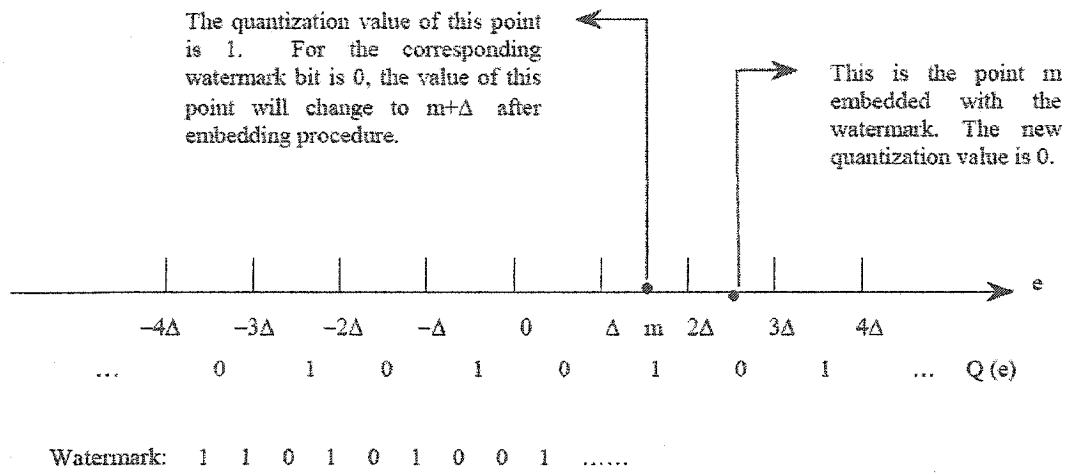


Figure 4.3: Effect of embedding watermark on one sample point.

of the selected coefficients in all of the selected decomposition levels.

Step8: Inverse Discrete Wavelet Transform In the last step of embedding, we assemble the wavelet coefficients to a watermarked audio by using the corresponding Lth-length inverse discrete wavelet transform.

Step9: Matching Filter Before we deliver the watermarked audio into the communication network, we will use it to make a matching template. This matching template is a small part of the watermarked audio, which is stored for watermark extraction.

4.2 Watermark Extraction

Fig. 4.4 is the flowchart of the watermark extraction. In this procedure, the secrete keys of the system are required. The secrete keys include coefficient blocking key, coefficient selection key and the quantization parameter.

Step1: Start Point Matching Due to some intentional or non-intentional audio pro-

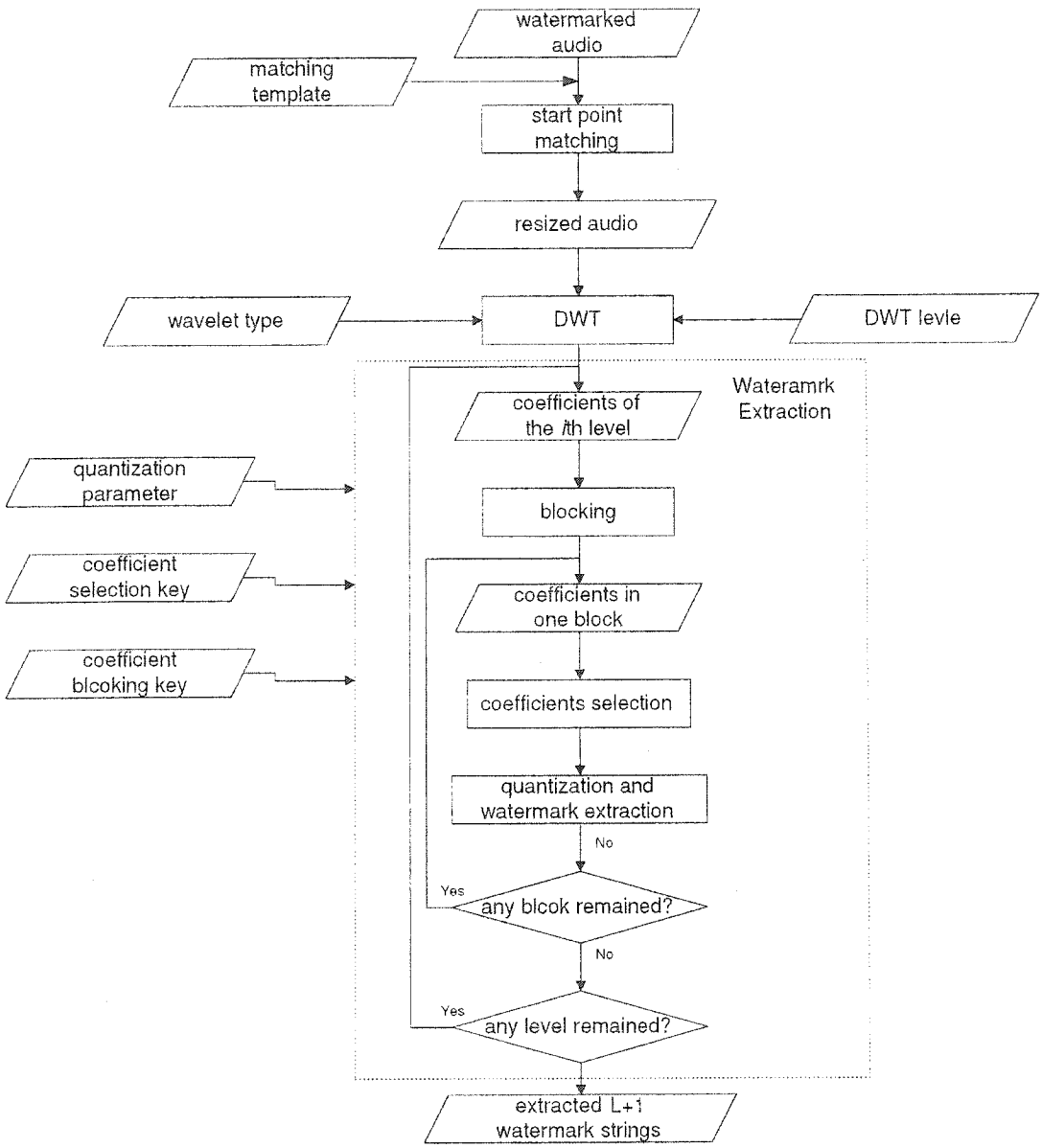


Figure 4.4: Flowchart of watermark extraction procedure.

cessing such as MP3 compression and decompression, the watermarked audio may be shifted and the total number of the sample points may be changed. Therefore, the start point of the audio must be located before watermark extraction. We employ a matched filter to efficiently and fast fulfill this task. The template for the filter is obtained from the watermark embedding process.

Step2: Discrete Wavelet Transform Once the start point is located, the Lth-length discrete wavelet transform is applied. The mother wavelet should be the same one as what is used in the embedding procedure.

Step3: Blocking In every level which has embedded with the watermark, blocks are divided. In this step, the coefficient blocking key is used to decide the number of blocks.

Step4: Coefficient Selection The first N largest coefficients in each block are selected for extraction. The number of N is decided by the coefficient selection key.

Step5: Watermark Extraction The quantization function $Q(e)$ in Equ. 4.1 is applied to these selected coefficients. In each block, these N coefficients represent the same watermark bit. Therefore the quantization values of these N coefficients should be the same one in the case that there is no attack to the watermarked audio. However, if there is any manipulation applied to the audio signal, the quantization value of these 50 binary numbers might be different. Thus, we introduce the following function to decide whether the watermark bit is 1 or 0 in one block.

$$W'(i) = \begin{cases} 1 & if \quad N_1 > N_0 + C \\ 0 & if \quad N_0 > N_1 + C \\ 2 & o.w. \end{cases} \tag{4.2}$$

In this function, N_1 is the number of binary ones in N quantization values of one block, N_0 is the number of binary zeros, C is a constant. The decision will be 1 if the binary ones are more than zeros over the threshold C. And the decision will be 0 if zeros are more than ones over C. In order to minimize false detection, we introduce state 2, which represents a grey area where a decision cannot be made.

step6: Iteration Repeat from step 3 to step 5 until we extract the watermark from all of the selected levels.

The result of the extraction is K watermark strings $w_1, w_2, \ldots, w_k$, where K is the number of selected levels. Each string corresponds to one decomposition level. For the purpose of copyright verification, we can check whether there is a watermark in the audio by using the original watermark string.

The following equation is first used to calculate the similarities between the original watermark and the extracted watermark strings:

$$P_i = \frac{\sum_{j=1}^{N_i} w(j) \oplus w_i^*(j)}{N_i} \tag{4.3}$$

where w is the original watermark, w^* is the extracted watermark. N is the length of the watermark. i means the ith level of the coefficients. $\oplus$ is exclusive-OR operator. The similarity is in the range of [0,1].

After the computation, we get K similarities $P_1, P_2, \ldots, P_K$. A threshold T is set for the decision of whether the extracted watermark is similar to the original one. If

$P_i > T$, we can conclude that there is a watermark in level i; otherwise, this is no watermark. The copyright verification will succeed if there is at least one similarity P_i larger than the threshold. In other words, we can say that the audio is watermarked if we find a watermark in any wavelet decomposition level.

Another application of this approach is content authentication. The authentication succeeds only if the extracted K watermarks are all identical to the original one. It means that it must satisfy the following function:

$$\frac{1}{K}\sum_{i=1}^{K} P_i = 1 \tag{4.4}$$

Almost any manipulations of audio will cause the authentication to fail. If the authentication fails, an additional program will be employed to determine where the audio content is modified.

In this additional program, we choose a similarity which is not equal to 1 from $P_K, P_{K-1}, \ldots, P_1$. The following equation is used to figure out the start and the end point of the modified region:

$$\begin{aligned} start &= (sp + length \times (m-1) + 1) \times 2^i \\ end &= (sp + length \times n) \times 2^i \end{aligned} \tag{4.5}$$

where *start* and *end* are the first and last samples of the modified region. i denotes the decomposition level. *sp* is the start point of the first block in ith level. *length* is the length of each block. m and n mean that from the mth extracted watermark bit to the nth watermark bit are incorrect. We can often find out several values which are not equal to 1 in the similarities. Any one of them could be used to place the modified region.

The modified region figured out by the additional program is just an approximation. The accuracy depends on the length of blocks in each level. The shorter the blocks, the more accurate the result.

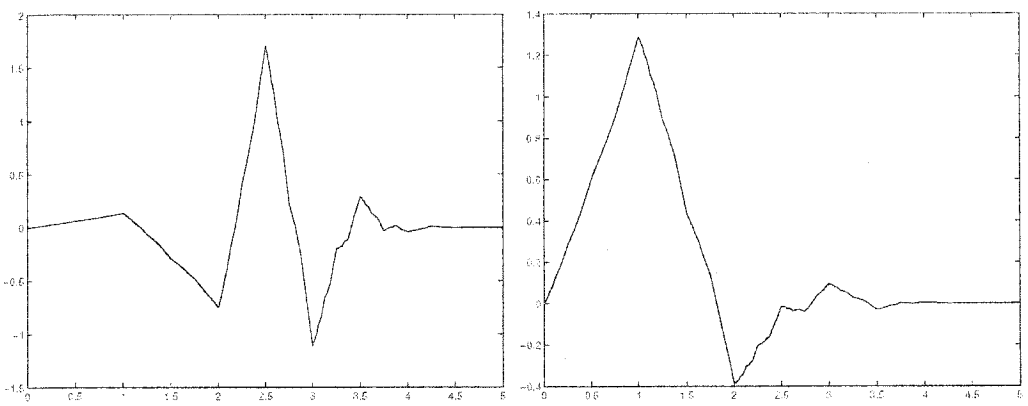
4.3 Implementation Strategies

In this section, we list out several important aspects for implementing our scheme.

4.3.1 Mother Wavelet and Decomposition Level

In the implementation, we use Daubechies-3 wavelet with 10 levels to derive DWT coefficients.

The wavelet function and scaling function of Daubechies-3 wavelet is shown in Fig. 4.5.



(a) db3 wavelet function (b) db3 scaling function

Figure 4.5: The db3 wavelet.

The four FIR filters for Daubechies-3 are as follows. And they are illustrated in Fig. 4.6.

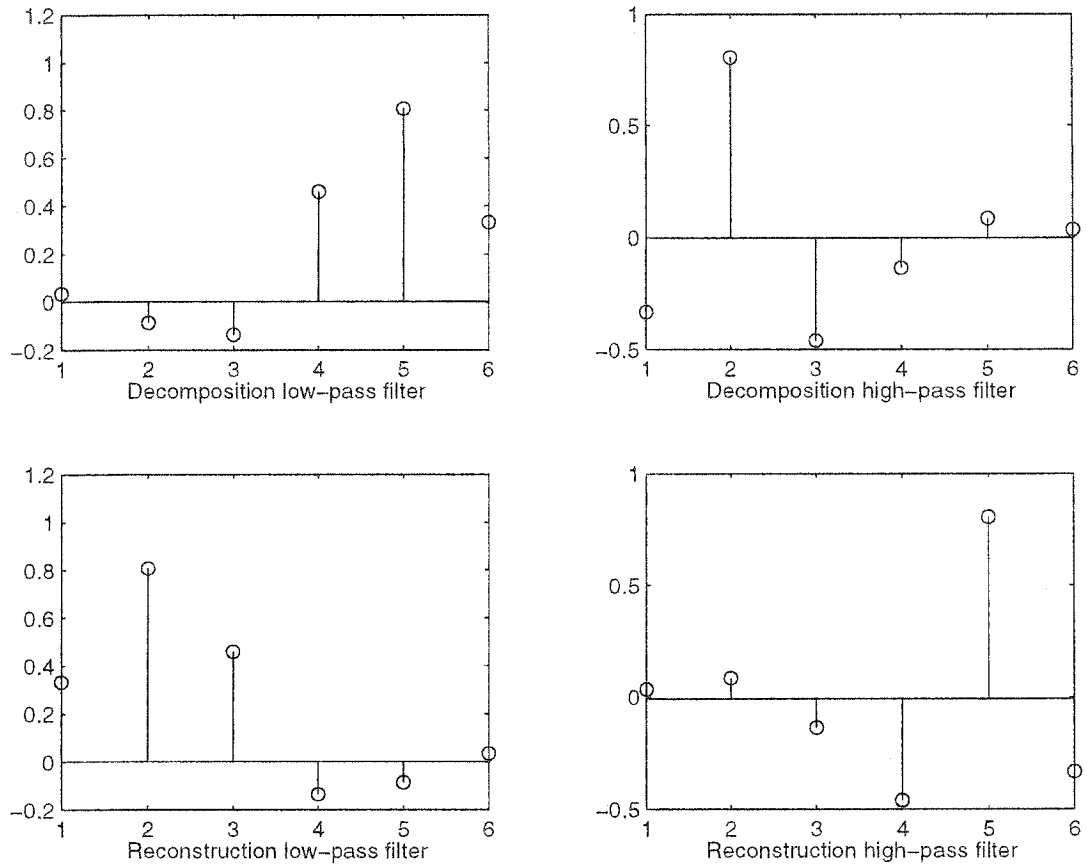


Figure 4.6: The four filters for db3.

$$Lo_D = [0.035, -0.085, -0.135, 0.459, 0.807, 0.333]$$
$$Hi_D = [-0.333, 0.807, -0.459, -0.135, 0.085, 0.035]$$
$$Lo_R = [0.333, 0.807, 0.459, -0.135, -0.085, 0.035]$$
$$Hi_R = [0.035, 0.085, -0.135, -0.459, 0.807, -0.333]$$

We use a 10-level discrete wavelet transform. The testing audio signals we use are all sampled at 44.1kHz. According to the Nyquist Theorem which states that the highest frequency which can be accurately represented is less than one-half of the sampling

rate, the maximum frequency exists in these testing audio signals are 22.05kHz. As we mentioned in Section 3.1.1, the signal is decomposed by passing through half-band low pass filters and high pass filters. We can calculate the frequency boundaries in different decomposition levels. Table 4.1 shows the results.

Table 4.1: Frequency bound in different DWT level

coefficients	corresponding frequency (Hz)
cD1	11025~22050
cD2	5512~11025
cD3	2756~5512
cD4	1378~2756
cD5	689~1378
cD6	344~689
cD7	172~344
cD8	86~172
cD9	43~86
cD10	21~43
cA10	0~21

4.3.2 Secrete Keys

The coefficient selection key N in our approach is 50. The constant C in Equ. 4.2 is defined based on the experiment results, which will be shown in Section 5.1.2. The coefficient blocking key is 64.

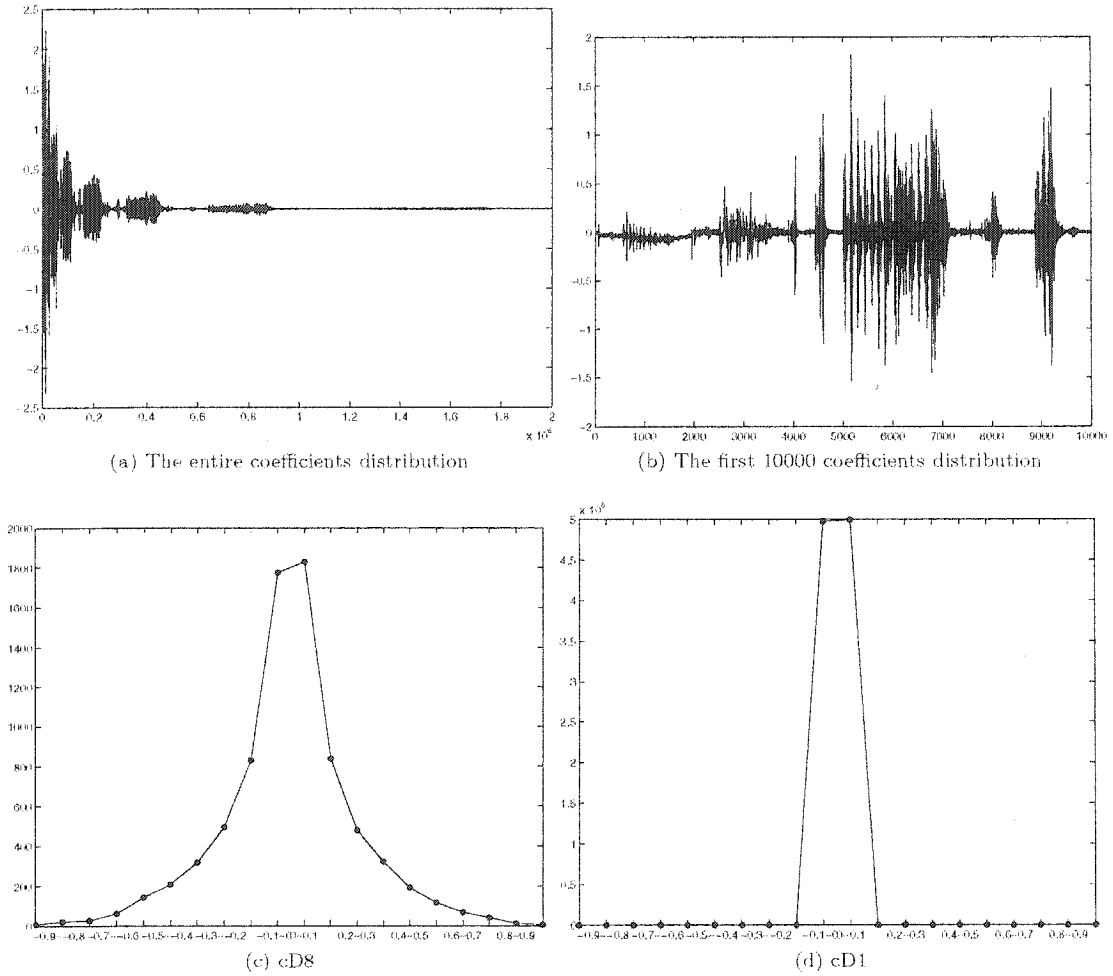
As we mentioned in the previous section, our approach could embed watermark in all of the DWT levels, or the watermark could be embedded in some selected levels. In our implementation, we choose the latter. The first reason for this choice is that any changes in low-frequency part may affect the audio perceptibility much. The second reason is the number of coefficients in low-frequency components. The testing audio signals we use all have 1994752 samples. As we know, the signal will be downsampled by two after one step of wavelet decomposition. With a 10-level discrete wavelet transform,

the lowest level $cD10$ and $cA10$ has only 1948 samples ($1994752/2^{10} = 1948$), while the highest level $cD1$ contains 997376 samples ($1994752/2^1$). As the coefficient selection key is 50, we should guarantee that there are at least 100 coefficients in one block. In this case, we can only embed a watermark with the maximum length of 19, which can not meet the requirement of the coefficient blocking key. Therefore, we decide to embed the watermark only in the highest 8 levels of coefficients. These levels are from $cD8$ down to $cD1$. In these 8 levels, the coefficients are all divided into 64 blocks. Thus, the watermark in each level has 64-bit length.

4.3.3 Quantization Parameter Δ

As we mentioned in the previous section, there is a tradeoff between robustness and imperceptibility of a watermark with different value of the quantization parameter Δ. A large parameter makes the watermark robust, but it will deteriorate the original quality of an audio. On the other hand, a small parameter allows us to achieve the goal of authentication though its immunity of modification is poor.

In our approach, we use different quantization parameters in different DWT levels. This means, if we want to embed a watermark in L levels, we have to select L parameters which correspond to these L levels. This idea is based on the characteristic of DWT coefficients. In wavelet domain, the absolute values of coefficients in medium frequency level are larger than those in low frequency and high frequency levels. Fig. 4.7 demonstrates this characteristic. Fig. 4.7(a) shows the entire wavelet coefficients distribution of testing audio 02. Fig. 4.7(b) only focuses on the first 10000 coefficients of this testing audio. In these two figures, the frequency grows up from the left to the right. Fig. 4.7(c) and (d) shows the coefficients distribution in $cD8$ and $cD1$ of the test audio. In the highest frequency $cD1$, the coefficients centralize in the field of [-0.1,



(a) The entire coefficients distribution

(b) The first 10000 coefficients distribution

(c) cD8

(d) cD1

Figure 4.7: The wavelet coefficients distribution of testing audio 02.

0.1]; while in the medium frequency $cD8$, the coefficients spread across the area from -0.9 to 0.9. The choice of the quantization parameter depends on the value of the coefficients. The larger the coefficients, the larger the parameter we could use. Therefore, the quantization parameter used in the medium frequency could be larger than that in the low and high frequency. Another reason for using multiple parameters is that the low frequency part contains most of the information of an audio. A slight modification of this part will cause some audible noise. The high frequency part contains the details

of the audio signal. It can be easily removed by many attacks. Thus, it is good to use large parameters in medium frequencies and small parameters in high frequencies.

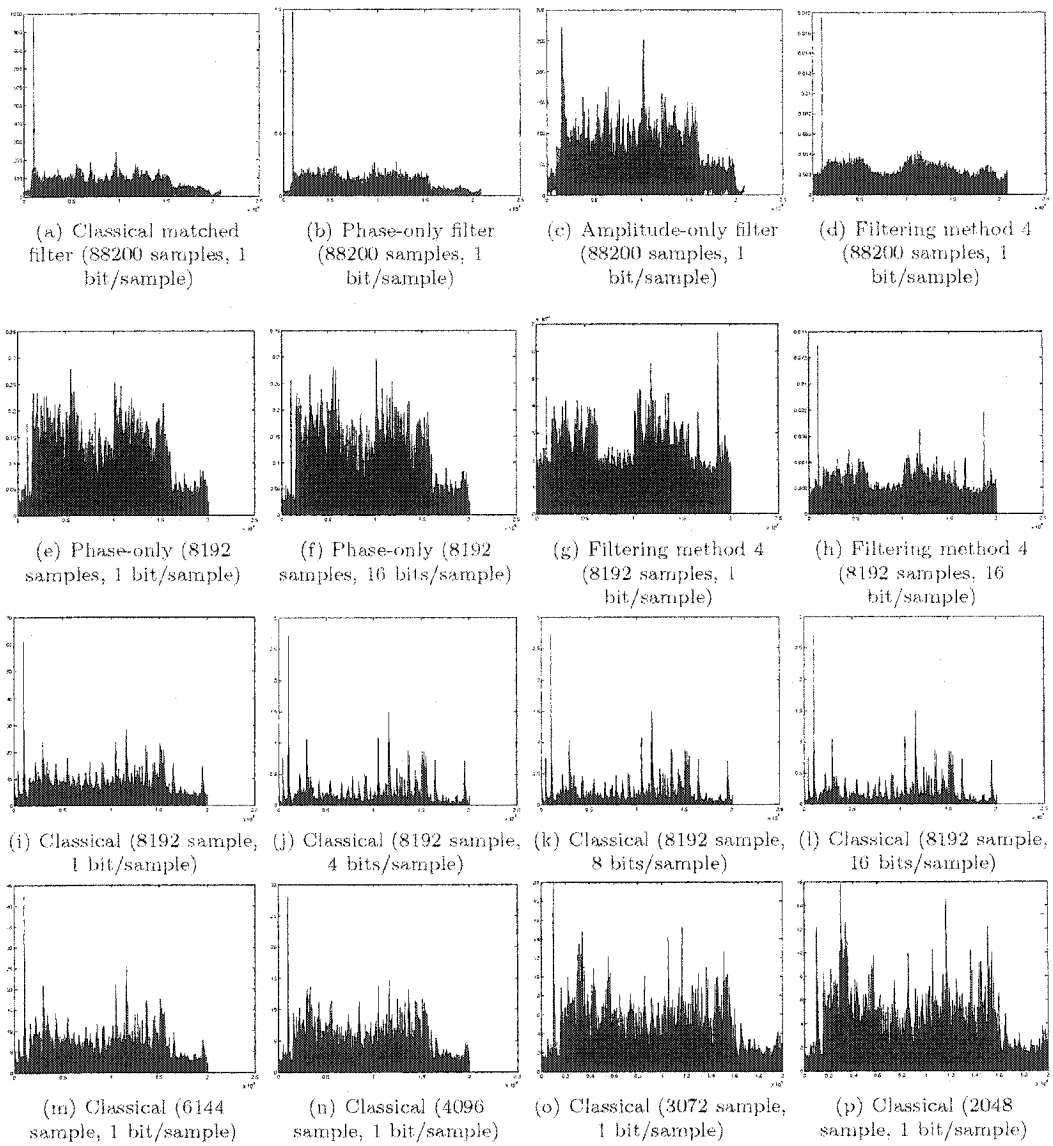
In our approach, we will find 8 quantization parameters for the decomposition levels from $cD8$ to $cD1$. The selection of quantization parameters is based on the experiments. In Section 5.1.1, we will list out the extraction results of using single quantization parameter and those of using multiple parameters. The results will prove that the multiple one is a better choice.

4.3.4 Matching Filter

In Section 3.3, we list four matching filters, which are classical matched filter, phase-only filter, amplitude-only filter and filtering method 4. In image matching, experiments show that phase-only filter produce much better results than other filter methods. However, in our audio case, we did several experiments and the final result is that the classical filter is the most robust and performs the best (refer to Fig.4.7).

Fig. 4.7(c) shows that the amplitude-only filter still has unacceptable discrimination. Both phase-only filter and the filtering method 4 perform well when the matching template is long enough, e.g., 88200 samples (refer to Fig. 4.7 (b) and (d)); and fail when the length of the template becomes small (refer to Fig. 4.7 (e) to (h)). The number of bits per sample does not affect the performance of the phase-only filter much, but does affect the performance of the filtering method 4.

According to our experiments, the length of the template affects matching performance more than the number of bits per sample. For the classical filter, the number of bits per sample does not affect its performance in a great deal, refer to Fig. 4.7 (i) to (l). The filter can match the template even when the template length is 4096. When the length of the template is under 3072, the classical filter begins to fail.



(a) Classical matched filter (88200 samples, 1 bit/sample)

(b) Phase-only filter (88200 samples, 1 bit/sample)

(c) Amplitude-only filter (88200 samples, 1 bit/sample)

(d) Filtering method 4 (88200 samples, 1 bit/sample)

(e) Phase-only (8192 samples, 1 bit/sample)

(f) Phase-only (8192 samples, 16 bits/sample)

(g) Filtering method 4 (8192 samples, 1 bit/sample)

(h) Filtering method 4 (8192 samples, 16 bit/sample)

(i) Classical (8192 sample, 1 bit/sample)

(j) Classical (8192 sample, 4 bits/sample)

(k) Classical (8192 sample, 8 bits/sample)

(l) Classical (8192 sample, 16 bits/sample)

(m) Classical (6144 sample, 1 bit/sample)

(n) Classical (4096 sample, 1 bit/sample)

(o) Classical (3072 sample, 1 bit/sample)

(p) Classical (2048 sample, 1 bit/sample)

Figure 4.8: The matching results of different filters with different parameters. The template is cut from the watermarked test audio, and the audio to match, 45 seconds long with sample rate of 44.1K and 16 bits/sample, is the watermarked audio undergone MP3 compression and decompression.

The classical filter performs much better than all other three, even when length of the matching template is 4096 samples with 1 bit/sample. Therefore we choose to use the classical filter. The length of the template we use is 8192, and the number of bits per sample is 1. The cost of the matching template is 1024 bytes.

Chapter 5

Experimental Results and Evaluations

In this chapter, we first discuss how to set some parameters by experiments. Then we evaluate the performance of the proposed scheme against noising, filtering, mp3 attacks and its performance on authentication.

5.1 Experiments on Choosing Parameters

In the proposed scheme, there are two parameters which are based on experiments. They are quantization parameters Δ and the constant C. In this section, we conduct several experiments to decide the value of these parameters.

5.1.1 Quantization Parameter Δ

In Section 4.3.3, we discussed about single quantization parameter and multiple quantization parameters. Table 5.1 and Table 5.2 show the extraction results of using these two types of parameters in a testing audio. In the experiments, we perform two attacks

on each watermarked audio. One is low-pass filtering with 9000Hz and the other is mp3 compression with bit rate of 128kbps. The results in the table are similarities of the original watermark and the extracted watermark, which are calculated from Equ. 4.3.

Table 5.1: Results with single quantization parameter

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
low-pass filter	0.9844	0.8281	0.4531	0.4063	0.3906	0.4688	0.5781	0.5625
mp3 compression	0.3594	0.3594	0.3281	0.5938	0.9688	1	1	0.5156

Table 5.2: Results with multiple quantization parameters

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
low-pass filter	1	1	0.9375	0.8906	0.4688	0.5	0.2969	0.3438
mp3 compression	0.9375	0.9688	0.9531	1	0.9688	1	1	0.5781

The eight columns in these two tables correspond to the highest eight levels, cD1 represents the highest frequency coefficients, and cD8 represents is the lowest frequency level among these eight levels. The numbers in the table denotes the similarity between the original watermark and the extracted watermark.

The results demonstrate that our watermarking approach will be more robust using multiple quantization parameters than using single one. Therefore, we implement our approach with multiple ones.

A difficulty of the multiple parameter solution is how to select the proper set of the parameters. As we know, the modifications of coefficients in one wavelet decomposition level may affect the coefficients in other levels if we perform inverse DWT followed by DWT. It means that the change of the quantization parameter for one level may effect other levels' performance. Therefore, for each testing audio, we conduct a number of experiments with different sets of quantization parameters. We then choose a better

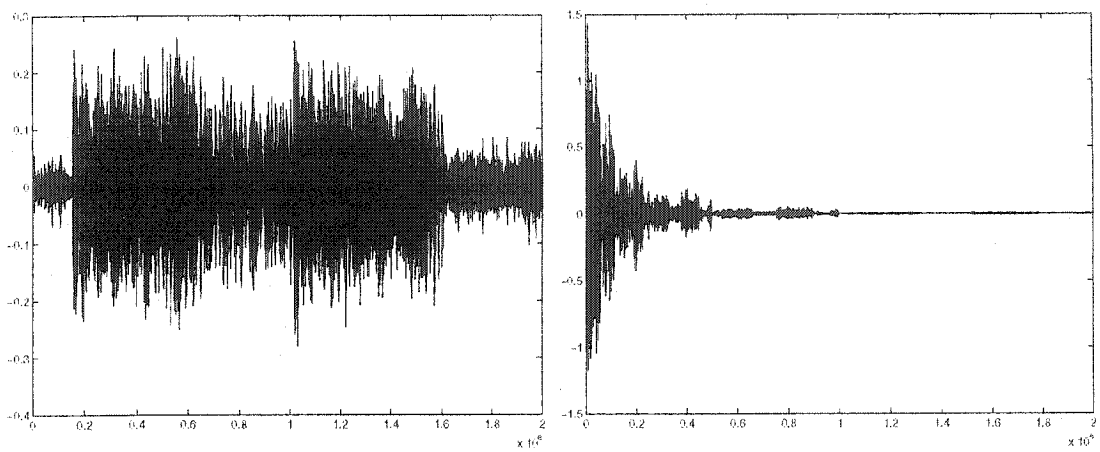
one according to the results. This procedure may cost a lot of time and it can only be performed manually.

Another difficulty of quantizing audio coefficients is that different audio clips may have different strength in the spectrum. This fact makes the parameter set dependent on the audio signals. Fig. 5.1 shows two testing audios in time domain and in their corresponding wavelet domain. In Fig. 5.1(a) and (c), x-axis is the sample points and y-axis is the amplitude; in (b) and (d), x-axis is the coefficient point and y-axis is the values of the coefficients.

Fig. 5.1(a) is time domain of testing audio 01, and Fig. 5.1(c) is time domain of testing audio 06. It is clear that audio 06 is stronger than audio 01. Fig. 5.1(b) and (d) are wavelet coefficients of testing audio 01 and audio 06, respectively. The coefficients of audio 06 value from -15 to 15; while coefficients of audio 01 are from -1.5 to 1.5. The figure demonstrates that if an audio signal is stronger in the time domain than the other one, the coefficients of the stronger signal in wavelet domain will be larger than coefficients of the weaker one. In this situation, a single quantization parameter set cannot satisfy these two signals at the same time. Therefore, we have to choose different quantization parameters with different audio signals.

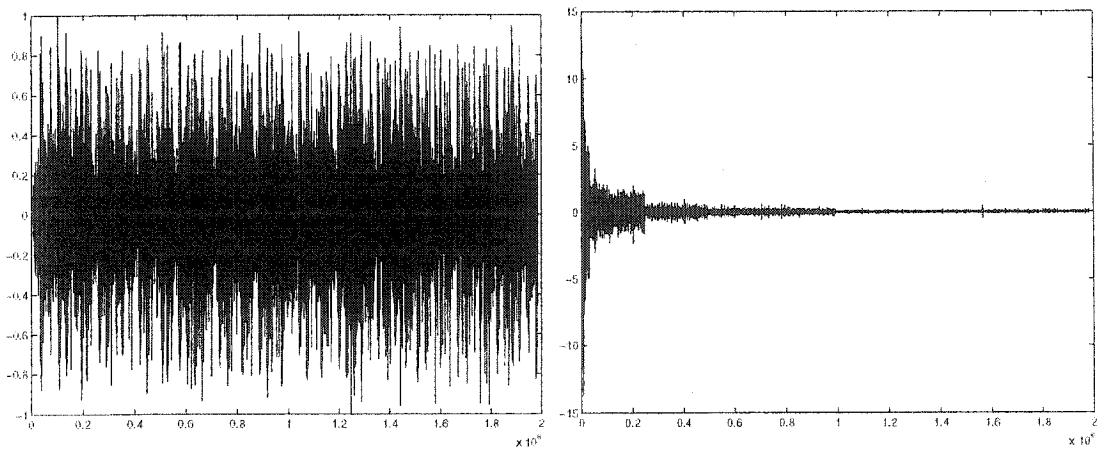
5.1.2 Extraction Constant C

In Equ. 4.2, the constant C is a threshold that is used for deciding whether the extracted binary bit is 0 or 1. A small value of C may lead to a high detection value for unwatermarked audio, which is defined as the percentage of correct bits that is detected from non-watermarked content. On the other hand, a large C will cause a high bit error rate, which is the percentage of erroneous bits in an extracted message out of all bits. Therefore, it is important to give C a proper value.



(a) Testing audio 01 in time domain.

(b) Testing audio 01 in wavelet domain.



(c) Testing audio 06 in time domain.

(d) Testing audio 06 in wavelet domain.

Figure 5.1: The difference between testing audios in time and wavelet domain.

In the implementation of our scheme, we use 50 coefficients to represent one watermark bit. It means that in the extraction procedure: $state1 + state0 + state2 = 50$, where $state1$ means that the embedded watermark bit in these 50 coefficients is binary 1, $state0$ means that the watermark bit is binary 0, and $state2$ is a gray area which means it cannot make a decision. Therefore, the range of C should be [1,49].

We perform several experiments on two testing audios with different values of constant C. The experiments are of two types: one is to extract the watermark from the

original audio in order to calculate the detection value for unwatermarked audio; the other one is to extract the watermark from audio compressed with mp3 in order for the bit error rate. We then decide a value for C according to the results. In this experiment, the detection value for unwatermarked audio is computed by using Equ. 4.3. The bit error rate is computed by :

$$BER_i = 1 - \frac{\sum_{j=1}^{N_i} w(j) \oplus w_i^*(j)}{N_i} \tag{5.1}$$

where w is the original watermark, w^* is the extracted watermark. N is the length of the watermark. i means the ith level of the coefficients. $\oplus$ is exclusive-OR operator.

Table 5.3, Table 5.4, Table 5.5 and Table 5.6 show the results. The numbers in the tables represent percentage. cDs are coefficients in different DWT levels. The rows of the tables correspond to different values of C. We choose the experimental range of C from 12 down to 6.

The results prove our conclusions that bit error rate declines with decreasing C and detection value for unwarermarked audio grows up with increasing C.

In our approach, we set the threshold of detection value for unwatermarked audio as 0.3. It means that the detection value of the watermark in each level should be under 0.3. The results show that neither of the two signals could reach the threshold if C is smaller than 8. Therefore, we choose $C = 8$.

5.2 Experimental Results and Evaluation

We perform the watermark embedding and extraction on six testing audio signals, which are sampled at 44.1kHz with 16 bits per sample in mono. The testing audio are divided in two categories: the first three signals are classical music and the last three are pop

Table 5.3: False positive rate (Audio-01)

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
C = 12	0.0469	0.0781	0.0156	0.125	0.1406	0.1719	0.1094	0.1094
C = 11	0.0469	0.0938	0.0469	0.1563	0.1563	0.1875	0.1406	0.125
C = 10	0.0469	0.0938	0.0469	0.1563	0.1563	0.1875	0.1406	0.125
C = 9	0.0781	0.1094	0.0938	0.2188	0.2344	0.2031	0.2188	0.1563
C = 8	0.0781	0.1094	0.0938	0.2188	0.2344	0.2031	0.2188	0.1563
C = 7	0.1719	0.1719	0.1406	0.3125	0.2969	0.2188	0.2813	0.2344
C = 6	0.1719	0.1719	0.1406	0.3125	0.2969	0.2188	0.2813	0.2344

Table 5.4: Bit error rate with 64kbps MP3 compression (Audio-01)

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
C = 12	0.6406	0.625	0.6563	0.3906	0.5781	0.3906	0.8438	0.8438
C = 11	0.5938	0.5938	0.6094	0.3281	0.5469	0.3281	0.7656	0.7969
C = 10	0.5938	0.5938	0.6094	0.3281	0.5469	0.3281	0.7656	0.7969
C = 9	0.5313	0.5313	0.5313	0.2344	0.4063	0.2813	0.7188	0.7344
C = 8	0.5313	0.5313	0.5313	0.2344	0.4063	0.2813	0.7188	0.7344
C = 7	0.4375	0.5	0.4688	0.1719	0.3438	0.2031	0.6719	0.7031
C = 6	0.4375	0.5	0.4688	0.1719	0.3438	0.2031	0.6719	0.7031

Table 5.5: False positive rate (Audio-04)

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
C = 12	0.2031	0.0469	0.0469	0.1406	0.1563	0.1094	0.2344	0.0469
C = 11	0.2031	0.0469	0.0781	0.2031	0.2188	0.2031	0.2656	0.0469
C = 10	0.2031	0.0469	0.0781	0.2031	0.2188	0.2031	0.2656	0.0469
C = 9	0.2188	0.0781	0.125	0.2188	0.2813	0.25	0.2813	0.0938
C = 8	0.2188	0.0781	0.125	0.2188	0.2813	0.25	0.2813	0.0938
C = 7	0.25	0.1875	0.1719	0.2188	0.3281	0.2656	0.2969	0.2031
C = 6	0.25	0.1875	0.1719	0.2188	0.3281	0.2656	0.2969	0.2031

Table 5.6: Bit error rate with 64kbps MP3 compression (Audio-04)

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
C = 12	0.2188	0.2188	0.4531	0.8906	0.8438	0.9063	0.875	0.9063
C = 11	0.1406	0.1563	0.3594	0.8281	0.7969	0.8594	0.8125	0.8906
C = 10	0.1406	0.1563	0.3594	0.8281	0.7969	0.8594	0.8125	0.8906
C = 9	0.0625	0.0938	0.2969	0.7969	0.7344	0.8125	0.7969	0.8594
C = 8	0.0625	0.0938	0.2969	0.7969	0.7344	0.8125	0.7969	0.8594
C = 7	0.0469	0.0974	0.2188	0.5938	0.6094	0.75	0.7031	0.8281
C = 6	0.0469	0.0974	0.2188	0.5938	0.6094	0.75	0.7031	0.8281

music. All of these audio signals are stored in WAV format.

We embed the watermark in the highest 8 levels of the DWT. In each level the watermark is 64-bit long. Therefore, we embed 512 bits into the audio.

In the following section, we perform several experiments on imperceptibility of the watermark, its robustness against some attacks such as additive noise, MP3 compression and filtering. The final part of this section is about the experiments on content authentication.

5.2.1 Perceptual Quality

Perceptual quality refers to the imperceptibility of embedded watermark data within the host signal. As we know, the imperceptibility is one of the most important property of most watermarking algorithms. In our approach, the peak signal-to-noise ratio (PSNR) of the watermarked signal versus the original signal is used as a quality measure:

$$PSNR(f,g) = 10log_{10}\frac{(max_{\forall i}(f(i)))^2}{\frac{1}{N}\sum_{\forall i}(f(i)-g(i))^2} \tag{5.2}$$

where f and g are respectively the original audio and the watermarked audio, and N is the length of the audio.

Table 5.7 shows the peak signal-to-noise ratio for the 6 testing audio signals. From the results, we can conclude that our watermarking scheme does not bring much distortions to the original signals.

Table 5.7: PSNR of watermarked audio signals

audio name	PSNR
audio-01	58.529
audio-02	55.94
audio-03	56.307
audio-04	49.558
audio-05	52.903
audio-06	45.632

5.2.2 Copyright Verification

In the application of copyright verification, we perform three types of attacks: additive noise, low-pass filtering, and MP3 compression. Table 5.8 to Table 5.15 show the results.

The 8 columns in these tables correspond to the higest 8 levels of DWT, which are level $cD8$, $cD7$, $cD6$, $cD5$, $cD4$, $cD3$, $cD2$ and $cD1$. In these 8 levels, $cD1$ represents the highest frequency coefficients, and the represented frequencies in each level fall down from $cD1$ to $cD8$. The represented frequencies boundary for each level could be found in Tab. 4.1. The rows of the tables are different testing audio signals. The number in the tables denotes the similarity between the original watermark and the extracted watermark. We use the Equ. 4.3 to calculate the similarities.

As we mentioned in Section 4.2, there is a threshold T which is used to decide whether there is a watermark in one DWT level. In the experiment, we set $T = 0.6$. It means that if the similarity P is larger than 0.6, we can conclude that there is a watermark in the corresponding DWT level. Otherwise, there is no watermark.

Table 5.8: Low-pass filtering attack with frequency of 9000Hz

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	1	1	0.9375	0.8906	0.4844	0.4688	0.1719	0.2188
audio-02	1	1	1	0.8281	0.4531	0.4219	0.1563	0.0938
audio-03	1	1	1	0.75	0.7188	0.4531	0.3438	0.3125
audio-04	1	1	1	0.8125	0.6563	0.1875	0.0781	0.1719
audio-05	1	1	1	0.4844	0.75	0.5781	0.2031	0.25
audio-06	1	1	1	0.9219	0.6563	0.1094	0.2031	0.1094

Table 5.9: Additive noise attack with strength of 100

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	1	1	1	1	1	1	0.2188	0.1719
audio-02	1	1	1	1	1	1	0.3281	0.1875
audio-03	1	1	1	1	1	1	0.25	0.1875
audio-04	1	1	1	1	1	1	1	1
audio-05	1	1	1	1	1	1	0.2813	0.2031
audio-06	1	1	1	1	1	1	1	1

Table 5.10: Additive noise attack with strength of 500

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	1	1	1	1	0.8281	0.6719	0.2188	0.2188
audio-02	1	1	1	1	0.8231	0.3594	0.25	0.0938
audio-03	1	1	1	0.7969	0.9219	0.2344	0.1563	0.1406
audio-04	1	1	1	1	1	1	1	0.125
audio-05	1	1	1	0.8125	0.8281	0.2813	0.2344	0.1406
audio-06	1	1	1	1	1	1	1	0.1406

Table 5.11: Additive noise attack with strength of 900

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	0.8906	0.9219	0.9063	0.9375	0.2656	0.3438	0.2188	0.0313
audio-02	0.9844	1	1	0.9219	0.2969	0.1875	0.2656	0.125
audio-03	1	1	0.9844	0.1875	0.3281	0.1875	0.125	0.07813
audio-04	1	1	1	1	1	0.8906	0.9531	0.1719
audio-05	1	1	1	0.1406	0.2969	0.3125	0.1563	0.125
audio-06	1	1	1	1	1	0.9531	0.9063	0.0938

Table 5.12: MP3 compression attack with bit rate 320Kbps

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	1	0.9688	0.9688	1	1	1	0.9375	0.2344
audio-02	1	1	1	1	0.9844	1	0.7188	0.1719
audio-03	1	1	1	1	1	0.9687	0.9531	0.2031
audio-04	1	1	1	0.8906	0.9844	0.7969	0.9375	0.0781
audio-05	1	1	1	0.875	1	1	1	0.0781
audio-06	1	1	1	1	1	0.8438	0.9844	0.2031

Table 5.13: MP3 compression attack with bit rate 128Kbps

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	0.9063	0.9375	0.9219	1	1	1	0.7813	0.2188
audio-02	1	0.9844	1	1	0.9531	0.9688	0.5625	0.2031
audio-03	1	1	1	0.9688	1	0.9063	0.8906	0.2344
audio-04	1	1	1	0.75	0.9063	0.7188	0.8281	0.125
audio-05	1	1	1	0.7344	1	1	0.9675	0.1875
audio-06	1	1	1	0.9844	0.9688	0.6719	0.9219	0.2188

Table 5.14: MP3 compression attack with bit rate 64Kbps

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	0.4688	0.4688	0.4688	0.7656	0.5938	0.7188	0.2813	0.2656
audio-02	0.6094	0.625	0.7031	0.625	0.4375	0.5	0.3906	0.2188
audio-03	0.8594	0.828	0.9219	0.6406	0.7813	0.5781	0.3594	0.25
audio-04	0.9063	0.8906	0.7188	0.2656	0.2656	0.2031	0.2344	0.1406
audio-05	1	1	1	0.2813	0.9531	0.8438	0.4063	0.1563
audio-06	0.875	0.7969	0.8906	0.2344	0.2344	0.1719	0.2031	0.3281

Table 5.15: Extraction from the original audio (unwatermarked)

	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
audio-01	0.07813	0.1094	0.0938	0.2188	0.2344	0.2031	0.2188	0.1563
audio-02	0.2344	0.1875	0.1719	0.2656	0.1719	0.25	0.1875	0.1875
audio-03	0.1406	0.1563	0.1406	0.1563	0.2969	0.2656	0.2656	0.2188
audio-04	0.2188	0.0781	0.125	0.2188	0.2813	0.25	0.2813	0.0938
audio-05	0.0938	0.1094	0.07813	0.0313	0.2969	0.2813	0.2969	0.2031
audio-06	0.0469	0.1094	0.1094	0.14063	0.14063	0.1719	0.2344	0.1875

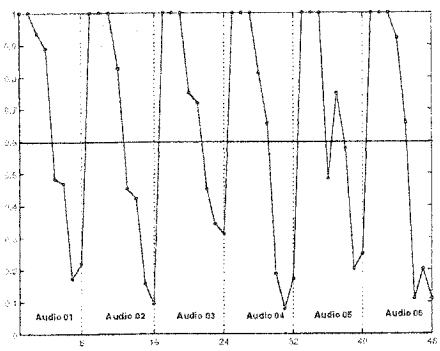
Table 5.8 shows the results of passing the watermarked audio through the 9000Hz low-pass filter. For the reason that we embed watermarks in different frequency levels, after passing the low-pass filter, we can detect the watermarks in the lower frequency levels. Based on this idea, our approach is robust against low-pass filtering even with lower frequency.

Table 5.9, Table 5.10 and Table 5.11 are results of additive noise attack. We use Stirmark for Audio [49] to add noise into watermarked audio. The noise has different strength. Results show that the noise affects high frequency coefficients the most, but it does not have much effect on middle frequency. Therefore, we can detect watermarks in middle frequency.

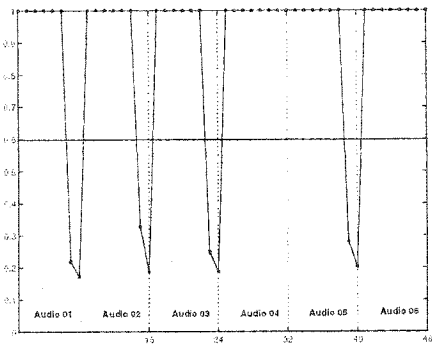
The next three tables (refer to Table 5.12, Table 5.13 and Table 5.14) show the results of MP3 compression. Different bit rate is used in each table. Our approach performs well against MP3 attacks, for we can still find watermarks even after the compression with 64 kbps bit rate.

The last table (refer to Table 5.15) is the results of extracted watermark from original audio signals. Since there is no watermark in the signal, the similarities are very small.

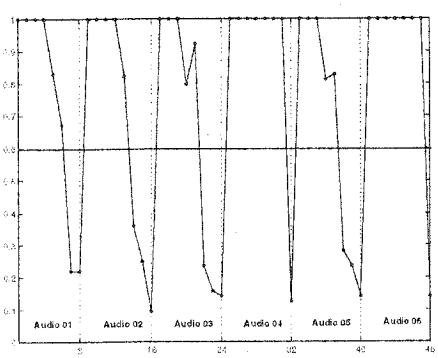
Fig. 5.2 illustrates the results from Table 5.8 to Table 5.15. In these figures, the black points are experimental results which are shown in the above tables. As there are six testing audios, each figure is divided into six blocks by dotted lines. Each block corresponds to one testing audio. In each block, there are 8 black points, which correspond to the results from $cD8$ to $cD1$ from left to right. The values of the black points are similarities between the original watermark and the extracted ones. The horizontal line in each figure represents the threshold. The first seven figures (refer to Fig. 5.2(a) to (g)) shows that for each testing audio there are always black points above the threshold no matter which attack it is undergone. It means that watermarks could always be found out. Fig. 5.2(f) is the results of extracting the watermark from the



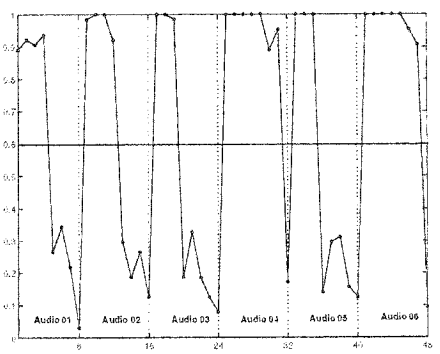
(a)Low-pass filtering attack with frequency of 9000Hz .



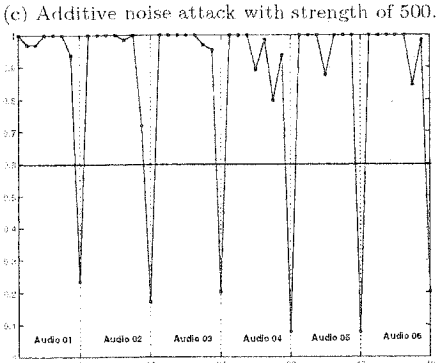
(b) Additive noise attack with strength of 100.



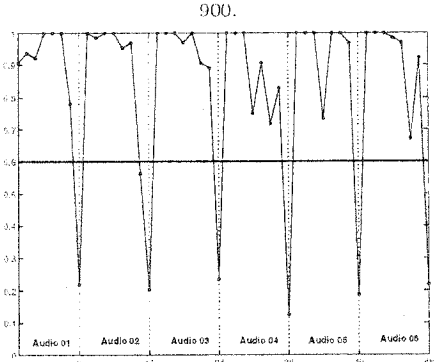
(c) Additive noise attack with strength of 500.



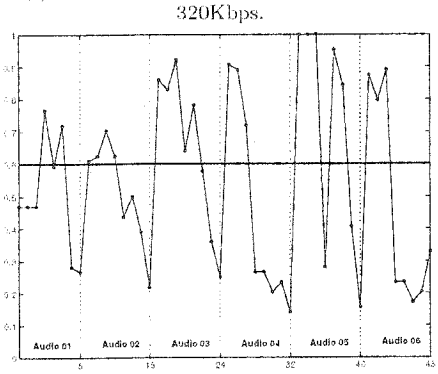
(d) Additive noise attack with strength of 900.



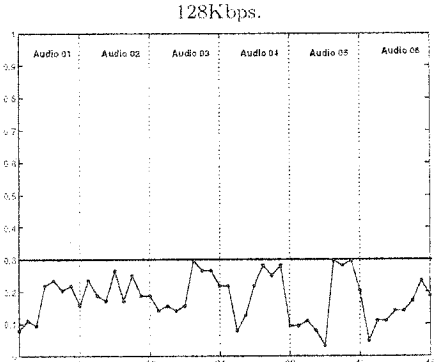
(e) MP3 compression attack with bit rate 320Kbps.



(f) MP3 compression attack with bit rate 128Kbps.



(g) MP3 compression attack with bit rate 64Kbps.



(h) Extraction from the original audio(unwatermarked).

Figure 5.2: The experimental results of six testing audio signals against additive noise, filtering and MP3 compression attacks.

original audio signal. The threshold of this experiment is 0.3. The figure shows that the computed false positive rates are all under this threshold.

5.2.3 Content Authentication

In the application of content authentication, we conduct the experiments of substitution in time domain. We first generate a random string with a random length, choose an arbitrary number as the start point, and then replace the portion of the watermarked audio with the random string from the selected starting point. The similarities of all the 8 levels are computed.

The authentication will succeed only if all of the similarities are equal to 1. Any number that is below 1 indicates that some parts of the audio have been modified. We can determine the modified audio content by comparing the extracted watermark with the original one. After finding the incorrect watermark bits, we can use spatial information of the wavelet coefficients to locate the modified content.

Take testing audio 01 as example, we generate a random string with length of 10000. This random string replaces the watermarked audio from the sample 900000. After performing the extraction procedure to the modified audio, we get the similarities as shown in Table 5.16.

Table 5.16: The results of substituting audio 01 with a random string of length 10000.

DWT level	cD8	cD7	cD6	cD5	cD4	cD3	cD2	cD1
similarity	1	1	1	1	1	0.9844	0.9688	0.9688

The result of Equ. 4.4 in this example is 0.99025, which is not equal to 1. It indicates that this audio clip has been modified and the authentication procedure is failed. At this time, the assistant program will be used to locate the modified part of the audio.

The first step in the assistant program is to compare the extracted watermark in

the level which the similarity is not equal to 1 with the original one. We compare these two watermarks bit by bit and find out which bits are not the same.

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

(a) The original watermark.

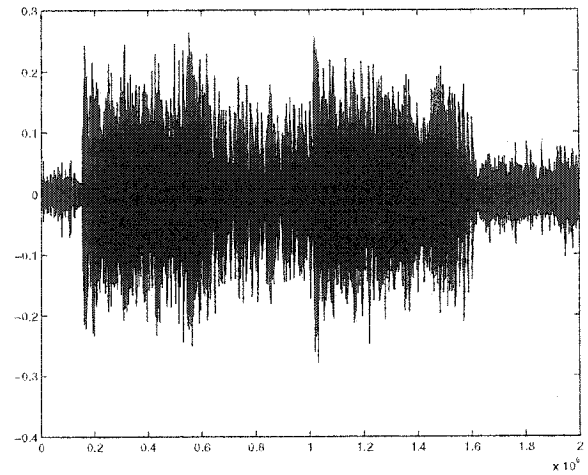
1	1	0	1	1	1	0	0
0	0	1	1	0	0	1	1
1	1	1	1	1	1	0	1
2	2	1	1	0	1	1	1
1	0	0	0	0	0	1	1
1	1	1	0	0	1	0	1
0	0	1	1	0	0	0	0
1	1	0	1	1	0	1	0

(b) The extracted watermark.

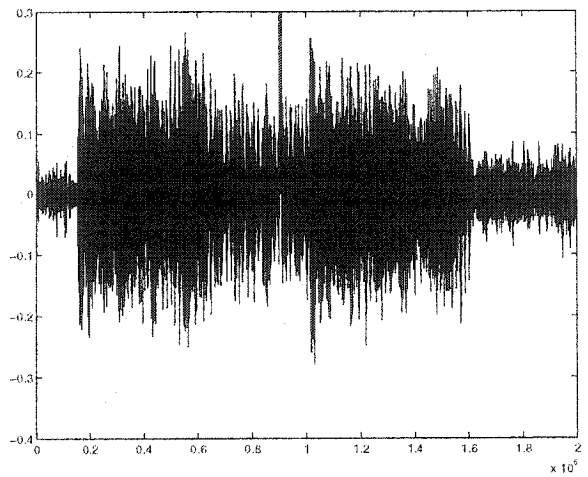
Figure 5.3: The original and the extracted watermarks.

In the example of audio 01, we choose the extracted watermark in level $cD1$. Fig. 5.3 lists out the original watermark and the extracted watermark in $cD1$. The comparison shows that there are two bits which are not the same in these two watermarks. Equ. 4.5 is then used to compute the start point and the end point of the modified part. The result is that the modified part is from the 872448th sample to the 928520th sample. Fig. 5.4(a) and (b) are the watermarked audio and the modified one in time domain. Fig. 5.4 illustrates the modified part of this audio.

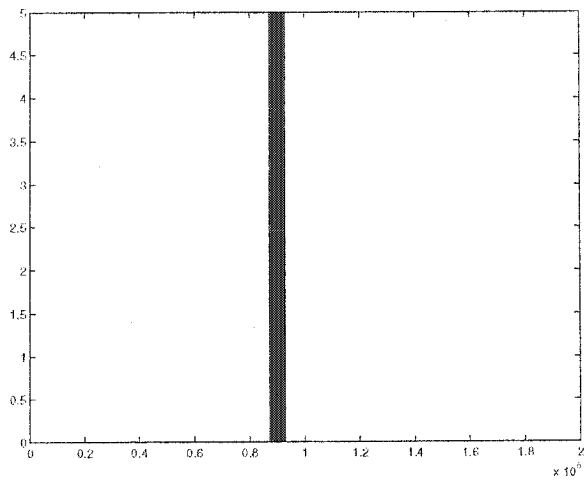
The modified region located by the assistant program is larger than the real one. As in our example, the modified region is from the 900000th sample to the 910000th sample. The region located by the assistant program is from the 872448th sample to



(a) The watermarked audio 01



(b) The modified audio 01



(c) The modified part.

Figure 5.4: The difference between the watermarked audio and the modified one.

the 928520th sample. The accuracy of the assistant program relates to the number of blocks in each DWT level. The more blocks, the more accurate the results.

We perform substitution and extraction procedure on all the other 5 audio signals. The results are shown from Fig. 5.5 to Fig. 5.9.

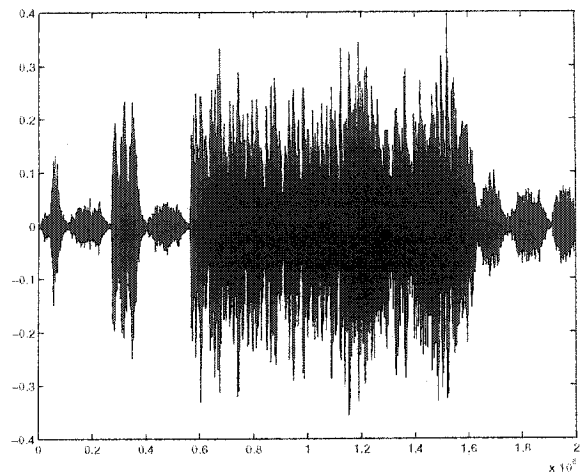
Substitution is one of the simplest modification for audio signals. Other type of modifications may produce more differences between the watermarked audio and the modified one. Therefore, our approach have a good performance in content authentication.

5.3 Summary

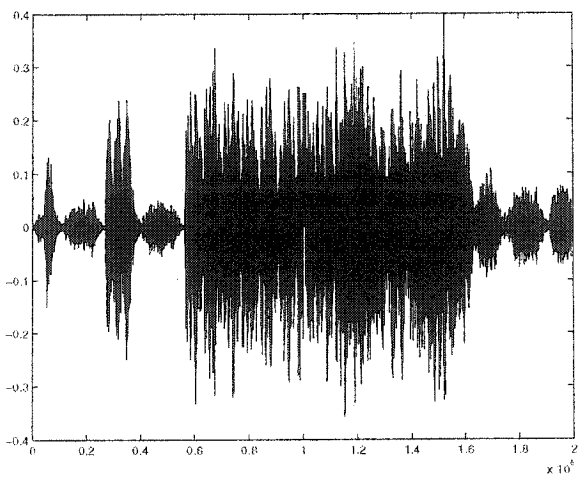
In this chapter, we first presented the experiments to decide the values of the quantization parameter Δ and the extraction constant C. The quantization parameter Δ is dependent on audio signals.

Then we tested our watermarking scheme with different kind of attacks. Our watermarking scheme shows robustness against additive noise, filtering, and MP3 compression attacks.

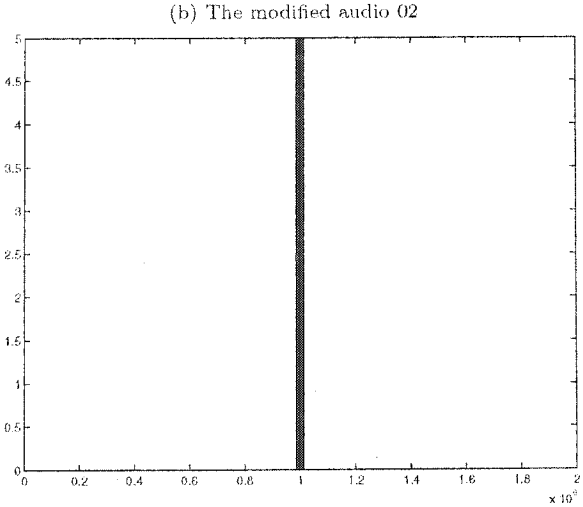
As a multipurpose watermark, our watermarking scheme is efficient in audio authentication. After checking the extracted watermark, we can decide whether the audio signal is modified or not, and with the assistant program, we can locate where have be modified.



(a) The watermarked audio 02

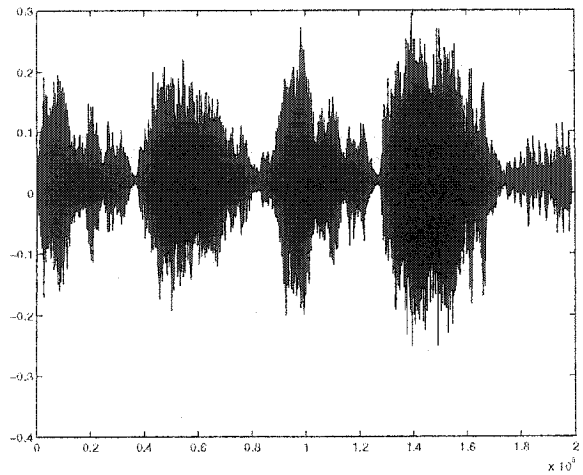


(b) The modified audio 02

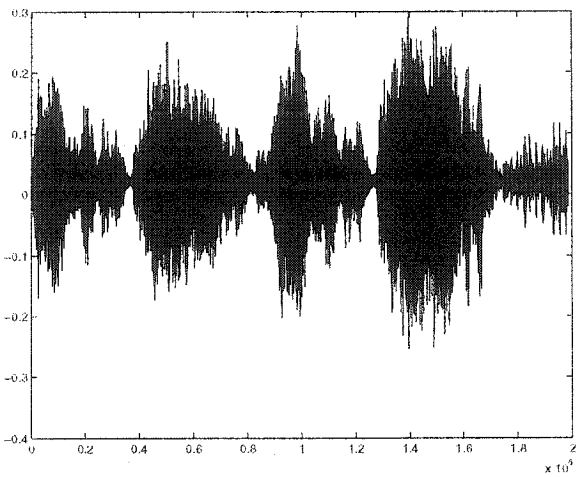


(c) The modified part.

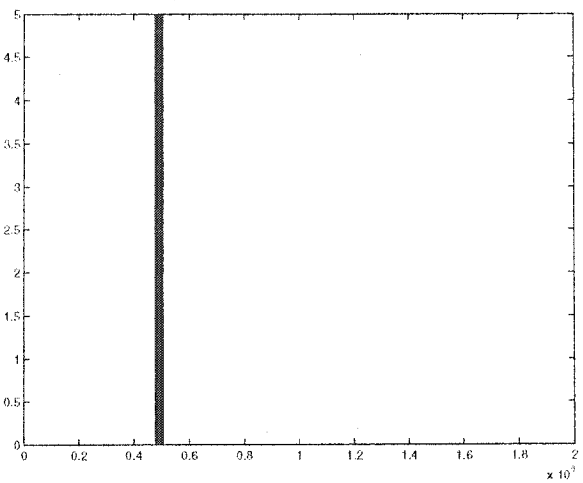
Figure 5.5: The results of substitution in audio 02 with string length of 10000 and start point of 1000000.



(a) The watermarked audio 03

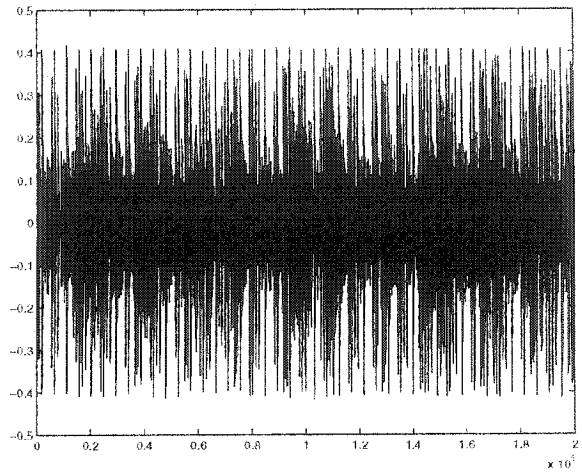


(b) The modified audio 03

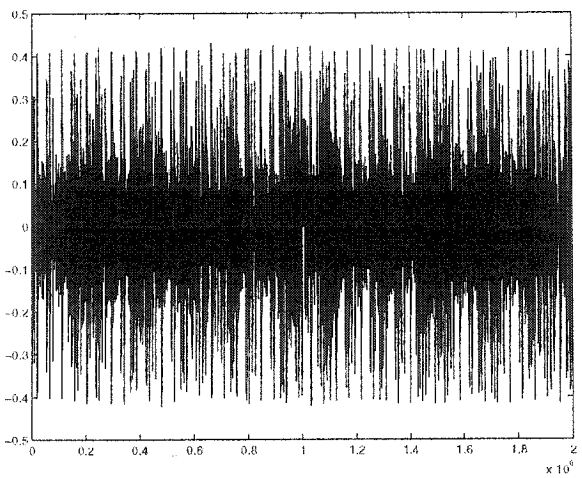


(c) The modified part.

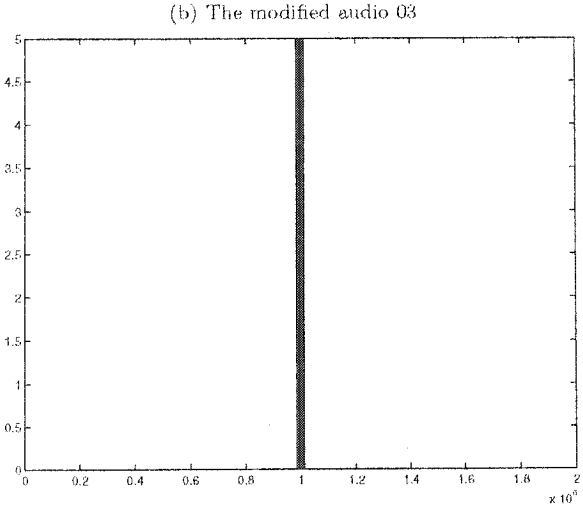
Figure 5.6: The results of substitution in audio 03 with string length of 5000 and start point of 500000.



(a) The watermarked audio 03

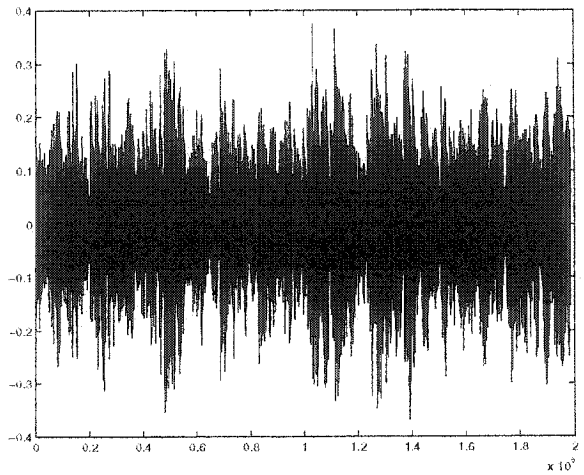


(b) The modified audio 03

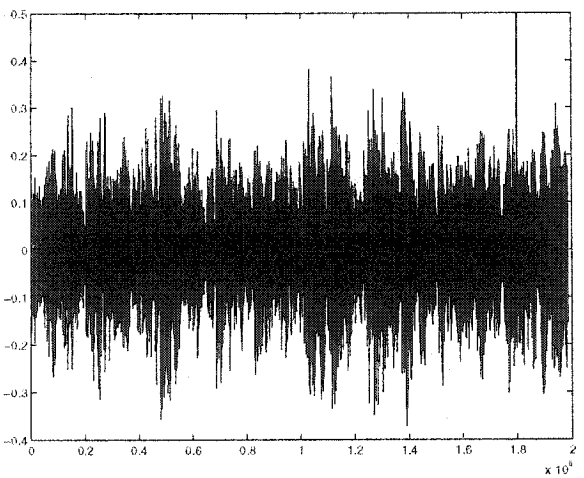


(c) The modified part.

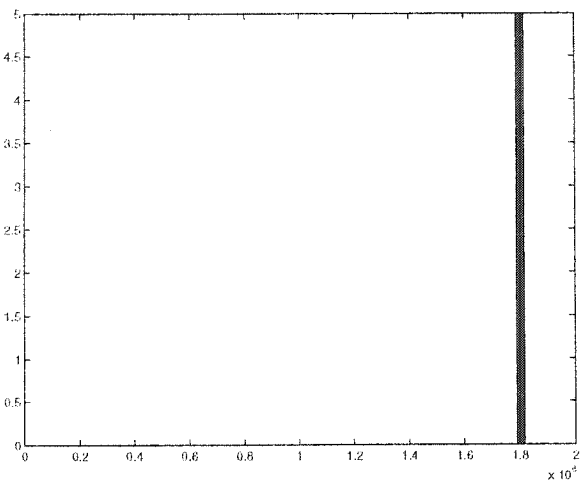
Figure 5.7: The results of substitution in audio 04 with string length of 10000 and start point of 1000000.



(a) The watermarked audio 03

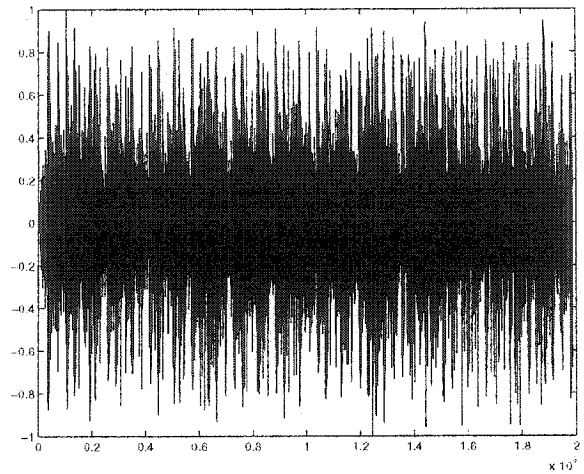


(b) The modified audio 03

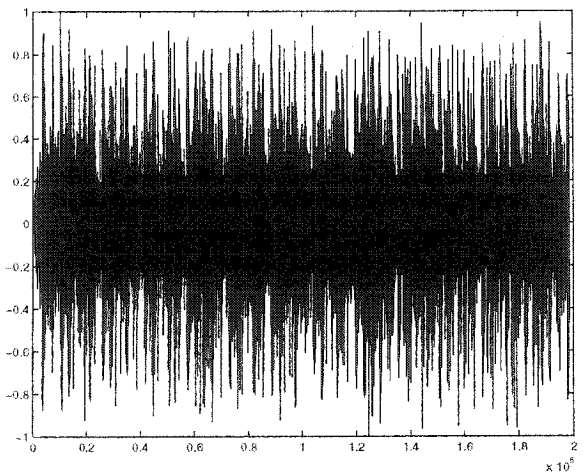


(c) The modified part.

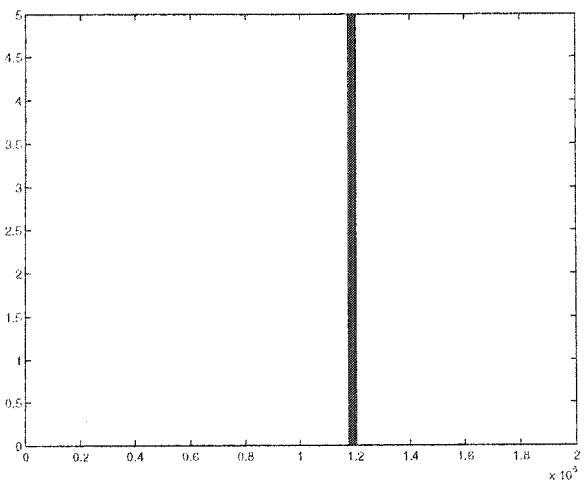
Figure 5.8: The results of substitution in audio 05 with string length of 1000 and start point of 1800000.



(a) The watermarked audio 03



(b) The modified audio 03



(c) The modified part.

Figure 5.9: The results of substitution in audio 06 with string length of 3000 and start point of 1200000.

Chapter 6

Conclusions and Future Works

Wavelet transform is an interesting technique. This domain has both spatial and frequency information simultaneously. In this thesis, we used this advantage of wavelet domain to embed the watermark. We made use of the frequency information to embed watermark robust against some attacks; while the spatial information is useful for content authentication. In our scheme, the watermark is embedded into wavelet domain of an audio by quantizing the wavelet coefficients.

The quantization technique makes the computation simple and the extraction procedure could be processed even without original audio signal. The quantization parameter Δ is experiment-based. We made a discussion on how to choose a proper value for this parameter. In extraction procedure, a classical matching filter is employed. The filter could locate the beginning of the original audio accurately since some attacks may change the length of the audio.

The testing results demonstrated that our watermarking scheme is robust to some attacks such as additive noise, filtering and MP3 compression. In the application of authentication, our scheme could not only check whether the audio is modified, it can also locate where it is modified. The main contribution of this thesis is the introduction

of authentication into audio signals.

As for our future work, We will improve our watermarking scheme so that we could get the knowledge of the quality of modified audio by checking the extracted watermark. The quality of an audio in our research work relates to its compression rate. The more an audio is compressed, the worth its quality is. In MP3 compression, with different compression rate, different frequency band will be cut from the audio signal. As our scheme could embed the watermark in all of the frequency band of an audio, it is possible to connect our watermark with the quality of the audio. Another further improvement is the method on quantization parameter selection. In our recent work, we select the quantization parameter manually. We hope we could develop an arithmetic method which can calculate the value of the parameters automatically.

Bibliography

[1] H. Berghel, Watermarking cyberspace, Communications of the ACM, Vol. 40, pp. 19-24, 1997.

[2] J.T. Brassil, S.H. Low, N.F. Maxemchuk, and L.O. Gorman, Electronic marking and identification techniques to discourage document copying, Proceedings of IEEE Infocom, pp. 1278-1287, 1994.

[3] D. Kundur and D. Hatzinakos, Towards a telltale watermarking technique for tamper-proofing, Proceedings of ICIP 1998, Vol. 2, pp. 409 -413, 1998

[4] C.S. Lu, H.Y.M. Liao, and L.H. Chen, Multipurpose audio watermaring, in Proc. 15th Int. Conf. Pattern Recognition, Barcelona, Spain, pp. 286-289, 2000.

[5] M. Wu, Multimedia data hiding, Ph.D. thesis, Princeton University, 2001.

[6] C. Xu, J. Wu, and Q. Sun, Digital audio watermarking and its application in multimedia database, IEEE international symposium on signal processing and its applications, Vol. 1, pp. 91-94, 1999.

[7] F. Hartung and M. Kutter, Multimedia watermarking techniques, Proceedings of the IEEE, Vol.87 Issue 7 , pp. 1079-1107, 1999.

[8] M. Swanson, B. Zhu, and A. Tewfik, Current state of the art, challenges and future directions for audio watermarking, IEEE International Conference on Multimedia Computing and Systems, Vol. 1, pp. 19-24, 1999.

[9] I.J. Cox, M.L. Miller, and J.A. Bloom, Digital watermarking, Morgan Kaufmann publishers, pp. 12-26, 1999.

[10] Secure Digital Music Initiative (SDMI): http://www. sdmi.org.

[11] S. Voloshynovskiy, S. pereira, T. Pun, J.J. Eggers, and J.K. Su, Attacks on digital watermarks: classification, estimation based attacks, and benchmarks, Communications Magazine, IEEE, Vol.39 Issue.8, pp. 118-126, 2001.

[12] F. Hartung, J.K. Su, and B. Girod, Spread spectrum watermarking: malicious attacks and conterattacks, Proceedings SPIE, Security and watermarking of Multimedia Contents, Vol.3657, pp. 147-158, 1999.

[13] M. Steinebach, F.A.P. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, and S. Seibel, StirMark benchmark: audio watermarking attacks, Proceeding. International Conference on Information Technology: Coding and Computing (ITCC '01), pp. 49-54, 2001.

[14] Checkmark benchmark: http://watermarking.unige.ch/Checkmark/.

[15] Certimark: www.igd.fhg.de/igd-a8/projects/certimark.

[16] http://ms-smb.darmstadt.gmd.de/stirmark/stirmarkbench.html.

[17] M.D. Swanson, M.Kobayashi, and A.H. Tewfik, Multimedia data embedding and wateramrking technologies, Proc. IEEE, Vol. 86, pp. 1064-1087, 1998.

[18] D. Gruhl, A. Lu, and W. Bender, Techniques for data hiding, IBM Systems Journal, Vol. 35. pp. 313-336, 1996.

[19] S. Czerwinski, R. Fromm, and T. Hodes, Digital music distribution and audio watermarking, UCB IS 219 (Tygar) project report, Spring 1999.

[20] M. Arnold, Audio watermarking: features, applications and algorithms, IEEE International Conference on Multimedia and Expo. Vol. 2, pp. 1013-1016, 2000.

[21] G. Voyatzis and I. Pitas, The use of watermarks in the protection of digital multimedia products, Proceedings of the IEEE, Vol. 87, pp. 1197-1207, 1999.

[22] L.F. Turner, Digital data security system, Patent IPN WO 89/08915, 1989.

[23] D.Gruhl, W. Bender, and A. Lu, Echo hiding, Proceedings of information hiding 1-st international workshop, pp. 231-265,1996.

[24] B.Ko, R. Nishimura, and Y. Suzuki, Time-spread echo method for digital audio watermarking using PN sequences, Proceedings on Acoustics, Speech, and Signal Processing, Vol. 2, pp. 2001-2004, 2002.

[25] Hyen O Oh, Jong Won Seok, Jin Woo Hong, and Dae Hee Youn, New echo embedding technique for robust and imperceptible audio watermarking, Proceedings of Acoustics, Speech, and Signal Processing, Vol. 3, pp. 1341-1344, 2001.

[26] Say Wei Foo, Theng Hee Yeo, and Dong Yan Huang, An adaptive audio watermarking system, Proceedings of IEEE Region 10 International Conference, Vol. 2, pp. 509-513, 2001.

[27] J.K. Su and B. Girod, Power-spectrum condition for energy-efficient watermarking, IEEE Transactions on Multimedia, vol. 4, no. 4, pp. 539-560, 2002.

[28] I.J.Cox, J. Kilian, F. T. Leighton, and T.Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing, Vol. 6 Issue. 12, pp. 1673 -1687, 1997.

[29] P. Bassia, I. Pitas, and N. Nikolaidis, Robust audio wateramrking in the time domain, IEEE Transactions on Multimedia, Vol. 3 Issue. 2, pp. 232-241, 2001.

[30] D. Kirovski and H.S. Malvar, Spread-spectrum watermarking of audio signals, IEEE Transactions on Signal Processing, Vol. 51 Issue. 4, pp. 1020-1033, 2003.

[31] S. Cheng, H. Yu, and Z. Xiong, Enhanced spread spectrum watermarking of MPEG-2 AAC audio, IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 3728-3731, 2002.

[32] Q. Cheng and J. Sorensen, Spread spectrum signaling for speech watermarking, IEEE International Conference on Acoustics, Speech, and Signal Processing. , Vol. 3, 1345-1348, 2001.

[33] M. Ikeda, K. Takeda, and F. Itakura, Audio data hiding by use of band-limited random sequences, IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2315-2318, 1999.

[34] L. Boney, A. Tewfik, and K. ha,dy, Digital watermarks for audio signals,

Proceddings of the third IEEE International Conference on Multimedia Computing and Systems, PP. 473-480, 1996.

[35] N. Cvejic, A. Keskinarkaus, and T. Seppanen, Audio watermarking using m-sequences and temporal masking, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 227-230, 2001.

[36] W. Lie adn L. Chang, Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking, The 2001 IEEE International Sysmposium on Circuits and Systems, Vol. 2, pp. 45-48, 2001.

[37] Hong Oh Kim, Bae Keun Lee, and Nam-Yong Lee, Wavelet-based audio watermarking techniques: robustness and fast synchronization.

[38] S. Lee and Y. Ho, Digital audio watermarking in the cepstrum domain, IEEE Transaction on Consumer Electronics, Vol. 46, pp. 334-335, 2000.

[39] G.L. Friedman, The trustworthy digital camera: restoring credibility to the photographic image, IEEE Transaction on Consumer Electronics, Vol. 39, pp. 905-910, 1993.

[40] S. Walton, Image authentication for a slippery new age, Dr. Dobb's Journal, Vol. 20, pp. 18-26, 1995.

[41] M. Wu and B. Liu, Watermarking for image authentication, Proceedings of ICIP 98, Vol. 2, pp. 437-441, 1998.

[42] D. Kundur and D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, Proceedings of the IEEE, Vol. 87, No. 7, pp. 1167-1180, 1999.

[43] C. Lu and H. Liao, Multipurpose watermarking for image authentication and protection, IEEE Transactions on Image Processing, Vol. 10, No. 10, pp. 1579-1592, 2001.

[44] C. Hsu and J. Wu, Hidden digital watermarks in images, IEEE Transactions on Image Processing, Vol. 8, No. 1, pp. 58-68, 1999.

[45] R.M. Gray, D.L. Neuhoff, Quantization, IEEE Transctions on Information theory, Vol. 44 Issue. 6, pp. 2325-2383, 1998.

[46] J. Horner and P. Gianino, Phase-only matched filter, Applied Optics, Vol. 23, No. 6 pp. 812-816, 1984.

[47] Y. Liu, X. Wu, D. Zheng, J. Zhao and J. Yao, Phase information in RST invariant image watermarking, submitted to SPIE Electronic Imaging.

[48] L.G. Brown, A survey of image registration techniques, ACM Computing Surveys, Vol.24, No. 4 pp. 325-376, 1992.

[49] http://amsl-smb.cs.uni-magdeburg.de.

[50] R. Tu and J. Zhao, A novel semi-fragile audio watermarking scheme, IEEE International Workshop on Haptic, Aduio and Visual Environments and their Applications, Ottawa, Ontario, Canada, 2003.

[51] R. Tu and J. Zhao, A Semi-fragile audio watermarking scheme based on wavelet transform and quantization, submitted to Signal Processing.