Reversible watermarking scheme with watermark and signal robustness for audio

By:

María Alejandra Menéndez Ortiz

A dissertation submitted in partial fulfillment
of the requirements for the degree of:

DOCTOR OF SCIENCE IN COMPUTER SCIENCE

at

Instituto Nacional de Astrofísica, Óptica y Electrónica
June, 2017
Tonantzintla, Puebla

Supervised by:

Dr. Claudia Feregrino Uribe, INAOE
Dr. José Juan García Hernández, CINVESTAV

Agradecimientos

Agradezo profundamente a mi familia, por su apoyo incondicional y sus constantes enseñanzas. A mis padres, Angélica Ortiz Cabrera y Miguel Ángel Menéndez López, mi abuela, mis hermanos y mis sobrinos.

En estas simples líneas, quiero reconocer el invaluable apoyo y dirección de los doctores Claudia Feregrino Uribe y José Juan García Hernández, quienes con sus sabios consejos, paciencia y orientación me ayudaron a concluir este trabajo. Asimismo, agradezco a mis sinodales, doctores: Dinu Coltuc, René Cumplido Parra, Aurelio López López, Gustavo Rodríguez Gómez y Saúl Pomares Hernández, por sus sugerencias al trabajo de investigación y sus amables comentarios.

A mis amigos, por su tiempo, comprensión y compañía; ha sido un honor compartir esta amistad con ustedes. No me serían suficientes las líneas para mencionar a todos, pero quiero agradecer especialmente a Brisa Edith Méndez Cruz, Rigoberto Fonseca Delgado, Liliana Perea Centeno, Marisol Flores Garrido, Emmaly Aguilar Pérez, Gabriela Rodríguez Ruiz, Ana Patricia Torres, Anaely Pacheco Blanco y Andrea Muñoz Potosí.

Al INAOE, por haberme otorgado la oportunidad de realizar mis estudios de posgrado en esta reconocida institución. De igual manera, reconozco la excelente labor de todos los trabajadores de este instituto.

Agradezo al pueblo de México, que a través del CONACYT me fue otorgada la beca No. 351601, con la cual tuve la oportunidad de realizar este posgrado.

Dedicatoria

Así como mi vida, trabajo y esfuerzo, dedico este trabajo a mi Creador.

Con gran amor, respeto y admiración dedico esta tesis
a la memoria de María Luisa López López,
mi más grande amiga y abuela; vivirás por siempre en mi.

Abstract

In recent years, Internet use has increased the unauthorized distribution of multimedia content and digital watermarking is a means to protect these contents from illegal actions. However, conventional watermarking produces distortions to the watermarked signals that cannot be counteracted, which are not acceptable for applications in the medical or military field. Reversible watermarking schemes (RWS) allow to counteract such distortions; although they are fragile, which means that any modification to the watermarked signals will cause the loss of the embedded watermarks as well as the original signal.

In this doctoral research, a reversible watermarking scheme with watermark and signal robustness is proposed. It is a contribution to the state of the art because there are no such schemes in the literature. A framework is proposed to address this problem, that was validated through experimental results with images and audio signals, and it is an abstract construction to solve the research problem. It was assumed that since the framework proved to be effective for images, it could be followed in the same way to propose a solution for audio signals.

The framework consists of two stages, namely a fragile reversible one and a self-recovery one. The fragile reversible stage allows the insertion of the watermark and the self-recovery one inserts control information that allows the restoration after the watermarked signals have been attacked with content replacement. To implement the framework for audio signals, a reversible watermarking scheme and a self-recovery scheme had to be proposed. The self-recovery scheme is a contribution to the state of the art because, to the best of our knowledge, the scheme proposed is the first effort for this type of media.

The reversible watermarking scheme for audio signals uses the auditory masking properties of the signals to determine the frequencies where embedding of the watermark will cause unnoticeable distortions. Moreover, the strategy from the reversible watermarking scheme proposed was also included in the self-recovery scheme proposed, and it allowed the insertion of the control information that was required for perfect restoration of the signals after content replacement was applied. With perfect

restoration, the framework could be completed and tested for audio signals.

The results obtained with the framework indicate that the scheme has adequate transparency for copy distribution applications, as indicated by the ODG results obtained over -2 ODG for all the tested datasets. The restoration capabilities of the scheme show that perfect reconstruction of the original signals and perfect extraction of the watermarks can be obtained for content replacement attack of 0.1%, and approximate reconstruction can be obtained for greater percentages, which were tested up to 0.3%. The payload capacity that can be inserted to maintain the required restoration capabilities is $\leqslant 2$ kbps. With payloads > 2 kbps and $\leqslant 13$ kbps the scheme allows approximate reconstruction of the signals with high auditory quality, and the watermarks are extracted with an acceptable error rate, that can be compensated with error correction codes.

Resumen

En años recientes, el uso de Internet ha incrementado la distribución no autorizada de contenido multimedia y las marcas de agua digitales han sido utilizadas para proteger estos contenidos de acciones ilegales. Sin embargo, los esquemas convencionales producen distorsiones sobre las señales marcadas que no pueden ser contrarrestadas, lo cual no es aceptable para aplicaciones en los campos médicos y militares. Los esquemas de marcas de agua reversibles (RWS, por sus siglas en inglés) permiten contrarestar estas distortiones, aunque son frágiles, lo que significa que cualquier modificación a las señales marcadas causará la pérdida de las marcas insertadas así como de las señales originales.

En esta investigación doctoral, se propone un esquema reversible de marcas de agua con robustez de la marca y de la señal. Es una contribución al estado del arte porque tales esquemas no existen en la literatura. Se propone un marco conceptual para abordar el problema, éste fue validado a través de resultados experimentales con imágenes y señales de audio, y éste es una abstracción para resolver el problema de investigación. Se partió de la suposición que, dado que el marco conceptual demostró ser efectivo para imágenes, éste podría utilizarse de la misma manera para una solución en señales de audio.

El marco conceptual consiste en dos etapas, una frágil reversible y otra auto-recuperable; la etapa frágil reversible permite insertar la marca de agua y la etapa auto-recuperable inserta información de control que permite la restauración después que las señales marcadas han sido atacadas con reemplazo de contenido. Para implementar el marco para señales de audio, un esquema reversible de marcas agua y un esquema auto-recuperable para señales de audio tuvieron que ser propuestos. El esquema auto-recuperable es una contribución al estado del arte, dado que éste es el primer esquema de este tipo para señales de audio.

El esquema reversible de marcas de agua para señales de audio utiliza las propiedades de enmascaramiento auditivo para determinar las frecuencias donde la inserción de la marca produce distorsiones que no son notorias. Aún más, la estrategia usada por el esquema reversible de marcas de agua también fue incluida en el esquema

auto-recuperable propuesto, misma que permitió la inserción de la información de control requerida para la restauración perfecta de las señales después que el ataque de reemplazo de contenido. Con la restauración perfecta, el marco conceptual pudo ser completado y probado para señales de audio.

Los resultados obtenidos con el marco conceptual para señales de audio indican que el esquema tiene una transparencia adecuada para aplicaciones de distribución de copias, como lo indican los resultados de ODG obtenidos y que son superiores a -2 ODG para todas las bases de datos probadas. Las capacidades de restauración del esquema muestran que se puede obtener reconstrucción perfecta de las señales originales y extracción perfecta de las marcas de agua para ataques de reemplazo de contenido del 0.1%, y se puede obtener reconstrucción aproximada de las señales con porcentajes mayores, que fueron probados hasta un 0.3%. La carga útil que puede ser insertada para mantener las capacidades de restauración son $\leqslant 2$ kbps. Con cargas útiles > 2 kbps y $\leqslant 13$ kbps el esquema permite obtener reconstrucción aproximada de las señales con alta calidad, y las marcas de agua se extraen con tasas de error aceptables, que pueden ser compensadas con códigos de corrección de errores.

Contents

List of Figures

List of Tables

Acronyms

AWGN Additive White Gaussian Noise. 21, *Glossary:* AWGN

BER Bit Error Ratio. 7, *Glossary:* BER

DCT Discrete Cosine Transform. 21, *Glossary:* DCT

DRM Digital Rights Management. 1, *Glossary:* DRM

HAS Human Auditory System. 36, *Glossary:* HAS

intDCT Integer Discrete Cosine Transform. 16, *Glossary:* intDCT

IWT Integer Wavelet Transform. 20, *Glossary:* IWT

LSB Least Significant Bit. 16

MPEG Moving Picture Experts Group. 24, *Glossary:* MPEG

MSE Mean Squared Error. 7, *Glossary:* MSE

ODG Objective Difference Grade. ii, *Glossary:* ODG

PEAQ Perceptual Evaluation of Audio Quality. 12, *Glossary:* PEAQ

PEE Prediction Error Expansion. 13, *Glossary:* PEE

PSNR Peak Signal to Noise Ratio. 7, *Glossary:* PSNR

RWS Reversible Watermarking Scheme. i, *Glossary:* RWS

SDG Subjective Difference Grade. 12, *Glossary:* SDG

segSNR Segmental Signal to Noise Ratio. 17, *Glossary:* segSNR

SNR Signal to Noise Ratio. 11, *Glossary:* SNR

Glossary

AWGN The Additive White Gaussian Noise (AWGN) is a noise model that mimics the addition of white Gaussian noise. 21

BER The Bit Error Ratio (BER) is a metric that indicates the percentage of bit errors from a total number of transmitted bits. 7

DCT The Discrete Cosine Transform (DCT) expresses a finite signal as a sum of cosine functions at different frequencies. 21

DRM Digital Rights Management (DRM) systems are designed to control and restrict access to multimedia contents. 1

HAS The Human Auditory System (HAS) is the sensory system for human hearing. 36

intDCT The Integer Discrete Cosine Transform (intDCT) is an impairment scale to evaluate subjective listening tests. 16

IWT The Integer Wavelet Transform (IWT) is the integer approximation for the Wavelet Transform. 20

MPEG The Moving Picture Experts Group (MPEG) is a group that works on developing standard codecs for digital video and audio media. 24

MSE The Mean Squared Error (MSE) is an statistical metric that measures the average sum of the squared errors. 7

ODG The Objective Difference Grade (ODG) is an objective metric used to evaluate audio quality based on a psychoacoustic model of human hearing. ii

PEAQ The Perceptual Evaluation of Audio Quality (PEAQ) is an standardized algorithm to evaluate perceived audio quality as an objective measure. 12

PEE The Prediction Error Expansion (PEE) technique is one of the classical strategies used in reversible watermarking schemes. 13

PSNR The Peak Signal to Noise Ratio (PSNR) is metric that measures the maximum signal to noise ratio found on a signal. 7

RWS A reversible watermarking scheme (RWS) is a type of watermarking scheme that allows the counteraction of modifications caused by the embedding process. i

SDG The Subjective Difference Grade (SDG) is an impairment scale to evaluate subjective listening tests. 12

segSNR The Segmental Signal to Noise Ratio (segSNR) is the average SNR measured from short segments of the signal. 17

SNR The Signal to Noise Ratio (SNR) is a metric that measures the ratio between the signal power and the noise power. 11

Chapter 1

Introduction

In this chapter, the basic idea of digital watermarking is introduced, as well as reversible watermarking and the necessity of such schemes. An introduction to the work developed in this doctoral research is also given, and the scenarios that motivate the research are explained. The problem statement, the research questions, hypothesis, objectives, methodology and contributions of the thesis are stated in this chapter.

Digital media such as video, audio and images, is easily transmitted, manipulated and commercialized through digital channels. However, given the facility of manipulation it is easy to copy, modify or distribute digital media in an illegal manner. These illegal actions are known as piracy and looking for a way to counteract them, the Digital Rights Management (DRM) systems were created. These systems allow to control and restrict the access to digital multimedia; they include encryption, conditional access, copy control mechanisms, and media identification and tracing mechanisms. Digital watermarking is a key technology in these systems and is used for copy control, and media identification and tracing [Hartung and Ramme, 2000].

Digital watermarking schemes insert a secret message into a host signal in a way that is imperceptible for a human observer but that can be recovered given an extraction algorithm. However, conventional watermarking schemes produce a distortion in the carrier signal that causes loss of data. There are applications in the medical and military field where it is imperative that the carrier signal does not suffer loss of data and in those applications conventional watermarking schemes are not suitable.

With that situation, reversible watermarking schemes (RWS) arose. These schemes can insert a secret message within a carrier signal and later the modifications suffered during insertion can be reversed in order to obtain the host signal. However, reversible watermarking schemes can only reconstruct the host signal if the watermarked version does not suffer any additional modifications, *i.e.* if the watermarked signal is not submitted to attacks. If a signal marked with a reversible watermarking scheme goes through a modification, then the reversibility of the system is lost and the host signal cannot be reconstructed.

1.1 Motivation of proposal

There are some scenarios of great importance for society, such as military. In this scenario, suppose there is a pilot in a high-performance aircraft that operates in a heavy workload environment, where hands and eyes are busy. Speech recognition could allow the pilot to send instructions to choose weapons or dictate other commands to a base with automated speech recognition [Weinstein, 1991]. An enemy could interfere the transmission channel and send false commands, so the speech of the pilot could be protected by the inclusion of security codes into the speech (watermark). The substitution of the whole speech could be a complicated task for the enemy party but instead, they could substitute prerecorded commands that would be recognized by the automatic detection. For such scenario, it is necessary to have a scheme that allows the insertion of security codes to speech instructions of the pilot, and that allows the restoration of the original commands in case these were modified by an enemy party. Since the speech is to be processed by an automated system, the quality of the restored signals is crucial for speech recognition.

In a commercial scenario, suppose there is an on-line music distribution service, that distributes songs with enriched content. Free users can download "previews" of the complete songs with super-imposed slogans. With the purchase of a premium account, the customer is able to remove the slogans and extract the enriched content from the songs. The super-imposed slogans are short duration sounds dispersed throughout the song. The objective of the slogans is to "cover" regions of the original music and make it annoying for the listener, without modifying the duration of the songs. A similar scenario is one for free distribution of censored music, where it can be uncensored with the purchase of a key to remove censorship. There are songs that contain inappropriate language; for these songs to be distributed, the inappropriate content has to be censored by editing the songs. Offending content is removed through re-sampling, bleeping, and replacing words with silence, sound effects or single tones [Newton, 2012]. In a music distribution scenario, censored songs could be freely distributed but premium users could pay a fee to remove the censorship.

In the current literature, there are robust reversible watermarking schemes that allow the extraction of watermarks after the signals have suffered attacks [An et al., 2012a, Tsai et al., 2010, Chang et al., 2009]. With such schemes, the security codes or the enriched content could be extracted, but the original speech and music would be lost, which is not acceptable for any of the stated scenarios. Self-recovery schemes allow the restoration of the signals after they have suffered attacks. However, these schemes do not provide the capacity to insert watermarks, and they are not suitable for the mentioned scenarios either.

From this, it is clear that currently no solution exists that solves the problem of a reversible watermarking scheme with watermark and signal robustness for audio signals. A detailed specification of the problem that is approached in this investigation is given below.

1.2 Problem statement

Applications that require transmission of high-quality audio signals along with additional data require the following characteristics (Fig. 1.1):

- The encoding process inserts a secret message $\mathbf{m}$ into a host signal $\mathbf{x}$, this message contains information related to the signal. The insertion of the message, also known as watermark, produces a degradation (ϕ) to the watermarked signal $\mathbf{y}$, this degradation should be acceptable for practical applications.

- The transmission channel is a non-binary one, where one attack is considered to occur, namely content replacement. Content replacement attacks impose a distortion (ψ,) caused by the substitution of certain regions of the watermarked signal with regions of the same size from another signal, the substituted regions can also be random noise. The signal produced after content replacement is an attacked signal $\hat{\mathbf{y}}$.

- The decoding process should be able to counteract the modifications caused by the content replacement attacks; *i.e.* the modified regions in $\hat{\mathbf{y}}$ must be restored, in order to obtain the watermarked signal $\mathbf{y}$. From the watermarked signal, an extraction process should extract the secret message $\mathbf{m}$ that was originally embedded in the encoding process; and finally, a restoration process can reverse the modifications caused by embedding, to restore the original signal $\mathbf{x}$.

A system that meets all the previous restrictions does not currently exist. The design of a reversible watermarking scheme that satisfies these characteristics remains as an open problem and is the focus of this doctoral research.

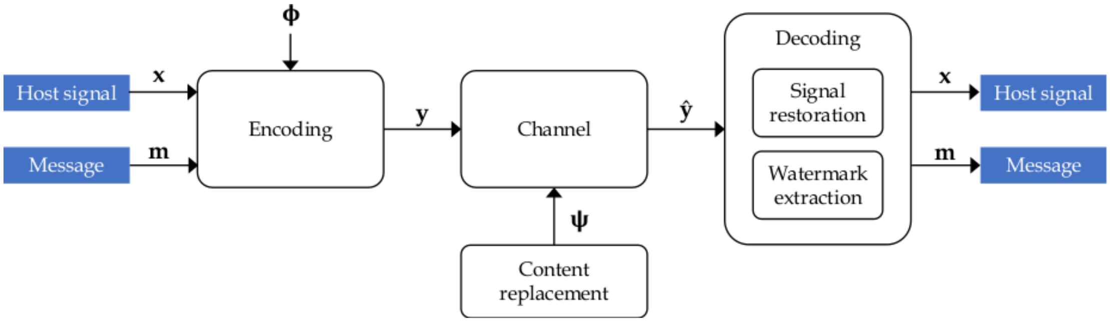


Figure 1.1: Elements in the problem situation.

1.3 Research questions

- How to design a reversible watermarking scheme that allows to extract embedded watermarks and perfectly reconstructs a host audio signal from a watermarked audio that was modified with a content replacement attack?

- How to construct the control data of an embedding process in such a way that this control data helps an extraction process in the reconstruction of a host audio that suffered a content replacement attack?

- How to design a mechanism that finds the locations in a watermarked audio where content replacement occurred and that compensates these modifications using the control data previously embedded?

1.4 Hypothesis

A reversible watermarking scheme with watermark and signal robustness can be designed though a framework, which considers an encoding and decoding process, and each process is further divided in two stages: a fragile reversible watermarking stage, and a self-recovery stage. Such robust reversible watermarking scheme for audio signals can be proposed following that framework. By designing a self-recovery watermarking scheme for audio signals, the framework can be implemented, and therefore obtaining the scheme that fulfils the objectives of this research.

An important property that the proposed scheme should meet is the transparency of the watermarked signals. The encoding process of the proposed scheme should produce signals that have been modified in a way that a human listener cannot distinguish. A metric to measure this transparency in audio signals is the Objective Difference Grade (ODG), and an acceptable value for an application such as copy distribution is of -2 ODG.

1.5 Objectives

General objective

Design a reversible watermarking scheme with watermark and signal robustness that can restore t second from a music audio signal of length T, and that has been tampered with a content replacement attack, maintaining an embedding transparency over -2 ODG, with $t = 1$ second, and $T = 5$ minutes.

Specific objectives

- Design an embedding mechanism that inserts control data into music audio signals with a distortion over -2 ODG, for its use in copy distribution applications.
- Design a restoration mechanism that is able to reconstruct 1 second every 5 minutes from a tampered audio signal attacked with content replacement.

1.6 Methodology

The steps that were followed to propose a reversible watermarking scheme with watermark and signal robustness for audio are outlined in Figure 1.2, and are detailed below. In the digital watermarking research line, most of the advances to the state of the art are proposed for images. Later, the general ideas and strategies for such media are generalized and extended to another type of media, such as audio. For this reason, the proposed methodology was followed using watermarking strategies for images, since the existing strategies were developed for this type of media, in order to propose and test a solution for the problem previously stated. Once a satisfactory solution for images was designed, then the general ideas were adapted for audio signals.

1. **Select relevant schemes.** In this step, current works in the literature related to the problem were selected and studied, in order to understand their underlying concepts and how those concepts could be used to solve the problem stated.
2. **Identify promising strategies.** Given the watermarking schemes selected before, some strategies were developed. Those strategies took consideration on the

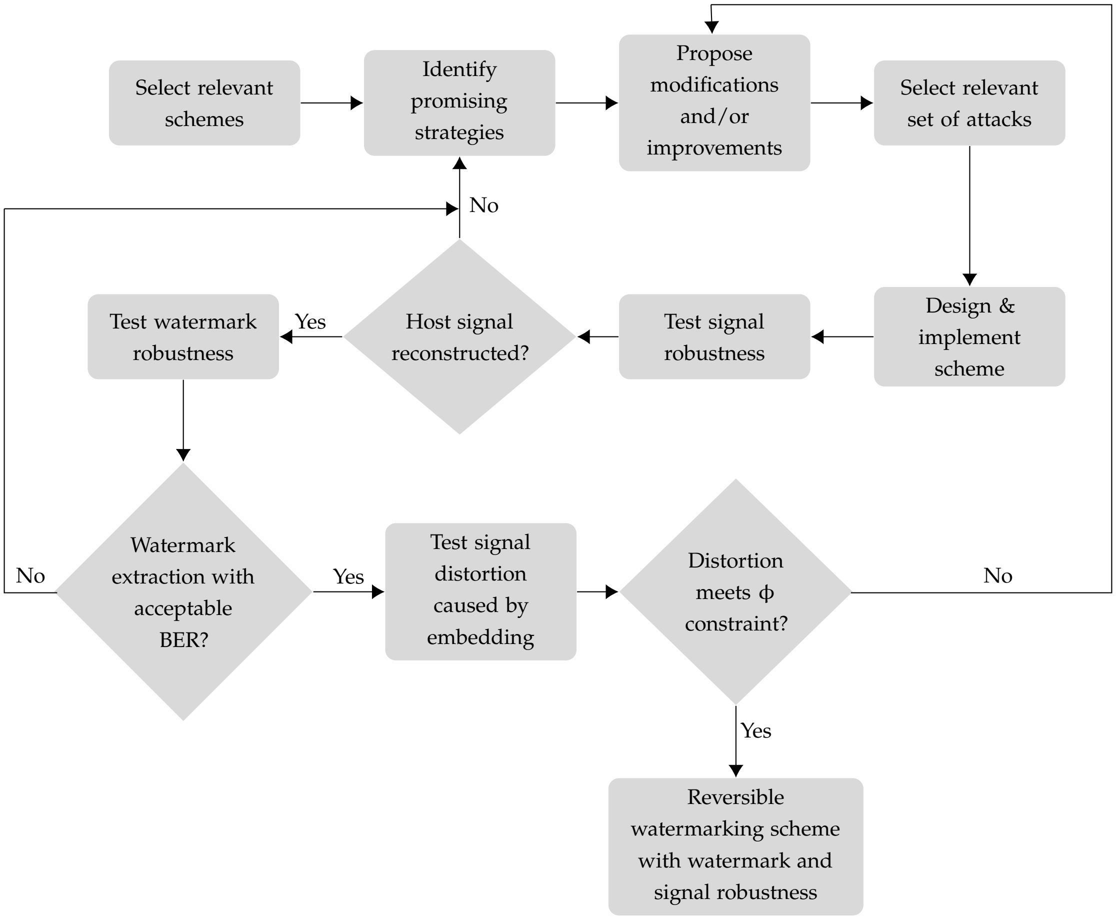

Select relevant schemes
Identify promising strategies
Propose modifications and/or improvements
Select relevant set of attacks
Design & implement scheme
Test signal robustness
Host signal reconstructed?
No
Yes
Test watermark robustness
Watermark extraction with acceptable BER?
No
Yes
Test signal distortion caused by embedding
Distortion meets ϕ constraint?
No
Yes
Reversible watermarking scheme with watermark and signal robustness

Figure 1.2: Steps in the proposed methodology.

fundamental ideas of the schemes, and how these ideas could be exploited, modified and extended to be incorporated into the proposed strategies.

3. **Propose modifications and/or improvements.** Once the possible strategies were stated, modifications and/or improvements to those concepts were designed. The new strategies were aimed to be incorporated into the proposed solution to fulfill the goals of the research.

4. **Select relevant set of attacks.** A detailed analysis of the problem addressed in this work was done, and the restrictions of the problem were clearly outlined. These restrictions are set by the trade-off between imperceptibility, payload capacity, robustness and perfect restoration, which is imposed by the reversibility property and it is the most severe requirement of the scheme. Because of this delicate trade-off, an initial set of possible attacks were selected; by further analyzing the restrictions and the embedding capacity required for perfect restoration, the

final attack considered in this work is content replacement.

5. **Design and implement scheme.** Taking into consideration the attack of content replacement, the improved and/or modified strategies were further adapted, in order to design and implement a reversible watermarking scheme with watermark and signal robustness. To do so, a framework was proposed; this framework includes two stages, namely a fragile reversible watermarking stage, and a self-recovery stage. An initial test of concept was carried out with images, proposing a framework for this type of media, in order to test the validity that this construction had to achieve the objectives of the research. Once the framework proved to be reliable, it was implemented for audio signals.

6. **Test signal robustness.** In this step, the signals watermarked with the proposed framework were then attacked. After applying these attacks, the decoding process of the framework was applied to try to restore the original host signals. Since reversibility is required, perfect restoration must be obtained, *i.e.*, the signals restored after the decoding process must be identical to the host ones. Several strategies were tested, modified and improved before reaching perfect restoration for audio signals. Once perfect restoration was achieved, the tests to evaluate watermark robustness were performed.

7. **Test watermark robustness.** With perfect restoration, the complete framework was tested to evaluate the robustness of the watermark embedded. After the signals were attacked, then restored with the decoding process of the framework, the watermarks were extracted and compared against the watermarks originally embedded, in order to determine the bit error rate (BER) between them. The obtained BER results were unacceptable only when perfect restoration was not achieved, in such cases, the scheme had to be modified to improve the robustness of the signal, to obtain the required results. The final tests were performed to evaluate the global distortion caused by the encoding process of the framework, and to ensure that an encoding degradation constraint ϕ (see Fig. 1.1) was meet. The distortion constraint was determined based on the target applications for the framework, and was set to -2 ODG.

8. **Test signal distortion caused by embedding.** In this final set of experiments the signals watermarked with the proposed framework were compared against the host signals to evaluate the differences between them, using the peak signal to noise ratio (PSNR), mean squared error (MSE), and objective difference grade (ODG). The initial strategies of the framework did not allow the insertion of the

required control information with the adequate transparency; therefore, these strategies had to be modified in order to comply with this constraint. The use of the auditory masking properties of audio signals allowed the proper behavior of the framework, complying with the distortion constraint; this strategy is the one that allowed the design of the final reversible watermarking scheme with watermark and signal robustness for audio signals.

1.7 Contributions

- General framework. This work proposes a general framework that permits the design of a reversible watermarking scheme with watermark and signal robustness. By the use of a fragile reversible stage, and a self-recovery stage, the framework allows the insertion of useful payload as a watermark; by means of the self-recovery strategy, the watermarked signals and the embedded watermark are protected against content replacement attack. The general framework is an abstraction of the solution, that proved to be valid through the experimental results obtained in this work.

- Framework for images. The first implementation of the framework was proposed for images. Through this implementation, the initial results that proved the validity of the framework were obtained. It is a contribution to the state of the art, because this construction is, as far as we know, the first effort to propose a reversible watermarking scheme with watermark and signal robustness in the literature.

- Self-recovery scheme for audio signals. In order to complete the framework for audio signals, a self-recovery scheme for audio signals had to be proposed. To the best of our knowledge, the scheme proposed in this work is the first effort done for this type of media.

- Reversible watermarking scheme with watermark and signal robustness for audio signals. The completion of the framework for audio signals allowed the construction of a reversible watermarking scheme with watermark and signal robustness, which is the goal of this research.

The rest of the document is organized in the following manner. Chapter 2 introduces the most common evaluation metrics for digital watermarking schemes, and provides a revision of the current literature for fragile reversible watermarking for both images and audio signals. The robustness property of reversible watermarking schemes for images and audio signals that exist in the state of the art is also presented. Chapter 3 presents the construction proposed in this work to design the required reversible watermarking scheme, the abstract framework is explained, as well as its implementation for images and for audio. Chapter 4 gives a detailed explanation of the proposed reversible watermarking scheme for audio, the necessity to design such a scheme is stated, and the strategies used for its construction are explained in depth. Chapter 5 explains the proposed self-recovery scheme that is designed in this work; the role that the reversible watermarking schemes from the previous chapter has on this construction has is given as well. The encoding and decoding processes that comply the self-recovery scheme for audio signals are explained in detail. Chapter 6 presents the results obtained with the audio reversible watermarking scheme, the audio self-recovery scheme, and finally the framework implemented for audio. Finally, Chapter 7 presents a summary of this doctoral research, the conclusions of the work are stated, as well as the future work derived from it.

Chapter 2

State of the art

This chapter begins with the introduction of the most common properties in a digital watermarking scheme. A detailed revision of the literature for reversible watermarking for images and audio signals is presented. Also, a detailed exploration in the state of the art with regard to the robustness property that exists in watermarking schemes for images and audio signals is given in this chapter.

The most important requirements in a digital watermarking scheme are the following:

- **Fidelity.** It is the similarity between an original (not watermarked) signal and a watermarked one. This similarity can be measured by a statistical metric, such as the *mean squared error* (MSE) which measures the difference between the original signal $\mathbf{x}$ and watermarked signal $\mathbf{y}$, it is calculated by [Gonzalez and Woods, 2011]:

$$\text{MSE} = \frac{1}{N} \times \sum_{n}^{N} (\mathbf{x}[n] - \mathbf{y}[n])^2, \tag{2.1}$$

  where N is the total number of samples in the signals, $\mathbf{x}[n]$ corresponds to the n^{th} sample of the original signal $\mathbf{x}$ and $\mathbf{y}[n]$ is the n^{th} sample of the watermarked signal $\mathbf{y}$; the closer the MSE value is to zero, the greater the similitude between signals. Another common metric is the *peak signal to noise ratio* (PSNR). The PSNR evaluates the difference between the original and the watermarked signals measuring the maximum signal to noise ratio found in a signal, and it is given by [Gonzalez and Woods, 2011]:

$$\text{PSNR (dB)} = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}}, \tag{2.2}$$

  where MAX is the maximum possible value of a sample. The *signal to noise ratio* (SNR) is the most common metric to measure the difference between audio signals, and is given by [Gonzalez and Woods, 2011]:

$$\text{SNR (dB)} = 10 \log_{10} \frac{\sum_n^N \mathbf{x}[n]^2}{\sum_n^N (\mathbf{x}[n] - \mathbf{y}[n])^2}. \tag{2.3}$$

An acceptable noise distortion for audio signal is of 35 dB [Lu, 2004]. However, the metrics mentioned above give an evaluation in terms of the numerical values of the signals being compared, but not in terms of the quality perceived by human listeners. In order to evaluate this, a perceptual quality assessment has to be performed [Lin and Abdulla, 2015], it can be classified into two categories: subjective listening tests by human acoustic perception and objective evaluation tests by perception modeling. In subjective tests, a five-grade impairment scale is given to the human listeners to evaluate the watermarked signals; this scale is known as the subjective difference grade (SDG) and its values are described in Table 2.1. Nonetheless, subjective tests are costly, time-consuming, and greatly depend on the subjects and surrounding conditions; therefore, an objective evaluation is preferred. The Perceptual Evaluation of Audio Quality (PEAQ) is used to provide an objective difference grade (ODG), which is an objective measure of the SDG and its scores are the same as those in Table 2.1. PEAQ establishes an auditory perception model to imitate the listening behavior of a human.

Table 2.1: Subjective difference grade (SDG).

Difference grade	Description of impairments
0	Imperceptible
-1	Perceptible but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

- **Data payload.** The data payload refers to the number of bits that a watermark encodes within a unit of time or a signal. A scheme that encodes N bits can be used to embed 2^N messages. In audio, the data payload is measured by the number of bits that can be encoded in one second, i.e., the number of *bits per second* (bps), also known as bit rate.

- **Robustness.** The robustness is the ability to detect a watermark after a watermarked audio has been subjected to modifications. These modifications can be common signal processing operation (non intentional attack) or intentional attacks. Some examples of non intentional attacks for audio are MP3 compression, additive noise, re-sampling, filtering, analogue-to-digital conversion (A/D), digital-to-analogue conversion (D/A), among others. There also exist some processes where attackers intentionally remove the watermarks or intentionally analyse the watermarked audio in order to estimate the embedded watermark, which is known as a collusion attack.

Depending on the application of the watermarking scheme some properties might be irrelevant or have a broader tolerance. Watermarking schemes are not designed in a way that all the properties comply the best results, but rather the most important ones are exploited in order to be better suited for certain applications.

2.1 Fragile reversible watermarking schemes

In recent years, proposed reversible watermarking schemes have tried to improve payload capacity and imperceptibility. Payload capacity, on the one hand, is difficult to increase because reversible watermarking schemes need control information to be able to restore the host signal after extraction and the actual space for data watermarking is significantly reduced. Imperceptibility, on the other hand, is hard to achieve given the payload capacities needed by current applications. When the payload capacity is increased, the imperceptibility of the scheme is reduced since more information is embedded, and therefore, the signal suffers more distortion. To increase the imperceptibility, the payload capacity has to be reduced in order to diminish the distortion of the watermarked signal. The trade-off between payload capacity and imperceptibility is further challenged in reversible schemes that also need to embed, along the watermarks, the control information. Reversible schemes must design strategies to reduce the size of the control information in order to reduce perceptual impact, but maintaining the original information to provide reversibility.

2.1.1 Summary of RWS for images

Some of the most relevant reversible watermarking schemes for images are summarized in this section. A recent strategy for RWS is prediction-error expansion (PEE), and it has reported better performance in terms of embedding capacity and image degradation than the performance achieved by other strategies such as difference expansion, or histogram shifting reversible watermarking. PEE schemes can insert up to 1 bit per pixel (bpp) with an acceptable image degradation.

The scheme proposed by Sachnev et al. [2009] uses a PEE strategy with rhombus prediction for data hiding; a histogram shifting technique is used to improve the performance of the scheme. The distortion caused by embedding is reduced by the use of a sorting technique that selects de prediction errors that cause lower visual distortion. The embedding capacity is increase to approximately 1 bpp through the use of a cascade embedding strategy. An appropriate performance depends on the assumption that local variance among neighboring pixels will be small and the error will therefore also be small.

Coltuc [2011] proposes a scheme based on PEE that inserts information not only in the prediction error calculated between the pixel and its predicted value, but also on the prediction context of the pixel. The optimization is applied to three image predictors, namely MED, GAP, and SGAP. The results obtained in this work demonstrate that the optimization on SGAP outperforms GAP schemes in terms of higher PSNR with the same payload and lower mathematical complexity.

Li et al. [2013] present a RWS where a general framework for histogram shifting with PEE is developed; by adjustment of the shifting and expansion functions, new schemes can be adapted. Two schemes derived from this framework are proposed. One of the schemes utilizes a predictor obtained from nine neighboring pixels, which is a modification to the rhombus predictor used by Sachnev et al. [2009], this new predictor extends Sachnev's predictor to include diagonal pixels to calculate de predictor. Smooth pixels are selected for embedding in order to reduce perceptual impact.

Dragoi and Coltuc [2014] propose a RWS with PEE, this scheme uses local prediction to estimate the predictors used for both embedding and detection. The local predictors are calculated for smaller regions of the image instead of global predictors used in most PEE schemes, which use a same predictor for the whole image. Local predictors are adapted for blocks from the image, and because of the strategy used for its construction, they allow detection without the need to insert any further information. The least square predictor of each pixel is obtained in a square block centered at the pixel. The local predictor strategy was analysed with four prediction contexts, namely rhombus context, MED, GAP, and SGAP; the best results in terms of embedding capacity and image degradation were obtained with rhombus context.

The scheme proposed by Li et al. [2015] uses PEE with histogram shifting strategy and generates multiple histograms, based on the complexity levels of the pixels. For each pixel, the prediction errors are obtained and a complexity measure is calculated according to its context, based on the complexity levels multiple histograms are constructed from the prediction errors. For each histogram corresponding to a complexity level, two bins are adaptively selected for expansion where data hiding is performed. However, the drawback of this scheme is its reduced embedding capacity compared to other PEE schemes.

Reversible watermarking schemes that allow the insertion of more than 1 bpp while maintaining acceptable image degradation use strategies to determine regions of the image where more bits can be embedded while cause a smaller distortion, and embedding fewer bits in regions where modifications would cause greater distortions. A scheme such as the one proposed by Li et al. [2011] that adaptively determines pixels located at flat and rough regions and inserts the watermark bits with adaptive PEE and histogram shifting. The prediction errors that correspond to pixels in flat regions are watermarked with two bits, while prediction errors from pixels that correspond to rough regions are watermarked with one bit. A complexity measure is proposed to determine which pixels correspond to flat and rough regions. The drawback of the scheme is its computational complexity, but it achieves an embedding capacity up to 1.8 bpp with degradation above 25 dB.

The work proposed by Peng et al. [2012] presents a RWS based on difference expansion with an integer transform. The scheme adaptively calculates the amount of secret bits to be embedded in each pixel block. An estimated distortion in the integer transform is pre-calculated and by doing so, the amount of secret bits to be embedded is found. Following this adaptive embedding, more watermark bits are inserted in smooth blocks and less bits are inserted in rough blocks. In this manner, image distortion is controlled while allowing the insertion of higher payload. Because of the adaptive embedding, the computational complexity of the scheme is higher than other RWS, but payload capacity is up to 1.5 bpp with a PSNR of 28.3 dB.

Gui et al. [2014] propose a RWS based on PEE that adaptively determines the amount of secret bits to be embedded according to the complexity of the pixel. For each pixel, its complexity measure is determined based on a GAP predictor and a normalized measurement. According to the complexity of the pixel, a generalized PEE strategy finds the amount of watermark bits to be embedded. In this way, smoother pixels carry more watermark bits than rougher pixels; by doing so, the scheme allows the increment on embedding capacity while maintaining an acceptable image degradation. The scheme presents a payload capacity of 1.5 bpp with an average PSNR of 29.4 dB.

2.1.2 Fragile RWS for audio

The higher embedding capacities reported in reversible watermarking schemes for audio signals range from 1 kbps to 44.1 kbps and are summarized in Table 2.2. Nishimura [2012b] presented a scheme with an embedding capacity of 1 kbps with -1.0 of ODG; the scheme uses spatial masking along with the ambisonics technique to insert the watermarks and it can be used for stereo and mono signals; for mono ambisonics the scheme is always reversible, but the watermark is detectable; for stereo ambisonics the watermark cannot be detected, but the scheme cannot always restore the host audio signal; because the scheme uses a set of matrices to perform the embedding, no additional data is necessary to compensate the modifications caused by embedding. The scheme by Agaian et al. [2005] inserts 3.9 kbps; it applies an integer transform and insertion blocks are selected, the embedding is performed over the blocks with a least significant bit (LSB) modification; the scheme can be used directly on compressed audio signals but it has low robustness. Garcia-Hernandez [2012] reports a scheme with an embedding capacity of 16.6 kbps, it uses an interpolation-error expansion to embed the watermark, it uses a location map to solve the problem of underflow and overflow, in order to reconstruct the host signals; it proposes a multi-embedding approach to increase the embedding capacity by applying the scheme multiple times however, the multi-embedding could not be applied to all the test audio signals.

Huang et al. [2010] and Huang et al. [2011] proposed two schemes that reach high embedding capacities. The first one reaches 21.8 kbps and the second 21.7 kbps; both schemes use an integer discrete cosine transform (intDCT) and an amplitude expansion technique to embed the watermark, the schemes obtain a hash code from the audio signals to insert it into the signals in order to verify if it suffered from tampering; it does not need a location map to control the underflow and overflow, but the scheme is fragile and the perceptual impact for some audio signals is noticeable. Nishimura [2012a] presented a scheme that inserts 22.1 kbps; it uses a prediction-error expansion technique to embed the watermark, it uses more bits for the secret key that enhances the security of the scheme, however the control data increases with the number of bits for the secret key. The work by Chen et al. [2013] also embeds 22.1 kbps; they present two methods, the first uses difference expansion and the second uses prediction-error expansion, they propose a new way to select a pair of coefficients that are used to calculate the differences or prediction errors and reduces the embedding distortion. This scheme can detect content replacement and localize the positions where cropping occurred.

The scheme by Bradley and Alattar [2005] inserts 29.4 kbps, it applies a generalized reversible integer transform (GRIT) and an error expansion technique to embed the watermarks, the scheme is more suited for music than other types of audio signals, such as speech and the performance of the scheme depends on the characteristics of the audio signals. In 2003 van der Veen *et al.* proposed a scheme that reaches 40 kbps, it uses difference expansion and a shifting technique; the bits from each sample are shifted to the left and one bit from the watermark is inserted in the LSB, the theoretical embedding capacity should be 44.1 kbps but in practice only 40 kbps are reached; the imperceptibility of the watermark depends on the characteristics of the audio signals, the control for the underflow and overflow problems depend on the shifting prior embedding.

The higher embedding capacity found in reversible watermarking schemes for audio signals is of 1 bit per sample, for audio signals with CD quality this is 44.1 kbps. Nishimura [2011] presented the first to reach this capacity, it uses a prediction-error expansion technique however, the occurrence of underflow or overflow cases produce loss of data in the watermark and the efforts in this work are aimed at the reduction underflow and overflow problems. The second work to insert 44.1 kbps was presented by Huo et al. [2013], it uses a prediction-error expansion technique and their main contribution is a new non-causal predictor that not only considers samples from the past but samples from the future as well. In this scheme odd and even samples are divided into two groups, the non-causal prediction for the second group is calculated and the data is embedded, then the prediction for the first group is calculated and data is embedded into the first group; however, in order to use the non-integer prediction errors in the prediction-error expansion, they simply round these errors.

The perceptual impact of reversible schemes for audio is not presented because, as it can be seen in Table 2.2, the works report results with different metrics that cannot be compared directly. Originally, the distortion of audio signals was measured with the SNR; however, these values did not always represent the perceived quality of the signals. Later, the segmental SNR (segSNR) was proposed to try to better convey the perceived quality of the signals; the segmental SNR is the use of the SNR measure within segments of audio instead of the whole audio. Nonetheless, both SNR and segSNR rely on the statistical values of the audio signals and do not necessarily reflect the perceived quality of the audio signals. The ODG is a more recent quality metric for audio signals, it reflects better how humans perceive audio because it is based on a psychoacoustic model. We highly encourage the use of the ODG metric to report the perceptual impact that a watermarking scheme has over the watermarked signals because, as already mentioned, it reflects the perceived quality more faithfully.

Table 2.2: Most relevant reversible watermarking schemes for audio.

Scheme	Payload	SNR	SegSNR	ODG	Domain	Technique
	(kbps)	(dB)	(dB)			
Chen et al. [2013]	22.1	—	48.6	—	intDCT	Error expansion
Huo et al. [2013]	44.1	28.3	—	—	Temporal	Error expansion
Nishimura [2012b]	1.0	—	—	-1.0	Temporal	Spatial masking
Garcia-Hernandez [2012]	16.6	—	—	-2.5	Temporal	Error expansion
Nishimura [2012a]	22.1	21.1	—	-1.0	Temporal	Error expansion
Huang et al. [2011]	21.7	—	32.9	-0.2	intDCT	Error expansion
Nishimura [2011]	44.1	—	—	-1.3	Temporal	Error expansion
Huang et al. [2010]	21.8	—	—	-2.2	intDCT	Error expansion
Agaian et al. [2005]	3.9	20.9	—	—	Integer	LSB modification
Bradley and Alattar [2005]	29.4	35.0	—	—	GRIT	Error expansion
van der Veen et al. [2003]	40.0	—	—	—	Temporal	Error expansion

▨ Payload > 30 kbps

The works analyzed in this section are the ones that report higher embedding capacities or lower distortions for both image and audio signals, but we do not further explore schemes that use traditional reversible techniques, such as error expansion or histogram shifting. Notwithstanding, a comprehensive survey on reversible watermarking is presented by Khan et al. [2014], where a comprehensive classification of fragile schemes can be found and reversible watermarking schemes based on the prediction-error expansion technique are analyzed in detail.

The works reviewed so far focus on either enhancing the payload capacity or minimizing the perceptual impact of the scheme, however they leave aside the robustness. Traditional reversible watermarking schemes are inherently fragile, *i.e.* they are able to extract the watermarks and reconstruct the host signals as long as the watermarked signal does not suffer attacks. But in the presence of attacks these schemes can no longer reconstruct the host signal neither extract the watermarks, so most reversible watermarking schemes are regarded as *fragile*. Nonetheless, there are applications where apart from the reversibility constraint, it is also necessary to resist modifications.

In application scenarios where attacks do occur, fragile reversible schemes cannot be trusted to carry additional information because it will be lost and they will not be able to reconstruct the host signals either. Although the scenarios where attacks occur cannot use fragile reversible schemes, there still is the need of perfect reconstruction of the host signals after data embedding and transmission through a noisy channel. Commercial, military and medical fields require the use of high-quality signals (such as images, audio or video). Some applications in these fields require, besides the high-quality signals, the transmission of additional data that must be embedded as a secret watermark and the watermarked signals are to be transmitted through a noisy channel. After data extraction the high-quality signals should maintain their quality without loss, so the watermarking schemes should be able to compensate the distortions caused by data embedding and the possible modifications suffered in the transmission channel.

An extensive review of the possible types of robustness that current reversible watermarking schemes present is given below.

2.2 Robustness in RWS

Reversible watermarking schemes can exhibit a certain degree of robustness against attacks, *i.e.*, the capacity that these schemes posses to extract the watermarks and/or reconstruct the host signals after some modifications have occurred to the watermarked signal. This robustness property has rarely been explored and these efforts began just a decade ago. So far, robustness in reversible watermarking has been addressed from the following points of view: robustness of the watermark, robustness of the signal and robustness of both the watermark and signal.

2.2.1 Robustness of the watermark

In this robustness scenario, both the embedded watermarks and the host signal can be recovered if no attack occurred in the communication channel. When attacks do occur in this channel, only the embedded watermarks can be retrieved, and there is no guarantee that the host signal can be reconstructed. These two scenarios are depicted in Figure 2.1. Depending on the robustness that the watermarks present, these schemes can be further classified as semi-fragile or robust.

Semi-fragile watermarks

Reversible schemes with semi-fragile watermarks have a certain degree of robustness against a reduced set of attacks. The watermarks in the semi-fragile schemes resist only unintentional attacks, like slight compression; on the other hand, the watermarks in the robust schemes should be able to survive intentional attacks, like signal processing

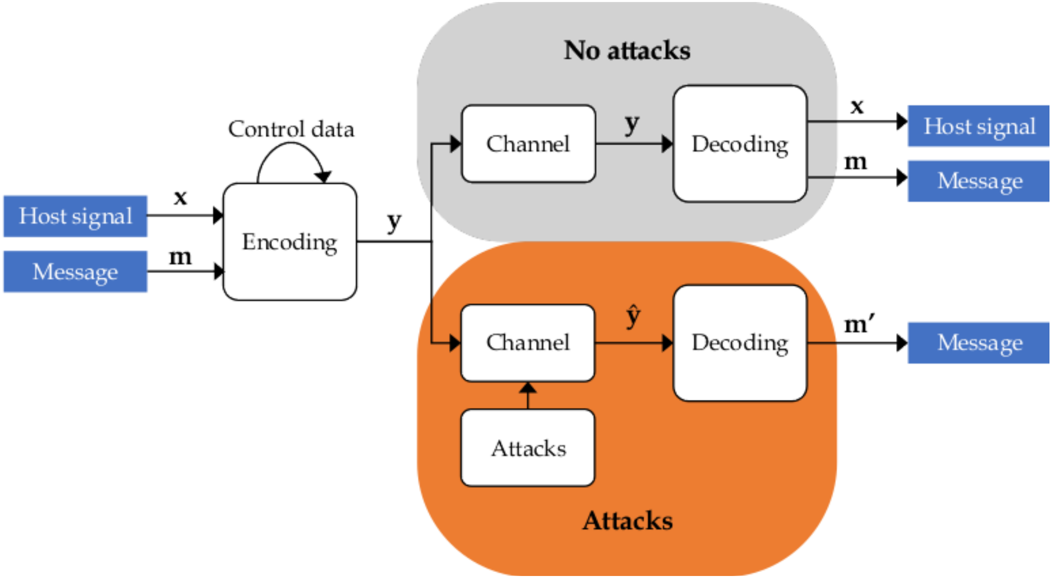


Figure 2.1: Elements in a RWS with robustness of the watermark.

operations. The former were the first efforts to preserve the embedded watermarks after some processing was applied to the watermarked signal.

The first semi-fragile reversible watermarking scheme found in the literature is the one proposed by Honsinger et al. [2001] in 2001, it can detect regions of the image that were tampered but the watermarked images suffer from salt-and-pepper noise. De Vleeschouwer et al. [2003] propose a scheme that works with images in the spatial domain and resists JPEG compression; however, the watermarked images suffer from salt-and-pepper noise. Ni et al. [2004, 2008] proposed a scheme where they solve the problem of salt-and-pepper in the previous scheme; this scheme also works with images in the spatial domain and resists JPEG and JPEG2000 compression; nonetheless, this scheme has short embedding capacity. Zou et al. [2004, 2006] presented another scheme that solves the salt-and-pepper issue; they use images in the integer wavelet transform (IWT) domain and their scheme resists JPEG2000 compression, but the embedding capacity is also little. Wu [2007] proposed a scheme that works in the IWT domain and resists JPEG compression. Finally, Kim et al. [2009] designed a scheme that utilizes images in the spatial domain and resists JPEG compression. Table 2.3 depicts a summary of these schemes.

Robust and reversible watermarking

The first robust reversible watermarking schemes appeared in 2007. Chrysochos et al. [2007] proposed a scheme that works with images in the spatial domain and is

Table 2.3: Reversible watermarking schemes with semi-fragile watermark.

Scheme	Domain	Host type	Attacks
Honsinger et al. [2001] *et al.*	Spatial	Image	—
De Vleeschouwer et al. [2003]	Spatial	Image	JPEG
Ni et al. [2004]	Spatial	Image	JPEG, JPEG2000
Zou et al. [2004]	IWT	Image	JPEG2000
Zou et al. [2006]	IWT	Image	JPEG2000
Wu [2007]	IWT	Image	JPEG
Ni et al. [2008]	Spatial	Image	JPEG, JPEG2000
Kim et al. [2009]	Spatial	Image	JPEG

robust to various attacks, such as flipping, rotation, up-sizing, increasing aspect ratio, cropping, drawing, among others; however, the embedding capacity is low. Coltuc and Chassery [2007] presented a scheme that works in the integer transform domain (ITD) and it is robust against cropping. Coatrieux et al. [2007] proposed a scheme that works with magnetic resonance images in spatial domain and is robust against JPEG compression. Gao and Gu [2007] introduced a scheme that works in the IWT domain and has robustness to cropping and salt-and-pepper noise.

Saberian et al. [2008] proposed a scheme for images and signals in the spatial and temporal domain, respectively, besides, both images and signals can be processed in the transform domain; this scheme is robust against the addition of white Gaussian noise (AWGN). Gu et al. [2009] presented a scheme in the wavelet domain that is robust against JPEG compression. Chang et al. [2009] introduced a scheme that works with images in the discrete cosine transform (DCT) domain and is robust to blurring, brightness, contrast and cropping, among others. Yang et al. [2010] proposed a method in the IWT domain that is robust against brightness, JPEG and JPEG2000 compression, cropping and inversion. Tsai et al. [2010] presented a scheme in the discrete wavelet transform (DWT) domain that is robust to JPEG compression, AWGN, salt and pepper noise, scaling and blurring, among others. Zeng et al. [2010] introduced a scheme that uses images in the spatial domain and is robust against JPEG compression. Gao et al. [2011] also presented a scheme that works with images in spatial domain and is robust to JPEG compression. Hernandez-Morales [2012] designed a scheme that works with images in the IWT domain and is robust against brightness, contrast, posterization, inversion, cropping, JPEG, and JEPG2000 compression. An et al. [2012c] and An et al. [2012b] proposed two schemes that work with images in spatial domain and both are robust against JPEG and JPEG2000 compression, and additive AWGN. An et al. [2012a] uses images in wavelet domain and is robust against JPEG and JPEG2000 compression, and additive AWGN as well. Table 2.4 presents a summary of these schemes.

Table 2.4: Reversible watermarking schemes with robust payload.

Scheme	**Domain**	**Host type**	**Attacks**
Chrysochos et al. [2007]	Spatial	Image	Flipping, rotation, translation, up-sizing, cropping, increasing aspect ratio, scattered tiles
Coltuc and Chassery [2007]	Integer	Image	Cropping
Coatrieux et al. [2007]	Spatial	Image	JPEG
Gao and Gu [2007]	IWT	Image	Cropping, salt-and-pepper noise
Saberian et al. [2008]	Spatial/ Temporal	Image/ Audio	AWGN
Gu et al. [2009]	Wavelet	Image	JPEG
Chang et al. [2009]	DCT	Image	Blurring, brightness, contrast, cropping, equalization, noise, JPEG, scaling, sharpening
Yang et al. [2010]	IWT	Image	Brightness, JPEG, JPEG2000, cropping, inversion
Tsai et al. [2010]	DWT	Image	JPEG, AWGN, salt and pepper noise, scaling, blurring, brightness, darkness, sharpen, equalization, cropping, painting
Zeng et al. [2010]	Spatial	Image	JPEG
Gao et al. [2011]	Spatial	Image	JPEG
Hernandez-Morales [2012]	IWT	Image	Brightness, contrast, posterization, inversion, cropping, JPEG, JEPG2000
An et al. [2012c]	Spatial	Image	JPEG, JPEG2000, AWGN
An et al. [2012b]	Spatial	Image	JPEG, JPEG2000, AWGN
An et al. [2012a]	Wavelet	Image	JPEG, JPEG2000, AWGN

2.2.2 Robustness of the signal

Another type of robustness is signal robustness, known in the literature as self-recovery schemes. With these schemes, the host signal can be reconstructed whether attacks occurred in the transmission channel or not. Nonetheless, embedding capacity of these schemes is significantly lower than other reversible schemes, which reduces their use in practical applications. In other words, the embedding capacity is used to insert control data and no watermarks. The control data may contain a compressed version of the signal or significant characteristics of the signal that help to the reconstruction process. These watermarking schemes can be further classified as those that obtain an approximate version of the signal and those that achieve a perfect reconstruction of the signal.

These schemes consider different kind of attacks, however in this work the content replacement attack is of particular interest because it is the attack for which perfect restoration can be achieved, and perfect restoration is required by the proposed framework. Content replacement was considered in self-recovery schemes as a form of tampering the watermarked images [Fridrich and Goljan, 1999b, Zhang and Wang, 2008, Zhang et al., 2011b, Qin et al., 2012]. Qin et al. [2012] refer to this tampering as the replacement of original content with fake information, Mobasseri and Evans [2001] define content replacement for video contents as the insertion of frames from similar looking scenes, from different portions of video or from another video altogether. Content replacement can be then understood as the replacement of regions in a signal with other information of similar size, and this regions for replacement can be taken from different sources.

Approximate reconstruction

Figure 2.2 shows the elements in a watermarking scheme with approximate signal reconstruction. This type of schemes can detect tampered regions of a signal and try to reconstruct the tampered regions using the embedded control data. Although this reconstructed signal may be similar to the original signal, it is not a lossless version of the host signal.

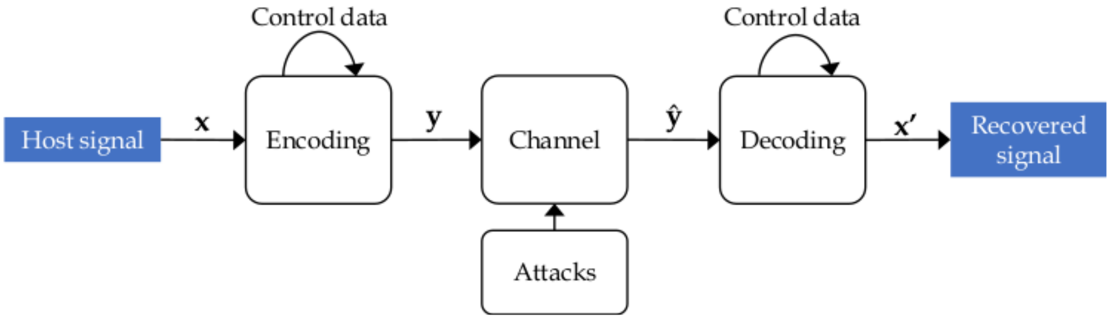


Figure 2.2: Elements in a RWS with approximate signal reconstruction.

Table 2.6 presents the watermarking schemes with approximate signal reconstruction. The first works that recovered a signal after this was subjected to attacks were proposed by Fridrich and Goljan [1999a,b]. From these works, many schemes have been proposed to date, some of them modify the domain where the embedding takes place, others improve the compression method to calculate the control data to be embedded in the signal itself. Some relevant works will be briefly described below.

As previously mentioned, the works of Fridrich and Goljan [1999a,b] were the first to be able to reconstruct an image after some malicious operation was applied to the watermarked signal. Their schemes utilized the DCT to embed in this domain the compressed version of the image into itself. Mobasseri [2000] proposed the first scheme for video signals and it was a modified version of the MPEG (Moving Picture Experts Group) compression standard. Gómez et al. [2002] proposed the first scheme for audio signals, it embedded the watermarks in the temporal domain and the scheme was only able to detect the places where the audio had been modified but could not correct these modifications. Chen et al. [2008] proposed the first scheme for audio signals that was able to correct, with a certain degree, the modifications imposed by cropping attack. Recent works like the ones proposed by Qin et al. [2012], and Korus and Dziech [2012] improve the quality of the reconstructed image after it has been subjected to attacks. But even though these are high-quality images, they are not lossless versions of the original image.

Perfect reconstruction

Figure 2.3 shows the elements in a watermarking scheme with perfect signal reconstruction. Like the schemes with approximate signal reconstruction, these can detect the regions where the signal was tampered and with the embedded control data, they can reconstruct the tampered regions. However, the reconstructed signal has exactly the same values that the ones in the host signal, *i.e.*, they achieve a perfect signal reconstruction.

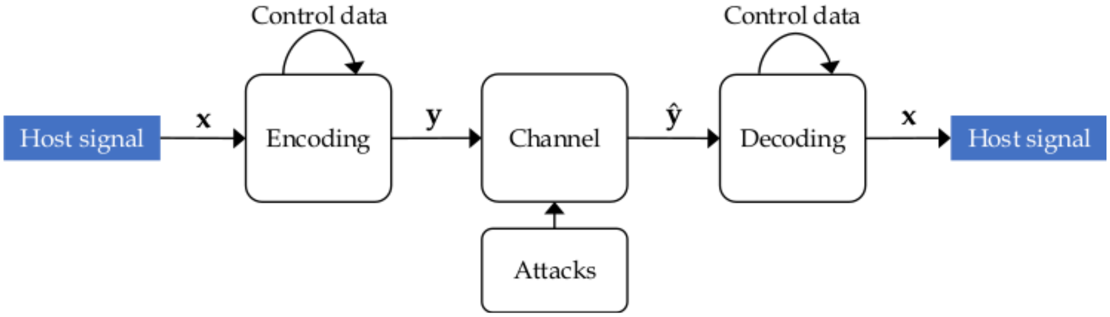


Figure 2.3: Elements in a reversible watermarking scheme with perfect signal reconstruction.

Table 2.5 presents the watermarking schemes with perfect signal reconstruction capability. All of them were proposed for image signals and work in the spatial domain. The first method was proposed by Zhang and Wang [2008]. This scheme is able to perfectly reconstruct the image after content replacement attack as long as the tampered areas are not too extensive. Zhang et al. [2011b] continued the scheme from Zhang and Wang [2008] where it can also reconstruct the image after content replacement attack as long as the tampered area is less than 24% of the image. Bravo-Solorio et al. [2012a] presented a scheme that can perfectly reconstruct an image after cropping attack, only if the tampered area is less than 25% of the image. The self-recovery scheme proposed in this research and further detailed in Section is, to the best of our knowledge, the first self-recovery scheme for audio. Moreover, it achieves perfect restoration for content replacement attacks of 0.1%.

Table 2.5: Watermarking schemes with perfect signal reconstruction.

Scheme	Domain	Host type	Tamper detection	Tamper correction	Attacks
Zhang and Wang [2008]	Spatial	Image	✓	✓	Content replacement
Zhang et al. [2011b]	Spatial	Image	✓	✓	Content replacement
Bravo-Solorio et al. [2012a]	Spatial	Image	✓	✓	Content replacement, cropping
Proposed self-recovery	intDCT	Audio	✓	✓	Content replacement

The drawback of both types of watermarking schemes is that they can reconstruct the signal only if the tampered area is not too extensive. If too many regions of the signal were corrupted, then the values of the compressed version that was originally embedded may be corrupted or lost during attacks. Because of this not all the tampered regions may be reconstructed and if there were too many tampered regions, the whole image would be lost.

In summary, non-fragile reversible watermarking schemes have proposed solutions that either restore the watermarks or the signal after the watermarked signal suffers modifications caused by attacks. Reversible watermarking schemes with watermark robustness can be semi-fragile as the works listed in Table 2.3 and more recent efforts have presented reversible schemes with robust watermarks as the works given in Table 2.4. Self-recovery schemes can have an approximate reconstruction of the signal, like the works presented in Table 2.6; but very few reach a perfect reconstruction of the signal, only three have this property and are given in Table 2.5; however perfect restoration is a solution that only exists for images. As it can be seen, reversible watermarking schemes with watermark and signal robustness have not yet been proposed to the best of our knowledge and it is an open problem in the state of the art.

The next chapter explains the strategy proposed in this doctoral research to design a reversible watermarking scheme for audio signals with watermark and signal robustness. A framework is introduced that solves this problem by using strategies from fragile reversible watermarking schemes and self-recovery schemes.

Table 2.6: Watermarking schemes with signal approximate reconstruction.

Scheme	Domain	Host type	Tamper detection	Tamper correction	Attacks
Fridrich and Goljan [1999a]	DCT	Image	✓	✓	Content replacement, random noise
Fridrich and Goljan [1999b]	DCT	Image	✓	✓	Content replacement, random noise
Mobasseri [2000]	DCT	Video	✓	✓	MPEG
Mobasseri and Evans [2001]	Spatial/ Temporal	Video	✓	✓	Frame removal, frame insertion
Wu and Chang [2002]	DCT	Image	✓	✓	Content replacement, blurring, sharpening, JPEG
Gómez et al. [2002]	Temporal	Audio	✓	×	Insertion, deletion
Celik et al. [2002]	Spatial/ Temporal	Video	✓	✓	Frame rate conversion, frame dropping, frame insertion, content replacement
Steinebach and Dittmann [2003]	Temporal	Audio	✓	×	Cropping
Caldelli et al. [2003]	DWT	Image	✓	✓	Content replacement JPEG
Lin et al. [2005]	Spatial	Image	✓	✓	—
Zhu et al. [2007]	Spatial	Image	✓	✓	Block replacement, filtering, noise, contrast modification
Zhao et al. [2007]	SLT	Image	✓	✓	Cut and paste, JPEG
Wang and Chen [2007]	Spatial	Image	✓	✓	Collage, vector quantization
Hung and Chang [2007]	Spatial	Image	✓	✓	Cropping, JPEG, content replacement
Hasan and Hassan [2007]	Spatial/ DCT	Image	✓	✓	Blind copy
Wang and Tsai [2008]	Spatial	Image	✓	✓	Content replacement

Continues on next page

Scheme	Domain	Host type	Tamper detection	Tamper correction	Attacks
Jiang and Liu [2008]	DCT	Image	✓	✓	Content replacement, adding text, block exchange, collusion, erasing
He et al. [2008]	Spatial	Image	✓	✓	Content replacement
Chen et al. [2008]	Temporal	Audio	✓	✓	Cropping
Karantonis and Ellinas [2009]	DCT	Image	✓	✓	—
He et al. [2009]	Spatial	Image	✓	✓	Content replacement, collage, constant average
Hassan et al. [2009]	Spatial/ Temporal	Video	✓	✓	Vector quantization, content replacement
Cheddad et al. [2009]	DWT	Image	✓	✓	—
Qian and Feng [2010]	DCT	Image	✓	✓	—
Qian and Qiao [2010]	DCT	Image	✓	✓	—
Mendoza-Noriega et al. [2010]	DCT	Image	✓	✓	JPEG
Korus et al. [2010]	DWT	Image	✓	✓	Blurring, JPEG
Iwata et al. [2010]	DCT	Image	✓	✓	Content replacement
Hassan et al. [2010b]	Spatial	Image	✓	✓	Vector quantization
Hassan et al. [2010a]	Spatial	Image	✓	✓	Vector quantization
Cruz et al. [2010]	DCT	Image	✓	✓	JPEG
Zhang et al. [2011a]	DCT	Image	✓	✓	Content replacement
Shi et al. [2011]	Spatial/ Temporal	Video	✓	✓	Content replacement
Mendoza-Noriega et al. [2011]	IWT	Image	✓	✓	Salt and pepper, drawing
Li et al. [2011]	DCT	Image	✓	✓	Collage, content tampering
Korus et al. [2011]	DWT	Image	✓	✓	—

Continues on next page

Scheme	Domain	Host type	Tamper detection	Tamper correction	Attacks
Qin et al. [2012]	NSCT	Image	✓	✓	Drawing
Korus and Dziech [2012]	Spatial	Image	✓	✓	—
He et al. [2012]	Spatial	Image	✓	✓	Collage, constant average
Bravo-Solorio et al. [2012b]	Spatial	Image	✓	✓	Content replacement, cropping
Ahsan et al. [2012]	DWT	Image	✓	✓	—
Shi et al. [2013]	Spatial/ Temporal	Video	✓	✓	—
Korus and Dziech [2013]	Spatial	Image	✓	✓	—

Chapter 3

Framework for reversible watermarking with watermark and signal robustness

In this chapter, the framework proposed to solve the research problem of this investigation is explored in detail. The framework uses strategies from fragile reversible watermarking schemes and self-recovery schemes. With these, the encoding and decoding processes of the framework are proposed. The first stage to validate the framework was done through an implementation for images, which is explained in this chapter, and the final implementation for audio is also detailed.

The framework proposed to construct a reversible watermarking scheme with watermark and signal robustness, presented in Figure 3.1, consists of two processes, namely encoding and decoding. Each process is further divided in two stages: a fragile stage and a self-recovery stage. In the encoding process, the fragile RWS embedding stage inserts the secret message $\mathbf{m}$ into the host signal $\mathbf{x}$, producing a watermarked signal $\mathbf{y}$, and the self-recovery stage inserts control information that allows the recovery of the signal after a content replacement attack, generating a protected signal $\mathbf{y}'$. The robustness of the whole construction is determined by the robustness of the self-recovery stage. After content replacement is performed to the watermarked signal, an attacked signal $\hat{\mathbf{y}}$ is generated. The decoding process receives this attacked signal, and first the self-recovery restoration stage is used to counteract the modifications caused by the attack. The self-recovery restoration stage extracts the control information embedded in the encoding process, and utilizes it along with the remaining non-attacked regions of the signal to restore the regions corrupted by the attack. This stage produces the same watermarked signal $\mathbf{y}$ received in the encoding process. From the watermarked signal $\mathbf{y}$, the fragile RWS extraction and recovery stage can extract the secret message and recover the original samples from the host signal.

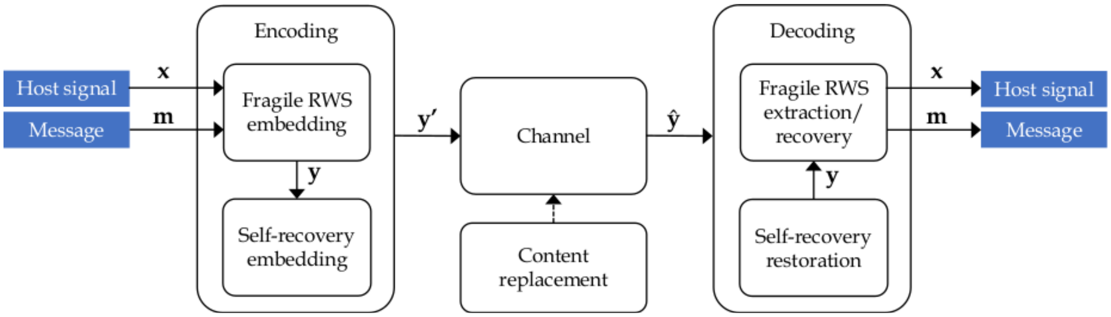


Figure 3.1: Proposed framework.

3.1 Framework for images

In general terms, the steps in a self-recovery watermarking scheme can be outlined by Algorithms 1 and 2. In the self-recovery encoding process, first the signal is divided into blocks, for the case of images; for each of those blocks, the reference and check bits are calculated. The reference bits are a compressed representation of the whole signal, and the check bits are a reduced version of hash values that uniquely represent each block. For effectiveness of the scheme, reference and check bits are pseudo-randomly permuted, to disperse these bits throughout the signal. Once the reference and check bits are permuted, these are divided into blocks for embedding; for each block in the signal, perform the embedding of the watermark. The watermark is constructed from the reference bits, and check bits of the corresponding block.

```
Input   : Watermarked signal (y), secret key (k)
Output: Protected signal (y′)
Divide y into blocks/windows
foreach block/window do
    Calculate reference bits
    Calculate check bits
end
Pseudo-randomly permute reference bits(k) of all blocks/windows
Pseudo-randomly permute check bits(k) of all blocks/windows
Divide all reference bits and check bits into blocks/windows
foreach block/window do                                  /* Embedding */
    y′(block/window) ← Insert into y(reference bits, check bits)
end
```

Algorithm 1: General steps for a self-recovery encoding process.

In the self-recovery decoding process, the attacked signal is divided in the same way as in the encoding process; then for each block, the watermark is extracted and the inverse embedding operations are applied to the attacked signal, to produce an interim version. The extracted watermark is divided into reference and check bits; then both reference and check bits are pseudo-randomly permuted in the same way as in the embedding process, to obtain both representations in the correct order. For each block, the check bits from the interim signal are calculated; these calculated check bits are to be compared against the extracted check bits to identify the tampered blocks. The extracted check bits correspond to a unique representation of the signal before attacks; when comparing check bits, tampered blocks can be detected by measuring the errors between check bits. For the tampered blocks, restoration of the original sample values can be performed through a restoration function that utilizes the reference bits and the interim signal. The sample values from non tampered blocks correspond to the sample values of the interim signal, and the remaining samples from tampered blocks can be obtained through restoration.

```
Input  : Attacked signal ($\hat{\mathbf{y}}$), secret key (k)
Output: Watermarked signal ($\mathbf{y}$)
Divide $\hat{\mathbf{y}}$ into blocks/windows
foreach block/window do
    Extract watermark
    $\mathbf{y}_{\text{interim}}$ ← Inverse embedding operations on samples
end
Divide watermark into reference bits and check bits
Pseudo-randomly permute reference bits(k) of all blocks/windows
Pseudo-randomly permute check bits(k) of all blocks/windows
foreach block/window do
    Calculate check bits from $\mathbf{y}_{\text{interim}}$
end
Identify tampered blocks/windows(extracted check bits, calculated check bits)
$\mathbf{y}$ ← $\mathbf{y}_{\text{interim}}$
foreach tampered block/window do
    $\mathbf{y}$(tampered block/window) ← Restore original samples(tampered
      block/window, reference bits, $\mathbf{y}_{\text{interim}}$)
end
```

Algorithm 2: General steps for a self-recovery decoding process.

The goal of this investigation is to propose a reversible watermarking scheme with watermark and signal robustness for audio signals. Nonetheless, the framework proposed to address the research problem is an abstract construction that can be implemented for different type of media. In order to validate the effectiveness of the framework, *i.e.*, to verify that it is a solution to the stated problem, it was first implemented for images. In the watermarking research line, most of the advances in the state of the art are proposed in schemes for imagery, and then the general ideas from these schemes are adapted and modified for another type of media, such as audio. Since the proposed framework consists on two stages, namely fragile RWS, and self-recovery; and given that these type of schemes in the literature existed only for images, the straightforward strategy to validate the framework, was to implement it using the existing fragile RWS and self-recovery schemes for images from the literature.

The implementation of the framework for images is the following. The fragile RWS embeds a watermark and control data to reconstruct the host image, the self-recovery scheme embeds information to counteract the modifications caused by the content replacement attack. The scheme proposed by [Sachnev et al., 2010] was selected as the fragile reversible one because it remains as one of the most efficient reversible watermarking schemes in terms of payload capacity which is greater than 1 bpp with acceptable degradation, although the computational complexity increases when embedding bit-rate is increased as well. The scheme by [Zhang and Wang, 2008] was selected as the self-recovery one because it can perfectly reconstruct a host image after attacks.

3.1.1 Encoding

The encoding process inserts a watermark into a host image in the fragile RWS stage, producing a first-stage watermarked image. The self-recovery stage inserts into the first-stage watermarked image, information to counteract modifications from the attack, producing a second-stage watermarked image, the global distortion of this process must meet a degradation constraint $\phi = 30$ dB.

Fragile RWS stage. In this stage, the embedding algorithm described in [Sachnev et al., 2010] is used to embed the watermark into the host image. This algorithm constructs the watermark to be embedded by concatenating the bits of the watermark with the least significant bits (LSB) from the first 30 pixels in the image. The watermark is hidden using the histogram shifting algorithm detailed in [Sachnev et al., 2010]. A header is constructed with parameters necessary for the decoder (T_p, T_n, i) and their binary representation is embedded into the LSB of the first 30 pixels. The embedding algorithm produces a location map that contains information about the overflow and underflow cases, that location map is necessary to restore the host image.

Self-recovery stage. This stage embeds data into the image itself to detect the regions where tampering took place (check bits) and data to restore the original values of the pixels from the tampered areas (reference bits). This stage uses the watermark embedding procedure described in [Zhang and Wang, 2008], where the reference bits are a compressed version of the binary image representation and the check bits for tamper-detection are the result of hashing every block in the image. Both the reference bits and the check bits are pseudo-randomly permuted prior embedding in order to be dispersed through the image during the embedding procedure, increasing in this way the robustness.

Attacks. Because of the characteristics from the framework, it inherits the robustness of the work by Zhang and Wang. This scheme only resists the content replacement attack when the tampered area is less than 3.2% of the image and so does the proposed solution. The content replacement attack consists on replacing some pixels from the watermarked image with the same amount of pixels from another image.

3.1.2 Decoding

In order to perfectly reconstruct the host image and to extract the hidden watermark, the self-recovery stage must be carried out first, followed by the fragile RWS stage. The self-recovery stage enables the detection of the tampered areas of the image and provides data to compensate the distortions caused by the attack and the fragile RWS stage, so the first-stage watermarked image is recovered. From the first-stage watermarked image, the fragile reversible stage is capable of extracting the hidden watermark and restoring the host image.

Self-recovery stage. This stage uses the image restoration procedure described in [Zhang and Wang, 2008], where reference bits and check bits are extracted from the image itself. A set of calculated check bits is obtained from the values of the attacked image, in the same fashion as the extracted check bits were calculated. To identify the tampered areas, extracted check bits are compared against calculated check bits. If the differences between them exceed a threshold, the block is considered tampered. Pixel values and reference bits extracted from tampered blocks are not reliable, so these missing bits are calculated from non-tampered blocks, using reference bits and the pixels' binary representation.

Fragile reversible stage. This stage employs the decoder algorithm given in [Sachnev et al., 2010]. It takes the first-stage watermarked image and extracts the first 30 pixel's LSB to construct parameters for data extraction (T_p, T_n, i). With these parameters along with the location map, the histogram shifting decoding procedure extracts the hidden watermark and reconstructs the original image pixel values. The 30 bits that were appended to the watermark and hidden in the embedding process are utilized to restore the first 30 pixel values of the host image.

3.2 Framework for audio

In the same way as the implementation of the framework for images consists of two stages, the implementation for audio is comprised of a fragile RWS stage, and a self-recovery stage. A fragile RWS and a self-recovery scheme were both proposed in this work, in order to implement the framework. The proposed fragile RWS uses the psychoacoustic characteristics of the human auditory system (HAS) to select frequencies in the Fourier domain, where the watermark can be inserted; these frequencies are then mapped to the intDCT domain, where embedding is carried out. Since reversibility is required, the transform domain where embedding takes place must be integer, therefore the intDCT domain was selected. The embedding of the watermark is performed through a modified prediction error expansion technique with multi-bit embedding. Prediction error expansion calculates the error between original intDCT values and predicted ones. By expanding the error, multiple bits from the watermark can be inserted into each intDCT coefficient. A detailed explanation of the proposed fragile RWS is given in Chapter 4.

A self-recovery watermarking scheme for audio signals with perfect restoration capabilities was proposed, in order to be used as part of the framework. This scheme broadly performs the steps for encoding and decoding described in Algorithms 1 and 2. The reference bits are calculated from the time domain representation of the audio signals. Each window of samples is converted to the intDCT domain, from which the check bits are calculated, and posterior embedding of the watermark is performed in this domain as well. Reference bits and check bits are pseudo-randomly permuted based on the secret key, in order to disperse these bits throughout the signal; from the permuted bits, the watermark is constructed and divided into windows. For each window, frequencies in the Fourier domain are selected based on the psychoacoustic characteristics of the window, and then mapped to the intDCT domain, in a similar fashion as in the proposed fragile RWS. Once the intDCT frequencies are selected, the watermark bits are embedded using a prediction error expansion technique with multi-bit embedding. The inverse intDCT transform is applied to the watermarked coefficients, to produce the watermarked audio signal in time domain. A detailed explanation of the proposed self-recovery scheme is given in Chapter 5.

The implementation of the framework for audio signals is divided in two stages: encoding and decoding.

3.2.1 Encoding

The encoding process of this implementation is outlined by Algorithm 3. This process first applies a fragile RWS stage, that receives a host audio signal and divides it into windows; for each window the spectrum, masking threshold, and intDCT transform are calculated, the Fourier frequencies in the spectrum that fall under the masking threshold are selected as candidate for embedding, then these candidate frequencies are mapped to the intDCT domain where multi-bit embedding takes place. The time domain representation of the first-stage watermarked audio signal is obtained by applying the inverse intDCT transform to the watermarked intDCT coefficients. The first-stage watermarked audio signal is then passed to the self-recovery stage, where it is divided into windows, which do not necessarily are the same size as the windows in the fragile RWS stage. For each window, the reference bits and intDCT coefficients are calculated, from the intDCT representation, the check bits of each window are calculated. The reference and check bits are pseudo-randomly permuted and further divided into windows, in a similar fashion as in the self-recovery scheme for images. For each window, a process similar to the fragile RWS one is carried out for embedding. Candidate frequencies are selected in the Fourier domain based on the masking threshold of each window, and then they are mapped to the intDCT frequencies; embedding takes place through a multi-bit strategy. The protected audio signal is obtained by applying the inverse intDCT transform to the second-stage watermarked intDCT coefficients.

A content replacement attack in audio signals for application scenarios like the ones mentioned in Chapter 1, would carefully select samples that correspond to words in the audio signal and replace them with other words, silences, single tones or sound effects. However, to evaluate a big set of watermarked audio signals, this process has to be automated. For this reason, the content replacement attack had to be simulated for this experimental set-up. The simulation of a content replacement is performed following Algorithm 4, where |.| indicates the size of a signal, randi(.), min(.), and max(.) are functions that generate random integer numbers, and obtain the minimum and maximum values within a signal, respectively.

```
Input   : Host audio ($\mathbf{x}$), watermark ($\mathbf{w}$), secret key ($k$)
Output: Protected audio ($\mathbf{y}'$)
                                                    /* Fragile RWS stage */
Divide $\mathbf{x}$ into windows
foreach $window_f$ do
    Calculate spectrum
    Calculate masking threshold
    Identify candidate Fourier frequencies
    $\mathbf{X} \leftarrow \mathrm{intDCT}(\mathbf{x})$
    Map Fourier frequencies to intDCT
    $\mathbf{X}'(\mathrm{window_f}) \leftarrow$ Multi-bit embedding into $\mathbf{X}(\mathbf{w})$
end
$\mathbf{y} \leftarrow \mathrm{invIntDCT}(\mathbf{X}')$
                                                    /* Self-recovery stage */
Divide $\mathbf{y}$ into $\mathrm{window_s}$
foreach $window_s$ do
    Calculate reference bits
    $\mathbf{Y} \leftarrow \mathrm{intDCT}(\mathbf{y})$
    Calculate check bits from $\mathbf{Y}$
end
Pseudo-randomly permute reference bits(k) of all $\mathrm{window_s}$
Pseudo-randomly permute check bits(k) of all $\mathrm{window_s}$
Divide reference bits and check bits into windows
foreach $window_s$ do
    Calculate spectrum
    Calculate masking threshold
    Identify candidate Fourier frequencies
    $\mathbf{Y} \leftarrow \mathrm{intDCT}(\mathbf{y})$
    Map Fourier frequencies to intDCT
    $\mathbf{Y}'(\mathrm{window_s}) \leftarrow$ Multi-bit embedding into $\mathbf{Y}$(reference bits, check bits)
end
$\mathbf{y}' \leftarrow \mathrm{invIntDCT}(\mathbf{Y}')$
```

Algorithm 3: Encoding process from framework for audio signals.

3.2.2 Decoding

The decoding process is outlined by Algorithm 5. For the decoding, first the self-recovery stage is applied, to restore the modifications caused by the content replace-

```
Input   : Watermarked audio ($\mathbf{y}'$), frame size ($\text{fr}_s$), percentage of attack ($\%_{\text{attack}}$)
Output: Attacked audio ($\hat{\mathbf{y}}$)
$\text{num}_{\text{samps}} \leftarrow \lfloor \text{fr}_s \times (\%_{\text{attack}}/100) \rfloor$
$\text{max}_{\text{pos}} \leftarrow |\mathbf{y}'| - \text{num}_{\text{samps}}$
$\text{rand}_{\text{pos}} \leftarrow \text{randi}([1, \text{max}_{\text{pos}}], 1, 1)$
$\hat{\mathbf{y}} \leftarrow \mathbf{y}'$
$\hat{\mathbf{y}}(\text{rand}_{\text{pos}} : \text{rand}_{\text{pos}} + \text{num}_{\text{samps}} - 1) \leftarrow \text{randi}([\min(\mathbf{y}'), \max(\mathbf{y}')], 1, \text{num}_{\text{samps}})$
```

Algorithm 4: Simulation of a content replacement attack.

ment attack, and then the fragile RWS decoding stage is carried out, in order to recover the host audio signal and to extract the watermark. The self-recovery stage receives an attacked audio signal, and it is divided into windows of the same size as in the self-recovery stage in the encoding process. Candidate frequencies in the Fourier domain are selected under the same criteria as in the encoding process, considering the masking threshold of the window. These Fourier frequencies are mapped to the intDCT frequencies, in order to extract the control bits from these coefficients. The operations carried out in the embedding process in this stage are inversed to recover the sample values, creating an interim version of the audio signal. The extracted control bits are divided into reference and check bits, and pseudo-randomly permuted based on the same secret key as in the encoding. For each window of samples, the check bits from the interim audio signal are obtained, to compare them against the extracted check bits. This comparison allows the identification of tampered windows. For each tampered window, the sample values before attacks can be obtained through a restoration process, that utilizes the extracted reference bits and the time domain representation of the interim audio signal, restoration produces the first-stage watermarked audio signal. From it, the fragile RWS decoding stage can be applied to complete the final stage of the framework. This signal is divided into windows, in the same way as in the encoding process. Again, the candidate Fourier frequencies are selected under the same criteria, and then are mapped to the intDCT domain, where extraction of the watermark takes place. The inverse embedding operations are applied to the intDCT coefficients, to recover the host intDCT values. The final host signal is obtained by applying the inverse intDCT transform to the recovered coefficients.

The next chapter explains the strategies used to propose a reversible watermarking scheme for audio signals that exploits the auditory masking properties of the audio signals in order to determine the best frequencies for watermark insertion. The use of the intDCT domain is proposed to carry the watermarks since this domain allows the adaptive selection of frequencies that reduced the perceptual impact of modifications, and since it is an integer domain a reversible watermarking scheme can be used for embedding and extraction.

```
Input  : Attacked audio ($\hat{\mathbf{y}}$), secret key (k)
Output : Host audio ($\mathbf{x}$), watermark ($\mathbf{w}$)
Divide $\hat{\mathbf{y}}$ into windows$_s$                    /* Self-recovery stage */
foreach windows$_s$ do
    Calculate spectrum
    Calculate masking threshold
    Identify candidate Fourier frequencies
    $\hat{\mathbf{Y}} \leftarrow \text{intDCT}(\hat{\mathbf{y}})$
    Map Fourier frequencies to intDCT
    Extract control bits from $\hat{\mathbf{Y}}$
    $\mathbf{Y}_{\text{interim}} \leftarrow$ Inverse embedding self-recovery operations in $\hat{\mathbf{Y}}$
end
Divide control bits into reference bits and check bits
Pseudo-randomly permute reference bits(k) of all windows$_s$
Pseudo-randomly permute check bits(k) of all windows$_s$
foreach windows$_s$ do
    Calculate check bits from $\mathbf{Y}_{\text{interim}}$
end
Identify tampered windows(extracted check bits, calculated check bits)
$\mathbf{y}_{\text{interim}} \leftarrow \text{invIntDCT}(\mathbf{Y}_{\text{interim}})$
$\mathbf{y} \leftarrow \mathbf{y}_{\text{interim}}$
foreach tampered window do
    $\mathbf{y}$(tampered window) $\leftarrow$ Restore original samples(tampered window,
      reference bits, $\mathbf{y}_{\text{interim}}$)
end
Divide $\mathbf{y}$ into windows$_f$                          /* Fragile RWS stage */
foreach window$_f$ do
    Calculate spectrum
    Calculate masking threshold
    Identify candidate Fourier frequencies
    $\mathbf{Y} \leftarrow \text{intDCT}(\mathbf{y})$
    Map Fourier frequencies to intDCT
    $\mathbf{w}(\text{window}_f) \leftarrow$ Multi-bit extraction from $\mathbf{Y}$
    $\mathbf{X}(\text{window}_f) \leftarrow$ Inverse RWS embedding operations in $\mathbf{Y}$
end
$\mathbf{x} \leftarrow \text{invIntDCT}(\mathbf{X})$
```

Algorithm 5: Decoding process from framework for audio signals.

Chapter 4

Proposed fragile RWS for audio

In this chapter, the strategies used to propose a reversible watermarking scheme for audio signals are detailed. The necessity to propose such a scheme is given, since there are several reversible watermarking schemes for audio signals that already exist in the literature. In this scheme, the intDCT domain is proposed to be the domain where the watermark is inserted, since it allows the selection of frequencies that maintain the transparency of the scheme given the payload capacities required, and because it is an integer domain, so reversibility can be achieved. The auditory masking properties of audio signals are exploited in order to select the frequencies that better mask the noise caused by embedding, and a mapping strategy is proposed to find the intDCT frequencies that correspond to the frequencies selected in the Fourier domain based on the masking threshold. Finally, a modification to the classical prediction error expansion (PEE) technique is used to increase the embedding capacity of the scheme.

As shown in Chapter 2, there are fragile RWS for audio in the literature with different properties. Existing schemes for audio signals that achieve the highest embedding capacities work with the time domain representation of the signals, but their transparency in terms of ODG are not adequate for practical applications. From Chapter 3, it can be appreciated that a fragile reversible scheme is required for the first stage of the proposed framework, but also as a part of the proposed self-recovery stage. Since the self-recovery scheme requires the property of perfect restoration, the amount of data that should be embedded into the signals has to be high, but maintaining an adequate transparency. Because of this trade-off, a fragile reversible watermarking scheme for audio signals had to be designed, in order to propose a scheme that can select regions of the audio signals where watermarks can be embedded at high embedding capacities, while preserving adequate transparency.

The selection of the embedding regions is carried out in the frequency domain, where frequency components can be adaptively selected, to modify only those which are imperceptible to human listeners. Since reversibility is required, an integer frequency domain is required; therefore, the use of the intDCT domain.

4.1 intDCT transform

The forward DCT-IV transform of an N-point audio signal $\mathbf{x}[n]$ is given by Eq. (4.1), and its inverse transform is given by Eq. (4.3) as follows:

$$\mathbf{X}[m] = C_N^{IV} \cdot \mathbf{x}[n], \qquad m = n = 0, 1, \cdots, N-1 \tag{4.1}$$

where $\mathbf{X}$ represents the DCT coefficients of $\mathbf{x}$. C_N^{IV} is the transform matrix, defined as:

$$C_N^{IV} = \sqrt{\frac{2}{N}} \left[\cos\left(\frac{(m+\frac{1}{2})(n+\frac{1}{2})\pi}{N} \right) \right], \tag{4.2}$$

where $m = 0, 1, \cdots, N-1$ and $n = 0, 1, \cdots, N-1$. Because C_N^{IV} is an orthogonal matrix, the inverse DCT transform is given by:

$$\mathbf{x}[n] = C_N^{IV} \cdot \mathbf{X}[m]. \tag{4.3}$$

The intDCT is an approximation of the DCT-IV, which is calculated using the fast intMDCT algorithm proposed by [Huang et al., 2006]. This implementation divides the transform matrix into five sub-matrices, the multiplication by each of the five matrices is done through a lifting stage with a rounding operation, which produces integer results. Through the five lifting stages the intDCT coefficients are obtained. A detailed explanation on the implementation of the intDCT is presented in Appendix C.

4.2 Masking threshold

The use of the intDCT domain is proposed to exploit the selection of frequencies that better mask the noise produced by the insertion of the watermark. This selection is based on the auditory masking in each segment of the signal. Auditory masking, exemplified in Fig. 4.1, occurs when one faint but audible sound (Masked sounds) is made inaudible in the presence of a louder audible sound (Masker) [Lin and Abdulla, 2015]. To determine which frequencies are masked by a predominant frequency, the masking threshold has to be obtained. The masking threshold indicates the frequency components that are unnoticeable for a human listener because of the existence of a predominant frequency. The predominant frequency 'masks' other frequencies near it, therefore, insertion of a watermark can be done in the masked frequencies without notorious differences for the human listener. The masking threshold is calculated from the Fourier spectra of the signal; all the frequencies in the Fourier spectra that fall under the masking threshold are candidates for embedding.

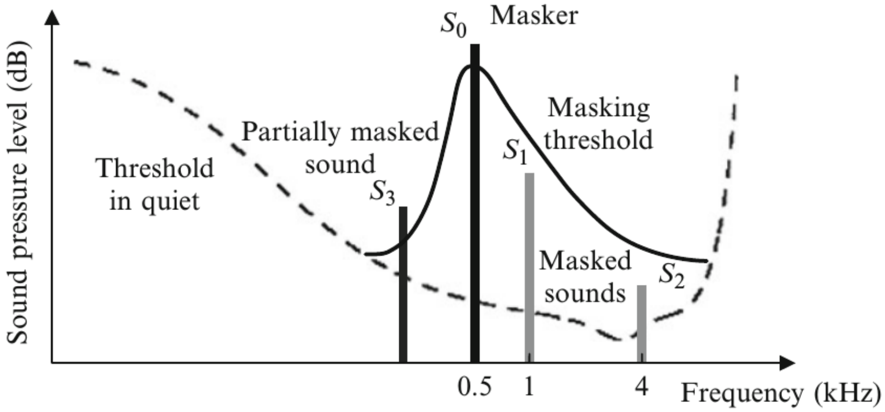


Figure 4.1: Masking threshold in auditory masking [Lin and Abdulla, 2015].

4.3 Frequency mapping

The masking threshold of an audio segment is calculated based on the Fourier spectrum of the segment, and the candidate frequencies selected are also in the Fourier spectrum. However, in order to propose a reversible watermarking scheme, the frequency representation needs to be integer, hence the intDCT domain. Although both domains give a representation of the frequency components that exist in the audio segment, frequencies in the Fourier domain do not directly correspond to frequencies in the intDCT domain; therefore, it is necessary to make a mapping from Fourier to intDCT.

The FFT spectrum of an N-point signal has $N/2$ frequency components, each corresponding to basis functions that linearly increase in frequency. The intDCT of the same signal yields N transform coefficients that correspond to cosine basis functions that also linearly increase in frequency; but, unlike the FFT basis functions, the number of periods in each basis function increases in steps of $1/2$ [Owen, 2007]. This implicates that, if the frequency f_i is at the i-th coefficient in the FFT spectrum, then f_i corresponds to the $2i$-th coefficient in the intDCT domain.

Suppose a watermark of length K is to be embedded into an audio segment. The masking threshold of the segment is calculated, and K FFT frequencies are selected as candidate ones, these candidate frequencies are at indexes $\{i_1, i_2, \cdots, i_K\}$, and the corresponding intDCT frequencies are at indexes $\{2i_1, 2i_2, \cdots, 2i_K\}$. For natural audio signals, it is expected that the highest frequencies fall under the masking threshold for most of the audio segments. Once the FFT candidate frequencies are selected, they are mapped to the intDCT domain; it is in the later domain where embedding takes place. Because of the mapping from FFT to intDCT, the intDCT coefficients are located at even indexes.

For example, suppose the candidate frequencies in the FFT spectrum are at indexes $\{\cdots, 253, 254, 255, 256\}$. When mapped to the intDCT domain, the indexes of these frequencies are $\{\cdots, 506, 508, 510, 512\}$. Indexes $\{507, 509, 511\}$ in the intDCT domain correspond to frequencies that are not represented in the Fourier domain, since they have periods that increase in $1/2$ a step, whereas in Fourier the increases of period are in steps of 1. Therefore, frequencies at odd indexes are not mapped, because they do not directly correspond to FFT frequencies. However, if the candidate FFT frequencies are located at contiguous indexes, it is expected that frequencies $1/2$ period away would also remain under the masking threshold, *i.e.*, it is assumed that if two adjacent FFT frequencies are selected and mapped to intDCT, then the frequency in between of the other two frequencies is also a frequency masked by the masker sound.

The selection of fewer frequencies for modification reduces the perceptual impact when embedding the watermark, yielding in better transparency; however fewer bits are inserted. To increase the embedding capacity, more frequencies have to be selected for insertion. Because it is assumed that intDCT frequencies at odd indexes, between frequencies that do correspond to FFT frequencies, are under the masking threshold as well, the final coefficients selected in intDCT domain can be doubled to those selected in FFT domain. For example, if a watermark of length K is to be embedded, in the FFT spectrum $K/2$ frequencies below the masking threshold are selected as candidates, and their corresponding intDCT frequencies at even indexes are mapped. From the selected intDCT frequencies at even indexes, the odd indexes between them can be included in the selection since it is assumed that these annexed frequencies will not have a repercussion in the transparency. In summary, $K/2$ intDCT frequencies mapped to even indexes are selected, and $K/2$ odd indexes are annexed to this selection, K bits are then inserted to the selected intDCT coefficients.

4.4 PEE with multi-bit expansion

Prediction error expansion (PEE) is a reversible watermarking technique, originally proposed for images [Thodi and Rodriguez, 2004] but also adapted for audio signals [Yan and Wang, 2008, Chen et al., 2013, Huo et al., 2013, Nishimura, 2011, 2012a].

The general idea of this strategy is that watermark bits can be embedded in the least significant bit (LSB) of the error calculated between an original sample and its predicted value. To avoid loss of data, instead of the direct substitution of the LSB used by classical watermarking schemes, PEE expands the error and then inserts the corresponding bit in the LSB.

4.4.1 Proposed embedding

The embedding algorithm proposed is outlined in Algorithm 6. An audio signal $\mathbf{x}[n], \{n = 1, 2, \cdots, N\}$ is received, where N is its length. It is then divided into non overlapping windows of size L_r, and the number of windows in the complete signal are $n_{win} = \lfloor N/L_r \rfloor$. The watermark to be embedded is denoted as $\mathbf{w}[k]$, where $k = \{1, 2, \cdots, K\}$, and K is the size of the watermark. The number of bits to be embedded in each window is $\lfloor K/n_{win} \rfloor$. The insertion of the watermark is done through PEE in the intDCT domain, in a similar fashion as in the scheme by [Chen et al., 2013]. It is assumed that coefficients in odd indexes are more similar to other coefficients at odd indexes, and coefficients at even indexes are more similar to other coefficients at even indexes, as highlighted in [Chen et al., 2013].

This assumption holds for the following reasons, the intDCT coefficients are obtained multiplying a square matrix by a column vector that corresponds to the audio signal. From the transform matrix C_N^{IV} obtained with eq. 4.2, it can be observed that the absolute sum of positive values at odd rows is greater than the absolute sum of negatives, and the absolute sum of negative values at even rows is greater than that of positives. The audio signals processed with the scheme are positive integer-valued ones, and it is also assumed that for most natural audio signals, the samples in segments of size L_r have very similar values. The multiplication of the transform matrix C_N^{IV} by the integer-valued audio segments, result in positive intDCT coefficients at odd indexes and negative coefficients at even indexes. This is true for most of the segments in all the audio signals tested; therefore, the embedding strategy is based on this assumption. The prediction value of the i^{th} coefficient, denoted by $\hat{\mathbf{X}}[i]$ is calculated as:

$$\hat{\mathbf{X}}[i] = \left\lfloor \frac{\mathbf{X}[i-2] + \mathbf{X}[i-4]}{2} \right\rfloor, \tag{4.4}$$

and the prediction-error, denoted as p, is given by:

$$p = \mathbf{X}[i] - \hat{\mathbf{X}}[i], \tag{4.5}$$

where i represents the indexes of the mapped intDCT frequencies. In the classical PEE strategy, only one bit per sample is inserted; however, the proposed algorithm uses

[Coltuc and Tudoroiu, 2012]'s idea, where multiple bits can be inserted in one sample. The proposed modification used for embedding inserts two bits per frequency, and the prediction-error p is expanded as follows:

$$p_w = 4 \times p + (2 \times \mathbf{w}[k]) + \mathbf{w}[k+1]. \tag{4.6}$$

The watermarked intDCT coefficients are obtained by:

$$\mathbf{Y}[i] = \hat{\mathbf{X}}[i] + p_w. \tag{4.7}$$

The watermarked signal in its time domain representation is obtained by applying the inverse intDCT transform to **Y**.

Proposed extraction

The extraction algorithm proposed is outlined in Algorithm 7. A received watermarked audio signal $\mathbf{y}[n], \{n = 1, 2, \cdots, N\}$ is divided into n_{win} windows, in the same way as in the embedding algorithm. The intDCT coefficients of each window are selected using the same criteria. The masking threshold of each segment is obtained, the frequencies in the Fourier spectrum are mapped to the frequencies in the intDCT domain, and PEE extraction is applied in the following way. The prediction value $\hat{\mathbf{Y}}[i]$ is calculated as:

$$\hat{\mathbf{Y}}[i] = \left\lfloor \frac{\mathbf{Y}[i-2] + \mathbf{Y}[i-4]}{2} \right\rfloor, \tag{4.8}$$

and the expanded prediction-error is given by:

$$p_w = \mathbf{Y}[i] - \hat{\mathbf{Y}}[i], \tag{4.9}$$

where i represents the indexes of the frequencies in the intDCT domain. The original prediction-error p is obtained by:

$$p = \left\lfloor \frac{p_w}{4} \right\rfloor, \tag{4.10}$$

and the watermark word, w_o, that contains 2 bits is extracted as:

$$\begin{aligned} w_o &= p_w - 4 \times p, \\ \mathbf{w}[k] &= \mathrm{mod}(w_o, 2), \\ \mathbf{w}[k+1] &= \mathrm{mod}(\lfloor w_o/2 \rfloor, 2). \end{aligned} \tag{4.11}$$

The original intDCT coefficients are restored by:

$$\mathbf{X}[i] = \hat{\mathbf{Y}}[i] + p. \tag{4.12}$$

```
Input  : Host audio ($\mathbf{x}$), watermark ($\mathbf{w}$), window size ($L_r$)
Output: Watermarked audio ($\mathbf{y}$)
$N \leftarrow |\mathbf{x}|$
$K \leftarrow |\mathbf{w}|$
$n_{win} \leftarrow \lfloor N/L_r \rfloor$
$k_{win} \leftarrow \lfloor K/n_{win} \rfloor$
for $n = 1 : n_{win}$ do
    $\mathbf{w}_{win} \leftarrow \mathbf{w}((n-1) \times k_{win} + 1 : n \times k_{win})$
    $FFT_{spectrum} \leftarrow$ FFT spectrum($\mathbf{x}((n-1) \times n_{win} + 1 : n \times n_{win})$)
    $masking_{threshold} \leftarrow$ Masking threshold($\mathbf{x}((n-1) \times n_{win} + 1 : n \times n_{win})$)
    $\mathbf{X} \leftarrow intDCT(\mathbf{x}((n-1) \times n_{win} + 1 : n \times n_{win}))$
    $candidate_{freqs} \leftarrow find(FFT_{spectrum} < masking_{threshold})$
    $selected_{freqs} \leftarrow candidate_{freqs}(|candidate_{freqs}| : -1 : |candidate_{freqs}| - (k_{win}/4))$
    $\mathbf{Y} \leftarrow \mathbf{X}$
    $j \leftarrow 1$
    for $k = 1 : 4 : k_{win}$ do
        $fft_{index} \leftarrow selected_{freqs}(j)$
        $i \leftarrow 2 \times fft_{index}$                    /* Mapping Fourier to intDCT */
        $\hat{\mathbf{X}}(i) \leftarrow \lfloor (\mathbf{X}(i-2) + \mathbf{X}(i-4))/2 \rfloor$
        $p \leftarrow \mathbf{X}(i) - \hat{\mathbf{X}}(i)$
        $p_w \leftarrow 4 \times p + (2 \times \mathbf{w}_{win}(k)) + \mathbf{w}_{win}(k+1)$
        $\mathbf{Y}(i) \leftarrow \hat{\mathbf{X}}(i) + p_w$                    /* Embedding at even index */
        $\hat{\mathbf{X}}(i-1) \leftarrow \lfloor (\mathbf{X}(i-3) + \mathbf{X}(i-5))/2 \rfloor$
        $p \leftarrow \mathbf{X}(i-1) - \hat{\mathbf{X}}(i-1)$
        $p_w \leftarrow 4 \times p + (2 \times \mathbf{w}_{win}(k+2)) + \mathbf{w}_{win}(k+3)$
        $\mathbf{Y}(i-1) \leftarrow \hat{\mathbf{X}}(i-1) + p_w$                    /* Embedding at odd index */
        $j \leftarrow j + 1$
    end
    $\mathbf{y}_{win} \leftarrow invIntDCT(\mathbf{Y})$
    $\mathbf{y}((n-1) \times n_{win} + 1 : n \times n_{win}) \leftarrow \mathbf{y}_{win}$
end
```

Algorithm 6: Embedding algorithm for the proposed RWS.

The original sample values in the time domain are obtained by applying the inverse intDCT transform to the restored intDCT coefficients.

The next chapter presents the proposed self-recovery scheme for audio signals.

The general idea of self-recovery schemes is explained, and the role that the fragile reversible watermarking scheme explained in this chapter has on the self-recovery scheme is also stated. The steps followed by the encoding and decoding processes are explained, examples are given as well as the algorithms.

```
Input  : Watermarked audio ($\mathbf{y}$), watermark size ($K$), window size ($L_r$)
Output: Host audio ($\mathbf{x}$),watermark ($\mathbf{w}$)
$N \leftarrow |\mathbf{x}|$
$n_{\text{win}} \leftarrow \lfloor N/L_r \rfloor$
$k_{\text{win}} \leftarrow \lfloor K/n_{\text{win}} \rfloor$
for $n = 1 : n_{win}$ do
    $\mathbf{y}_{\text{win}} \leftarrow \mathbf{y}((n-1) \times n_{\text{win}} + 1 : n \times n_{\text{win}})$ $\text{FFT}_{\text{spectrum}} \leftarrow \text{FFT spectrum}(\mathbf{y}_{\text{win}})$
    $\text{masking}_{\text{threshold}} \leftarrow \text{Masking threshold}(\mathbf{y}_{\text{win}})$
    $\mathbf{Y} \leftarrow \text{intDCT}(\mathbf{y}_{\text{win}})$
    $\text{candidate}_{\text{freqs}} \leftarrow \text{find}(\text{FFT}_{\text{spectrum}} < \text{masking}_{\text{threshold}})$
    $\text{selected}_{\text{freqs}} \leftarrow \text{candidate}_{\text{freqs}}(|\text{candidate}_{\text{freqs}}| : -1 : |\text{candidate}_{\text{freqs}}| - (k_{\text{win}}/4))$
    $\mathbf{X} \leftarrow \mathbf{Y}$
    $j \leftarrow 1$
    for $k = 1 : 4 : k_{win}$ do
        $\text{fft}_{\text{index}} \leftarrow \text{selected}_{\text{freqs}}(j)$
        $i \leftarrow 2 \times \text{fft}_{\text{index}}$                    /* Mapping Fourier to intDCT */
        $\hat{\mathbf{Y}}(i-1) \leftarrow \lfloor (\mathbf{Y}(i-3) + \mathbf{Y}(i-5))/2 \rfloor$
        $p_w \leftarrow \mathbf{Y}(i-1) - \hat{\mathbf{Y}}(i-1)$
        $p \leftarrow \lfloor p_w/4 \rfloor$
        $w_o \leftarrow p_w - 4 \times p$                       /* Extracting at odd index */
        $\mathbf{w}_{\text{win}}(k) \leftarrow \text{mod}(w_o, 2)$; $\mathbf{w}_{\text{win}}(k+1) \leftarrow \text{mod}(\lfloor w_o/2 \rfloor, 2)$
        $\mathbf{X}(i-1) \leftarrow \hat{\mathbf{Y}}(i-1) + p$          /* Restoring original intDCT value */
        $\hat{\mathbf{Y}}(i) \leftarrow \lfloor (\mathbf{Y}(i-2) + \mathbf{Y}(i-4))/2 \rfloor$
        $p_w \leftarrow \mathbf{Y}(i) - \hat{\mathbf{Y}}(i)$
        $p \leftarrow \lfloor p_w/4 \rfloor$
        $w_o \leftarrow p_w - 4 \times p$                      /* Extracting at even index */
        $\mathbf{w}_{\text{win}}(k+2) \leftarrow \text{mod}(w_o, 2)$; $\mathbf{w}_{\text{win}}(k+3) \leftarrow \text{mod}(\lfloor w_o/2 \rfloor, 2)$
        $\mathbf{X}(i) \leftarrow \hat{\mathbf{Y}}(i) + p$               /* Restoring original intDCT value */
        $j \leftarrow j + 1$
    end
    $\mathbf{w}_{\text{win}} \leftarrow \mathbf{w}_{\text{win}}(k_{\text{win}} : -1 : 1)$
    $\mathbf{w} \leftarrow \text{concatenate}(\mathbf{w}, \mathbf{w}_{\text{win}})$
    $\mathbf{x}_{\text{win}} \leftarrow \text{invIntDCT}(\mathbf{X})$
    $\mathbf{x}((n-1) \times n_{\text{win}} + 1 : n \times n_{\text{win}}) \leftarrow \mathbf{x}_{\text{win}}$
end
```

Algorithm 7: Extraction algorithm for the proposed RWS.

Chapter 5

Self-recovery for audio

This chapter presents the general idea of self-recovery schemes, and the necessity on the proposal of the fragile reversible watermarking scheme from the previous chapter is explained. This reversible watermarking scheme has a significant role on the proposal of the self-recovery scheme, since the insertion of the control bits that allow the restoration of the signal is achieved by increasing the payload capacity. The encoding and decoding processes of the self-recovery scheme are explained in detail, along with some examples and the algorithms.

Self-recovery watermarking schemes were first proposed for images, and arose by the necessity of restoring missing areas in the images besides identifying tampered regions in them. Although every scheme uses a different strategy, the general ideas for the encoding and decoding processes are as follows. The encoding process calculates reference bits and check bits from the image. The reference bits are a reduced version of the image itself (calculated by compressing or obtaining a descriptive representation of the image), and the check bits are the result of feeding regions of the image to a hash function. Both reference bits and check bits are scattered through the image for embedding, obtaining in this manner the watermarked image. The decoding process receives an image and extracts a watermark, from which the extracted reference bits and check bits are obtained. The extracted check bits are compared against the check bits calculated from the received image to identify the tampered regions. By using the reference bits from non-tampered regions of the image, the tampered reference bits can be restored, and with both the non-tampered and restored reference bits, the tampered areas of the image can be recovered.

The strategies for self-recovery schemes in the literature are designed for images and deal with 2D signals. Although there are fragile watermarking schemes for audio signals (1D signals), they only solve the authentication problem and in some cases they achieve tamper detection; however, a self-recovery scheme suitable for audio had to be proposed.

One of the greatest challenges with self-recovery for audio is the distortion caused by the embedding process. The goal applications where this scheme is to be used require audio signals with a transparency over -2 ODG. The objective difference grade (ODG) is the transparency metric recommended by ITU-R B.S.1387 [Thiede et al., 2000]. Because of this transparency restriction, a strategy to reduce perceptual impact had to be devised by the use of the auditory masking properties of the signals in the integer Discrete Cosine Transform (intDCT) domain. The intDCT domain was selected because it gives a representation of the signal in the frequency domain, where the watermark can be inserted selectively on frequency components that better mask the insertion noise. The intDCT also maps an integer time-domain signal to its integer frequency components; these components need to be integer because embedding and extraction of the watermark are performed with reversible algorithms, which require integer values to maintain reversibility.

5.1 Encoding process

The steps of the encoding process in the proposed scheme are presented in Figure 5.1. The work from [Zhang and Wang, 2008] was taken as base for this scheme, the general ideas were analysed and adapted for audio signals. Because of the dimensionality of audio signals, it is difficult to process them as a whole, as it is done in schemes for images. The proposed self-recovery strategy processes windows of samples. For an audio signal of size N, select a window of samples with length N_w, there is a total of $\lfloor N/N_w \rfloor$ windows for the signal. To increase the accuracy of tamper detection, and for implementation purposes, each window being processed is divided into segments of length N_s, for each window, there is a total of $\lfloor N_w/N_s \rfloor$ segments. The implementation of the proposed scheme uses windows of size $N_w = 44,032$, and segments of size $N_s = 512$; these values were determined experimentally, the window size was selected so it was the value power of two closer to one second of samples, and the segment size of 512 was found to be adequate in resolution for tamper detection.

5.1.1 Reference bits generation

In this step, the bits that will be used to restore the signal are generated. The audio signals considered for the scheme have CD quality, where each sample is represented by 16 bits. Since this amount of information cannot be embedded within the signal, it must be reduced. The binary representation for each sample in a window is obtained,

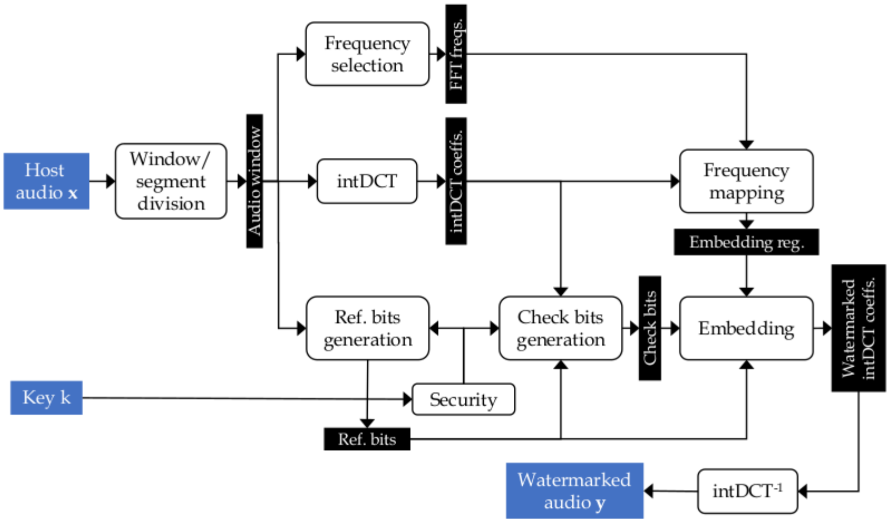


Figure 5.1: Block diagram of the encoding process.

producing $16 \times N_w$ bits per window. Pseudo-randomly permute those bits based on the secret key, and reshape them into N_w/n_g bit-groups. Variable n_g can take any value power of 2 smaller than the length of the window, *i.e.* $n_g = \{2^g | 2^g < N_w\}$, where $g = \{1, 2, \cdots\}$. Each bit-group contains $n_b = n_g \times 16$ bits. Denote the bits in a bit-group as $\mathbf{b_t}[1], \mathbf{b_t}[2], \cdots, \mathbf{b_t}[n_b]$ where $t = 1, 2, \cdots, N_w/n_g$. For each bit-group, calculate $n_{rb} = n_b/(16 \times cr)$ reference bits $\mathbf{r_t}[1], \mathbf{r_t}[2], \cdots, \mathbf{r_t}[n_{rb}]$ where 'cr' is the compression ratio of the bits, *i.e.* $cr = 2$ will keep $\frac{1}{32}$ of the $16 \times N_w$ original bits, $cr = 4$ will keep $\frac{1}{64}$ of the $16 \times N_w$ original bits, and so on. The reference bits are calculated in the following way:

$$\begin{bmatrix} \mathbf{r_t}[1] \\ \mathbf{r_t}[2] \\ \vdots \\ \mathbf{r_t}[n_{rb}] \end{bmatrix} = A_t \cdot \begin{bmatrix} \mathbf{b_t}[1] \\ \mathbf{b_t}[2] \\ \vdots \\ \mathbf{b_t}[n_b] \end{bmatrix}, \qquad (5.1)$$
$$t = 1, 2, \cdots, \frac{N_w}{n_g},$$

where A_t are pseudo-random binary matrices of size $n_{rb} \times n_b$, the matrices A_t are calculated based on the secret key and the arithmetic in eq. 5.1 is modulo-2. The final reference bits are pseudo-randomly permuted based on the secret key. These steps are described in Algorithm 8, where the function binaryRep(.) traduces each scalar within a vector to 16 scalars that correspond to its binary representation, *i.e.* sample value $\{255\}$ is traduced to $\{0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1\}$. Function randPermut(.) generates a pseudo-random permutation of a given length taking the value of 'key' as its seed. For this implementation, $N_w = 44,032$ was selected to process windows of approximately 1 second, $n_g = 256$ was determined experimentally to divide the binary representation of the signal for an adequate dispersion of the reference bits. With those values, $44,032/256 = 172$ bit-groups are constructed, and each bit-group contains $n_b = 4,096$ bits. The compression ratio is set to 2, which means that there are $n_{rb} = 128$ reference bits per group. The A_t matrices have sizes of $128 \times 4,096$, and a total of $N_w/2 = 22,016$ reference bits are obtained.

Suppose that the integer representation of the signal $\mathbf{x}$ is the following:

1	···	N_w
230	···	

Its binary representation $\mathbf{x}_{\text{bin}}$ would then be:

1	2								···								$16 \times N_w$
0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	···	

Suppose now that the permuted vector $\mathbf{x}_{\text{perm}}$ is the following:

1	2							···	$16 \times N_w$
0	1	0	0	1	1	0	0	···	

The $\mathbf{b_t}$ bit-groups would be separated in the following matrix:

	1	2	···	N_w/n_g
1	0			
2	1			
	0			
	0			
	1			
⋮	1			
	0			
	0			
	⋮			
n_b				

The reference bits calculated in lines 12-15 from Algorithm 8 would result in the following matrix:

	1	2	···	N_w/n_g
1	1			
2	0			
⋮	⋮			
n_{rb}				

```
**Input**  : Time-domain audio ($\mathbf{x}$), windows size ($N_w$), number of groups ($n_g$),
             compression ratio ($c_r$), secret key (key)
**Output**: Reference bits ($\mathbf{r_t}$)
$\mathbf{x}_{\text{bin}} \leftarrow \text{binaryRep}(\mathbf{x})$
perm $\leftarrow$ randPermut($|\mathbf{x}_{\text{bin}}|$, key)
$\mathbf{x}_{\text{perm}} \leftarrow \mathbf{x}_{\text{bin}}(\text{perm})$
$n_b \leftarrow n_g \times 16$
$t \leftarrow \lfloor N_w/n_g \rfloor$
**for** $n = 1 : t$ **do**                                  /* Divide into groups */
    $\mathbf{b_t}(:, n) \leftarrow \mathbf{x}_{\text{perm}}((n-1) \times n_b + 1 : n \times n_b)$
**end**
$n_{rb} \leftarrow n_b/(16 \times c_r)$
$A \leftarrow \text{randi}([0\ 1], [n_{rb}, n_b, t])$
**for** $n = 1 : t$ **do**                                  /* Calculate reference bits */
    $A_t \leftarrow A(:, :, n)$
    $\mathbf{r_t}(:, n) \leftarrow \text{mod}(A_t \times \mathbf{b_t}(:, n), 2)$
**end**
perm $\leftarrow$ randPermut($\lfloor N_w/c_r \rfloor$, key)
$\mathbf{r_t} \leftarrow \text{reshape}(\mathbf{r_t}, 1, [\,])$
$\mathbf{r_t} \leftarrow \mathbf{r_t}(\text{perm})$
```

Algorithm 8: Reference bits generation.

5.1.2 Check bits generation

This step calculates the check bits that will be used to identify the segments of the signal where tampering occurs. Because any modification in the intDCT domain affects all the time domain samples in a segment of audio, there is no way of knowing, just from the time-domain representation of a signal, which samples carry watermark information and which samples do not. For this reason, the check bits are obtained from the intDCT coefficients. For each segment in the window, calculate its forward intDCT transform. Collect the intDCT coefficients, and the reference bits that correspond to the segment. Feed these values to a hash function that produces 256 hash bits per segment. There is a total of $256 \times \frac{N_w}{N_s}$ hash bits per window. Pseudo-randomly permute the hash bits from the whole window, using the secret key to determine the order. To reduce the number of check bits, divide the hash bits into $N_w/4$ subsets, then calculate a modulo-2 sum of the four hash bits in each subset, the sum will produce 64 check bits per segment and $64 \times \frac{N_w}{N_s}$ check bits per window. These steps are described in

Algorithm 9.

```
Input  : Reference bits from window ($\text{refBits}_\text{w}$), intDCT coefficients from
         window ($\text{intDCT}_\text{w}$), total segments ($\text{total}_\text{segs}$), segment size ($N_s$), secret
         key (key)
Output : Check bits of window ($\text{checkBits}_\text{w}$)
numRefBits ← $N_s/2$
for $i = 1 : total_{segs}$ do                              /* Hash bits generation */
    $\text{refBits}_s \leftarrow \text{refBits}_w((i-1) \times \text{numRefBits} + 1 : i \times \text{numRefBits})$
    $\text{intDCT}_s \leftarrow \text{intDCT}_w((i-1) \times N_s + 1 : i \times N_s)$
    $\text{hashBits}(i,:) \leftarrow \text{hash}(\text{refBits}_s, \text{intDCT}_s)$
end
$\text{hashBits} \leftarrow \text{reshape}(\text{hashBits}, 1, [\,])$
$\text{perm} \leftarrow \text{randPermut}(|\text{hashBits}|, \text{key})$
$\text{hashBits}_\text{perm} \leftarrow \text{hashBits}(\text{perm})$
$\text{checkBits}_\text{m} \leftarrow \text{vec2mat}(\text{hashBits}_\text{perm}, |\text{hashBits}|/4)$
$\text{checkBits}_\text{w} \leftarrow \text{mod}(\text{sum}(\text{checkBits}_\text{m}), 2)$
```

Algorithm 9: Check bits generation.

5.1.3 Frequency selection, FFT mapping and embedding

After the reference bits and check bits are calculated for a given window of samples, the watermark to be embedded is constructed. The embedding strategy used as part of the proposed self-recovery scheme is the prediction error expansion strategy with multi-bit embedding described in Chapter 4. For each segment in a given window of samples, the candidate frequencies in the FFT domain are calculated by obtaining the masking threshold of the audio segment. All the FFT frequencies that fall under the masking threshold are candidate ones. The candidate FFT frequencies are then mapped to the intDCT domain as already explained in Chapter 4. Once the intDCT coefficients are selected for embedding, the watermark bits constructed from the reference and check bits are embedded using the proposed PEE strategy with multi-bit embedding. The watermark bits for each segment are obtained by concatenating $N_s/2$ reference bits with the corresponding 64 check bits of the segment, the watermark is denoted as $\mathbf{w}[k]$, where $k = \{1, 2, \cdots, K\}$, and K is the size of the watermark. In this case $K = N_s/2 + 64$.

5.1.4 Security layer

In the music censorship scenario, a customer could desire to restore the original uncensored version of the song without paying the corresponding fee. Both of these things can be done if the secret key used to disperse the reference and check bits can be predicted. If a small key-space is used, a brute force algorithm could find the key. With this key, the reference bits that correspond to a certain region of the speech signal can be found in the rest of the signal; by eliminating those reference bits, the original speech would not be restored. In the other scenario, if the secret key is predicted, a customer can restore the uncensored song without payment of the fee. Because of this, a big enough key-space is necessary; a key as the one used by the Advanced Encryption Standard (AES) is recommended, *i.e.* a symmetric key of 256 bits.

5.2 Decoding process

The steps in the decoding process for the proposed scheme can be seen in Figure 5.2. As in the encoding process, an audio signal of size N is divided into windows of samples of length N_w, and each window is further divided into segments of size Ls. The decoding process is applied to each of the N/N_w windows, and the general steps are detailed below. The watermark is extracted from the intDCT coefficients of each segment, after extraction from all the segments the reference bits and check bits of the window can be reconstructed using the secret key. The intDCT coefficients are selected using the same masking threshold criteria and FFT mapping used for embedding. The extracted check bits are compared against the check bits obtained from the received signal to detect the tampered regions. The reference bits and the sample values from non-tampered regions are used to restore the tampered samples.

5.2.1 Frequency selection, FFT mapping and extraction

In a similar way as in the encoding process, for each audio segment in a given window of samples the masking threshold is calculated and the candidate FFT frequencies are selected, then they are mapped to the intDCT domain from which the watermark is extracted using the PEE extraction algorithm described in Chapter 4, also with the PEE extraction algorithm a recovered version of the intDCT coefficients is obtained. Once the watermarks embedded in each segment are extracted, they can be divided into reference bits and check bits, and with the secret key the original order of reference and check bits can be obtained.

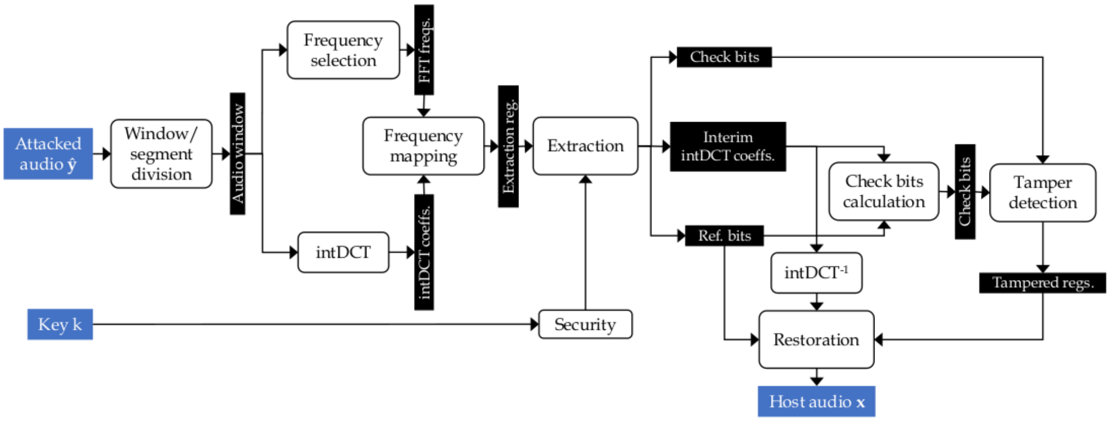


Figure 5.2: Block diagram of the decoding process.

5.2.2 Tamper detection

The check bits extracted in the previous step are compared against the check bits calculated from the extracted reference bits and the restored intDCT values obtained with the PEE extraction algorithm from Chapter 4. The consistency between these check bits is the criteria for judging a segment as *non-tampered* or *tampered*.

To calculate the check bits from the received signal, the restored intDCT coefficients of each segment are collected, along with the reference bits that correspond to that segment. All these values are fed to the same hash function to obtain 256 hash bits per segment, the $256 \times \frac{N_w}{N_s}$ hash bits are pseudo-randomly permuted in the same way as in the encoding process, and the hash bits are divided into $N_w/4$ subsets, as in the encoding process, calculate a modulo-2 sum of the 4 bits in each subset to obtain $64 \times \frac{N_w}{N_s}$ "calculated check bits". These calculated check bits are obtained following the same algorithm as in the encoding process (Algorithm 9).

The 64 calculated check bits are compared against the extracted check bits. Denote the number of extracted check bits in a segment as N_E, and be N_F the number of extracted check bits different to their corresponding calculated check bits, where $N_F \leqslant N_E$. If a segment is tampered, the probability that a calculated check bit is unequal to its corresponding extracted check bit is 0.5. Therefore, N_F follows a binomial distribution, and its probability distribution function is the following:

$$P_{T,N_F}(l) = \binom{N_E}{l} \times (0.5)^{N_E} \times (0.5)^{N_E - l}, \qquad (5.2)$$
$$l = 0, 1, \cdots, N_E.$$

For a given N_E, an integer T is found, such that it satisfies:

$$\sum_{l=0}^{T} P_T, N_F(l) < 10^{-9}, \tag{5.3}$$

and

$$\sum_{l=0}^{T+1} P_T, N_F(l) \geqslant 10^{-9}. \tag{5.4}$$

where $P_T, N_F(l)$ is the probability distribution function of having l successes in N_F trials. If $N_F > T$ then the segment is regarded as "tampered" and "non-tampered" otherwise. The probability of falsely identifying a tampered segment as a non-tampered one is less than 10^{-9}. The steps for tamper detection are outlined in Algorithm 10.

5.2.3 Signal restoration

In this final step, the original sample values from "tampered" segments are restored. Mark the reference bits and sample values from each tampered segment as 'NaN' values to facilitate its differentiation from the reference bits and samples from non-tampered segments in the next steps. The vectors and matrices from eq. 5.1 are recalculated with the extracted reference bits and the interim restored signal obtained so far (the time-domain signal obtained after watermark extraction). Because the received signal is quantized at 16 bits, each 'NaN' in the interim restored signal is traduced to 16 'NaN' values in the binary representation of the signal.

The $16 \times N_w$ bits of the binary representation of the signal are divided into Lw/n_g groups as in the encoding process, each group contains $n_b = n_g \times 16$ bits. The number of reliable reference bits in each bit-group, denoted as n_t, may be less than the original n_{rb} reference bits from encoding. Equation 5.1 implies that:

$$\begin{bmatrix} \mathbf{r_t}[s_1] \\ \mathbf{r_t}[s_2] \\ \vdots \\ \mathbf{r_t}[s_{n_t}] \end{bmatrix} = A_t^{(R)} \cdot \begin{bmatrix} \mathbf{b_t}[1] \\ \mathbf{b_t}[2] \\ \vdots \\ \mathbf{b_t}[n_b] \end{bmatrix}, \qquad t = 1, 2, \cdots, \frac{N_w}{n_g}. \tag{5.5}$$

The $\mathbf{r_t}$ vectors contain the reliably extracted reference bits and $A_t^{(R)}$ is a matrix that has all the rows from A_t that correspond to the reliably extracted reference bits, *i.e.* all the rows in $\mathbf{r_t}$ with 'NaN' values are removed and the same rows from A_t are removed to obtain $A_t^{(R)}$. On the other side of eq. 5.5, the n_b bits in each bit-group contain two

```
Input  : Extracted reference bits ($refBits_w$), extracted check bits ($checkBits_w$),
         restored intDCT coefficients ($intDCT_w$), total segments ($total_{segs}$),
         segment size ($N_s$), secret key (key), number of extracted check bits ($N_E$)
Output : Tampered segments ($tampered_{segs}$)
                                            /* Calculate check bits with Algorithm 9 */
$calcCheckBits_w \leftarrow checkBits_{gen}(refBits_w, intDCT_w, total_{segs}, N_s, key)$
                                                                 /* Find threshold T */
$T \leftarrow N_E + 1$
$l_b \leftarrow 10^{-9}$
$u_b \leftarrow 10^{-9} - 1$
while $l_b >= 10^{-9}$ or $u_b < 10^{-9}$ do
    $T \leftarrow T - 1$
    $l_b \leftarrow \sum_{l=0}^{T} P_T, N_F(l)$
    $u_b \leftarrow \sum_{l=0}^{T+1} P_T, N_F(l)$
end
                                                /* Tampered segment identification */
for $i = 1 : total_{segs}$ do
    $checkBits_s \leftarrow checkBits_w((i-1) \times 64 + 1, i \times 64)$
    $calcCheckBits_s \leftarrow calcCheckBits_w((i-1) \times 64 + 1, i \times 64)$
    $N_F \leftarrow compareCheckBits(checkBits_s, calcCheckBits_s)$
    if $N_F > T$ then
        $tampered_{segs}(i) \leftarrow$ 'True'
    else
        $tampered_{segs}(i) \leftarrow$ 'False'
    end
end
```

Algorithm 10: Tamper detection.

types of bits: 1) the missing bits from "tampered" segments, and 2) the recovered bits from other positions. The assumption of this restoration strategy relies on the fact that, if a small region of the signal was tampered, then the missing bits in each $\mathbf{b_t}$ are small (because those missing bits are dispersed throughout different bit-groups) and do not affect the restoration.

In this way, the reliable reference bits and the non-missing bits in the $\mathbf{b_t}$ groups can provide enough information to recover the original values of missing bits. Let $\mathbf{B_{t,1}}$ be a column vector that corresponds to the missing bits from $\mathbf{b_t}$, and $\mathbf{B_{t,2}}$ a column vector that corresponds to the recovered bits in $\mathbf{b_t}$, *i.e.* $\mathbf{B_{t,1}}$ is a column vector that corresponds to the rows in b_t that contain 'NaN' values and $\mathbf{B_{t,2}}$ is a column vector that corresponds to the rows in $\mathbf{b_t}$ with values different to 'NaN'. Then eq. 5.5 can be reformulated as:

$$\begin{bmatrix} \mathbf{r_t}[s_1] \\ \mathbf{r_t}[s_2] \\ \vdots \\ \mathbf{r_t}[s_{n_t}] \end{bmatrix} - A_t^{(R,2)} \cdot \mathbf{B_{t,2}} = A_t^{(R,1)} \cdot \mathbf{B_{t,1}}, \tag{5.6}$$

where $A_t^{(R,1)}$ is a matrix constructed from the columns of $A_t^{(R)}$ that correspond to the missing bits in $\mathbf{b_t}$, and $A_t^{(R,2)}$ is a matrix constructed from the columns of $A_t^{(R)}$ that correspond to the recovered bits in $\mathbf{b_t}$. From eq. 5.6, the left side and the matrix $A_t^{(R,1)}$ are known variables, so only $\mathbf{B_{t,1}}$ has to be found. Let n_{mb} be the number of elements in $\mathbf{B_{t,1}}$, then the size of the matrix $A_t^{(R,1)}$ is $n_t \times n_{mb}$. n_{mb} unknowns are solved according to the n_t equations in the binary system, so the idea is to solve eq. 5.6 for $\mathbf{B_{t,1}}$, therefore obtaining the missing bits. With those missing bits, the 16-bit representation of the signal can be restored.

The next chapter presents the experimental results obtained with the proposed reversible watermarking scheme, the self-recovery scheme, the implementation of the framework for images, and finally the implementation of the framework for audio signals.

Chapter 6

Experimental results

This chapter presents the experimental results obtained with the proposed reversible watermarking scheme, the self-recovery scheme, the implementation of the framework for images, and finally the implementation of the framework for audio signals. The first stage of the development of the reversible watermarking scheme was the improvement in its transparency, which is required for practical applications; the second stage consists in increasing its payload capacity to be able to insert the control bits required by the self-recovery scheme.

To test the proposed fragile reversible watermarking scheme, the self-recovery scheme and the framework for audio, experiments with 3 databases were performed, namely the Music Audio Benchmark (MAB) of the University of Dortmund [Homburg et al., 2005], the Ballroom (Ball) dataset [Gouyon, 2006], and a dataset compiled by our research group (Ours). The test audio signals from all the datasets have CD quality, quantized at 16 bits with a sampling frequency of 44.1 kHz, and each dataset has audio signals of various musical genres. The duration of the audio signals in each dataset are of 10, 30, and 20 seconds, respectively. A detailed explanation of the datasets is given in Appendix B.

6.1 Audio RWS

Initial efforts to construct the framework for audio signals utilized reversible watermarking schemes that dealt with audio signals in its time domain representation. If a sufficient amount of reference bits was embedded, the restoration capabilities of the self-recovery stage were acceptable; however, the transparency of such strategies was not acceptable for a watermarking scheme. Various classical reversible watermarking techniques that work with time domain signals were considered, in order to reduce perceptual impact. But because of the required payload capacity, these schemes could not provide the transparency needed.

A time-domain strategy that uses prediction-error expansion with a simple predictor that obtains the mean value between two samples was tested; the payload inserted was 24.8 kbps, which is the payload required by the self-recovery scheme. Table 6.1 presents the transparency results obtained for the dataset ´Ours' measured with the PSNR, MSE, and ODG. It is worth mentioning that the PSNR, and MSE values are calculated from the normalized audio signals $\in [-1, 1]$. As it can be seen from this table, the average ODG values are almost -4, which indicates a very annoying distortion to the watermarked signals; therefore, this PEE strategy in time-domain is not suitable for the framework.

Table 6.1: Transparency results for PEE strategy in time domain *.

	PSNR				MSE				ODG			
Genre	μ	σ	**Min**	**Max**	μ	σ	**Min**	**Max**	μ	σ	**Min**	**Max**
Jazz	26.73	2.12	23.46	30.92	2.35×10^{-3}	1.07×10^{-3}	8.10×10^{-4}	4.51×10^{-3}	-3.89	0.02	-3.91	-3.85
Orchestra	29.80	6.39	22.18	40.76	2.15×10^{-3}	2.24×10^{-3}	8.40×10^{-5}	6.05×10^{-3}	-3.90	0.01	-3.91	-3.88
Rock	17.83	2.27	14.75	21.48	1.85×10^{-2}	9.04×10^{-3}	7.11×10^{-3}	3.35×10^{-2}	-3.86	0.02	-3.88	-3.84
Pop	22.47	2.45	17.03	25.17	6.69×10^{-3}	4.97×10^{-3}	3.04×10^{-3}	1.98×10^{-2}	-3.90	0.01	-3.91	-3.88
Vocal	22.58	1.94	20.5	26.50	5.98×10^{-3}	2.23×10^{-3}	2.24×10^{-3}	8.92×10^{-3}	-3.90	0.01	-3.91	-3.89

* Embedded payload = 24.8 kbps

6.1.1 Transparency improvement

To improve the transparency of the self-recovery scheme and of the whole framework, it was identified that the frequency components of the audio signals had to be exploited to reduce the degradation of the signals during embedding. Therefore, a reversible watermarking scheme for audio signals in the intDCT domain was proposed. An experimental setup was devised to verify that the intDCT domain could provide the transparency required by the scheme, given the payload capacity needed for the insertion of control bits in the self-recovery scheme.

Motivation.

The goal of this experimental setup was to evaluate that the insertion of the watermark in the intDCT domain allows the reduction of perceptual impact of the scheme, by the selection of the intDCT coefficients that produce an imperceptible change in the watermarked signals.

Parameters.

The required payload to be embedded is of $N_s/4 + 64$ bits per segment, which is the amount of control bits that would be inserted with the self-recovery scheme, this is equivalent to 16.5 kbps. The window size is $N_w = 44,032$, and the segment size is $N_s = 512$. The selection of the intDCT coefficients is determined through a parameter 'lowBand' $\in [10, 340]$ that is increased in steps of 30, to find the appropriate value to reduce perceptual impact.

Assumptions.

It is assumed that the dynamic range of the audio signals to be processed has been previously adjusted to a range in $[0, 32768]$ in a preprocessing stage. This adjustment is required to prevent underflow and overflow problems, and is an integer range to apply the intDCT transform that maps an integer representation in the time domain to another integer representation but in the frequency domain. Because this reversible watermarking scheme is to be used as part of the self-recovery scheme for audio signals, the payload size is calculated as it would be for that scheme. It is also assumed that most natural audio signals have a similar range of frequencies where the highest part of their energy is concentrated. Following that assumption, the 'lowBand' parameter can be experimentally determined with just one dataset. The average 'lowBand' for that dataset can then be used for the rest of the datasets, since it is expected that the frequencies in those audio signals will behave in a similar fashion.

Results.

Table 6.2 presents the mean (μ), standard deviation (σ), minimum, and maximum values measured with the peak signal-to-noise ratio (PSNR), mean square error (MSE), and ODG between the host and watermarked audio signals. In this table, the numbers in bold indicate the offsets that produce better ODG results for each genre. These results show that with an adequate 'lowBand' offset value the average ODG is not only above the -2 threshold, but even reaches values over -1.2. These ODG results indicate that the transparency of the watermarked audio signals is more than sufficient for practical applications.

The quality results of the embedding process for the three datasets were evaluated with the MSE, PSNR, and ODG metrics. Table 6.3 presents the mean (μ), standard deviation (σ), minimum, and maximum values measured with PSNR, MSE, and ODG between the host and watermarked audio signals, the inserted payload is of 16.5 kbps.

As it can be seen in the column of mean ODG values, the ODG is over -2 for all the genres in the three datasets tested, except for 'Orchestra' marked in bold red in Table 6.3.

Discussion.

The first part of this experimental setup allowed the determination of the 'lowBand' parameter to be used for the three datasets. The assumption of the 'lowBand' parameter being fixed for all the datasets was corroborated with the embedding results from Table 6.3, since for almost all the genres in the three datasets, the ODG results are over the threshold of -2, except for the 'Orchestra' genre. This occurs because audio signals from this genre are low energy ones with a limited frequency distribution. Since the proposed strategy selects a continuous region of intDCT frequencies for embedding, the modification of some of the frequencies within the embedding region produces more noticeable distortions in these low energy signals.

Conclusions.

The use of the intDCT domain allows the reduction of perceptual impact, compared to the insertion of the watermarks in the time domain representation. This occurs because the intDCT coefficients can be selected through the 'lowBand' parameter, which was adapted for various genres.

Table 6.2: Effect of 'lowBand' offsets on embedding quality for dataset 'Ours'.

Genre	Low	PSNR				MSE				ODG			
	band	μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
Jazz	10	26.9	0.7	25.3	27.6	2.05×10^{-3}	3.75×10^{-4}	1.76×10^{-3}	2.96×10^{-3}	-3.9	0.0	-3.9	-3.9
	40	42.7	3.8	38.0	48.7	7.22×10^{-5}	5.28×10^{-5}	1.34×10^{-5}	1.60×10^{-4}	-3.4	0.5	-3.9	-2.1
	70	47.8	4.4	42.1	55.4	2.48×10^{-5}	2.05×10^{-5}	2.88×10^{-6}	6.23×10^{-5}	-2.7	0.8	-3.8	-1.4
	100	51.4	4.5	45.9	59.5	1.10×10^{-5}	9.46×10^{-6}	1.11×10^{-6}	2.57×10^{-5}	-2.2	0.9	-3.4	-1.1
	130	50.5	1.7	47.8	52.6	9.61×10^{-6}	3.85×10^{-6}	5.53×10^{-6}	1.67×10^{-5}	-1.9	0.6	-2.7	-0.9
	160	51.7	0.9	50.0	52.8	6.84×10^{-6}	1.46×10^{-6}	5.27×10^{-6}	9.94×10^{-6}	-1.5	0.7	-2.2	-0.4
	190	52.1	0.7	50.6	52.9	6.19×10^{-6}	1.04×10^{-6}	5.15×10^{-6}	8.65×10^{-6}	-1.2	0.8	-2.1	-0.3
	220	51.5	1.1	49.3	52.8	7.32×10^{-6}	2.03×10^{-6}	5.27×10^{-6}	1.18×10^{-5}	-1.4	0.7	-2.1	-0.5
	250	50.1	2.1	46.8	52.7	1.09×10^{-5}	5.42×10^{-6}	5.41×10^{-6}	2.07×10^{-5}	-1.5	0.6	-2.1	-0.8
	280	51.1	5.2	44.5	61.0	1.30×10^{-5}	1.22×10^{-5}	7.94×10^{-7}	3.52×10^{-5}	-1.1	0.1	-1.2	-1.0
	310	47.4	5.6	40.8	58.6	3.05×10^{-5}	2.61×10^{-5}	1.38×10^{-6}	8.27×10^{-5}	-1.5	0.1	-1.6	-1.3
	340	43.4	6.1	35.8	54.2	8.87×10^{-5}	8.74×10^{-5}	3.79×10^{-6}	2.62×10^{-4}	-2.0	0.4	-2.6	-1.7
Orchestra	10	26.3	1.5	23.6	27.7	2.48×10^{-3}	9.65×10^{-4}	1.71×10^{-3}	4.40×10^{-3}	-3.9	0.0	-3.9	-3.9
	40	45.6	4.6	36.8	49.5	4.97×10^{-5}	6.43×10^{-5}	1.11×10^{-5}	2.10×10^{-4}	-3.7	0.6	-3.9	-2.0
	70	55.4	3.5	48.1	57.7	4.18×10^{-6}	4.55×10^{-6}	1.68×10^{-6}	1.56×10^{-5}	-3.4	0.7	-3.9	-1.5
	100	61.2	3.6	53.1	63.4	1.15×10^{-6}	1.41×10^{-6}	4.57×10^{-7}	4.90×10^{-6}	-2.9	0.7	-3.8	-1.4
	130	52.6	0.5	51.2	52.9	5.56×10^{-6}	7.59×10^{-7}	5.18×10^{-6}	7.59×10^{-6}	-2.2	0.4	-2.6	-1.2
	160	52.8	0.4	51.8	52.9	5.32×10^{-6}	4.69×10^{-7}	5.08×10^{-6}	6.58×10^{-6}	-2.0	0.5	-2.2	-0.6
	190	52.8	0.3	51.9	53.0	5.27×10^{-6}	4.40×10^{-7}	5.05×10^{-6}	6.43×10^{-6}	-1.8	0.6	-2.2	-0.4
	220	52.8	0.4	51.6	53.0	5.32×10^{-6}	5.72×10^{-7}	5.03×10^{-6}	6.85×10^{-6}	-1.8	0.6	-2.2	-0.6
	250	52.7	0.6	51.1	53.0	5.45×10^{-6}	8.42×10^{-7}	5.03×10^{-6}	7.68×10^{-6}	-1.9	0.4	-2.2	-1.0
	280	71.7	12.4	52.3	85.0	9.67×10^{-7}	1.87×10^{-6}	3.15×10^{-9}	5.85×10^{-6}	-1.2	0.3	-1.9	-0.6
	310	65.5	13.5	46.7	86.0	3.87×10^{-6}	6.89×10^{-6}	2.49×10^{-9}	2.13×10^{-5}	-1.6	0.2	-1.9	-1.2
	340	48.6	12.1	32.5	74.1	9.83×10^{-5}	1.78×10^{-4}	3.93×10^{-8}	5.69×10^{-4}	-2.7	0.6	-3.5	-1.7

Continues on next page

Genre	Low	PSNR				MSE				ODG			
	band	μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
Pop	10	25.0	1.6	20.8	26.9	3.42×10^{-3}	1.80×10^{-3}	2.07×10^{-3}	8.40×10^{-3}	-3.9	0.1	-3.9	-3.6
	40	39.1	3.1	32.2	42.6	1.64×10^{-4}	1.67×10^{-4}	5.45×10^{-5}	6.08×10^{-4}	-3.3	0.7	-3.8	-1.9
	70	41.8	3.7	35.1	48.6	9.19×10^{-5}	8.78×10^{-5}	1.37×10^{-5}	3.07×10^{-4}	-2.9	0.9	-3.7	-1.2
	100	44.1	3.8	37.6	51.6	5.40×10^{-5}	4.96×10^{-5}	6.98×10^{-6}	1.73×10^{-4}	-2.6	0.9	-3.6	-1.1
	130	46.6	3.2	40.3	51.0	2.92×10^{-5}	2.66×10^{-5}	7.92×10^{-6}	9.35×10^{-5}	-2.0	0.7	-2.6	-0.8
	160	50.0	2.4	44.3	52.4	1.19×10^{-5}	9.53×10^{-6}	5.73×10^{-6}	3.72×10^{-5}	-1.3	0.6	-2.1	-0.5
	190	50.9	2.2	45.4	52.7	9.38×10^{-6}	7.10×10^{-6}	5.39×10^{-6}	2.90×10^{-5}	-0.8	0.5	-1.7	-0.3
	220	49.3	2.7	43.5	52.2	1.45×10^{-5}	1.18×10^{-5}	6.07×10^{-6}	4.42×10^{-5}	-0.8	0.3	-1.5	-0.5
	250	45.7	3.4	39.3	50.6	3.64×10^{-5}	3.42×10^{-5}	8.81×10^{-6}	1.18×10^{-4}	-1.1	0.2	-1.5	-0.9
	280	43.5	4.1	36.9	51.9	6.39×10^{-5}	5.90×10^{-5}	6.43×10^{-6}	2.04×10^{-4}	-1.2	0.1	-1.3	-1.1
	310	41.5	4.1	34.5	49.3	1.03×10^{-4}	1.04×10^{-4}	1.18×10^{-5}	3.58×10^{-4}	-1.6	0.1	-1.7	-1.4
	340	38.4	3.4	31.2	42.7	2.00×10^{-4}	2.12×10^{-4}	5.39×10^{-5}	7.64×10^{-4}	-2.2	0.3	-2.5	-1.7
Rock	10	21.7	1.5	19.1	24.3	7.13×10^{-3}	2.55×10^{-3}	3.74×10^{-3}	1.24×10^{-2}	-3.5	0.2	-3.7	-3.2
	40	30.3	2.1	27.8	33.4	1.05×10^{-3}	4.85×10^{-4}	4.61×10^{-4}	1.68×10^{-3}	-1.7	0.1	-1.9	-1.5
	70	36.0	2.6	32.3	40.4	2.90×10^{-4}	1.53×10^{-4}	9.08×10^{-5}	5.90×10^{-4}	-1.1	0.1	-1.2	-1.0
	100	40.7	2.3	37.3	44.0	9.68×10^{-5}	4.96×10^{-5}	4.00×10^{-5}	1.86×10^{-4}	-1.0	0.0	-1.0	-0.9
	130	44.1	2.1	40.6	47.4	4.29×10^{-5}	2.14×10^{-5}	1.82×10^{-5}	8.64×10^{-5}	-0.9	0.1	-0.9	-0.8
	160	48.0	1.9	44.5	51.0	1.75×10^{-5}	8.51×10^{-6}	8.03×10^{-6}	3.55×10^{-5}	-0.5	0.0	-0.6	-0.5
	190	49.3	1.8	45.7	51.9	1.28×10^{-5}	6.03×10^{-6}	6.49×10^{-6}	2.67×10^{-5}	-0.4	0.0	-0.4	-0.3
	220	47.0	2.1	43.4	50.2	2.22×10^{-5}	1.18×10^{-5}	9.48×10^{-6}	4.59×10^{-5}	-0.6	0.1	-0.7	-0.6
	250	43.1	2.1	39.6	46.3	5.46×10^{-5}	2.64×10^{-5}	2.34×10^{-5}	1.10×10^{-4}	-0.9	0.1	-1.0	-0.8
	280	39.6	2.5	35.7	43.4	1.27×10^{-4}	7.42×10^{-5}	4.60×10^{-5}	2.71×10^{-4}	-1.0	0.0	-1.1	-1.0
	310	34.7	2.5	31.1	38.6	3.90×10^{-4}	1.99×10^{-4}	1.37×10^{-4}	7.72×10^{-4}	-1.4	0.1	-1.5	-1.3
	340	28.4	2.1	25.5	31.7	1.59×10^{-3}	7.31×10^{-4}	6.71×10^{-4}	2.84×10^{-3}	-1.9	0.1	-2.2	-1.6

Continues on next page

Genre	Low	PSNR				MSE				ODG			
	band	μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
Vocal	10	25.9	0.7	24.8	26.8	2.63×10^{-3}	4.00×10^{-4}	2.09×10^{-3}	3.30×10^{-3}	-3.9	0.0	-3.9	-3.9
	40	39.6	3.1	35.8	46.5	1.31×10^{-4}	7.28×10^{-5}	2.26×10^{-5}	2.63×10^{-4}	-3.3	0.5	-3.9	-2.5
	70	43.6	4.6	37.8	54.2	6.30×10^{-5}	4.70×10^{-5}	3.82×10^{-6}	1.66×10^{-4}	-2.6	0.8	-3.8	-1.4
	100	47.1	4.8	41.7	58.0	2.81×10^{-5}	2.07×10^{-5}	1.57×10^{-6}	6.75×10^{-5}	-2.1	0.8	-3.5	-1.0
	130	49.4	1.8	46.8	52.3	1.24×10^{-5}	5.03×10^{-6}	5.89×10^{-6}	2.11×10^{-5}	-1.7	0.5	-2.2	-1.0
	160	51.9	0.9	49.9	52.7	6.56×10^{-6}	1.47×10^{-6}	5.38×10^{-6}	1.02×10^{-5}	-1.0	0.5	-2.2	-0.5
	190	52.3	0.5	51.1	52.8	5.88×10^{-6}	7.73×10^{-7}	5.23×10^{-6}	7.75×10^{-6}	-0.6	0.5	-2.1	-0.3
	220	51.6	1.1	49.2	52.6	7.17×10^{-6}	2.01×10^{-6}	5.47×10^{-6}	1.21×10^{-5}	-0.8	0.5	-2.1	-0.6
	250	48.3	2.5	44.6	52.3	1.71×10^{-5}	9.25×10^{-6}	5.84×10^{-6}	3.49×10^{-5}	-1.2	0.4	-2.1	-1.0
	280	46.6	5.1	41.1	59.0	3.28×10^{-5}	2.42×10^{-5}	1.26×10^{-6}	7.73×10^{-5}	-1.1	0.1	-1.3	-1.0
	310	43.1	5.3	36.8	55.6	7.59×10^{-5}	5.88×10^{-5}	2.75×10^{-6}	2.08×10^{-4}	-1.5	0.1	-1.7	-1.4
	340	38.7	2.4	35.5	43.4	1.52×10^{-4}	7.56×10^{-5}	4.61×10^{-5}	2.83×10^{-4}	-2.1	0.3	-2.6	-1.6

Table 6.3: Embedding* results for the tested datasets.

Dataset	Genre	PSNR				MSE				ODG			
		μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
Ball	ChaChaCha	44.1	3.6	38.9	48.6	5.30×10^{-5}	4.38×10^{-5}	1.39×10^{-5}	1.28×10^{-4}	-1.6	0.4	-2.1	-1.2
	Jive	41.4	2.6	37.9	44.5	8.48×10^{-5}	5.03×10^{-5}	3.57×10^{-5}	1.64×10^{-4}	-1.4	0.1	-1.4	-1.3
	Quickstep	43.9	1.7	41.2	46.5	4.40×10^{-5}	1.69×10^{-5}	2.24×10^{-5}	7.54×10^{-5}	-1.4	0.3	-2.1	-1.1
	Rumba	48.5	2.9	43.3	52.5	1.74×10^{-5}	1.27×10^{-5}	5.58×10^{-6}	4.67×10^{-5}	-1.4	0.3	-2.2	-1.1
	Samba	41.8	3.1	38.2	46.6	8.05×10^{-5}	4.74×10^{-5}	2.16×10^{-5}	1.52×10^{-4}	-1.4	0.4	-2.1	-1.2
	Tango	44.1	3.8	38.4	49.2	5.38×10^{-5}	4.32×10^{-5}	1.19×10^{-5}	1.44×10^{-4}	-1.6	0.2	-2.1	-1.4
	Waltz	47.9	2.4	43.6	51.9	1.89×10^{-5}	1.15×10^{-5}	6.42×10^{-6}	4.38×10^{-5}	-1.8	0.3	-2.1	-1.5
MAB	Alternative	48.0	4.0	40.0	52.6	2.47×10^{-5}	2.89×10^{-5}	5.53×10^{-6}	9.99×10^{-5}	-1.6	0.4	-2.1	-1.0
	Blues	45.4	3.3	39.4	50.4	3.73×10^{-5}	3.10×10^{-5}	9.08×10^{-6}	1.14×10^{-4}	-1.4	0.4	-2.1	-1.0
	Electronic	45.8	5.5	34.2	52.8	6.24×10^{-5}	1.15×10^{-4}	5.31×10^{-6}	3.84×10^{-4}	-1.7	0.5	-2.1	-1.1
	Folk	44.7	3.2	39.3	49.2	4.37×10^{-5}	3.44×10^{-5}	1.22×10^{-5}	1.17×10^{-4}	-1.4	0.4	-2.1	-1.1
	Funk	43.4	4.9	36.8	51.7	7.43×10^{-5}	6.63×10^{-5}	6.76×10^{-6}	2.11×10^{-4}	-1.3	0.3	-2.1	-1.1
	Jazz	47.8	4.0	41.0	53.0	2.40×10^{-5}	2.29×10^{-5}	5.05×10^{-6}	8.02×10^{-5}	-1.6	0.5	-2.2	-1.0
	Pop	45.6	4.0	38.3	52.1	4.11×10^{-5}	4.34×10^{-5}	6.20×10^{-6}	1.48×10^{-4}	-1.5	0.4	-2.1	-1.1
	Rock	40.6	4.1	35.9	47.9	1.20×10^{-4}	8.33×10^{-5}	1.64×10^{-5}	2.57×10^{-4}	-1.1	0.1	-1.5	-1.0
Ours	Jazz	47.5	4.5	39.2	55.4	2.85×10^{-5}	3.40×10^{-5}	2.88×10^{-6}	1.21×10^{-4}	-1.6	0.4	-2.0	-1.0
	Orchestra	57.3	10.6	50.4	81.5	4.71×10^{-6}	2.75×10^{-6}	7.04×10^{-9}	9.06×10^{-6}	**-2.0**	0.3	-2.2	-1.3
	Pop	43.8	3.4	37.7	49.7	5.58×10^{-5}	4.88×10^{-5}	1.06×10^{-5}	1.71×10^{-4}	-1.4	0.2	-1.8	-1.2
	Rock	40.8	2.2	37.4	43.7	9.34×10^{-5}	4.76×10^{-5}	4.24×10^{-5}	1.82×10^{-4}	-1.2	0.1	-1.3	-1.1
	Vocal	46.1	3.1	41.8	52.0	3.01×10^{-5}	1.91×10^{-5}	6.33×10^{-6}	6.66×10^{-5}	-1.5	0.3	-2.1	-1.3

*Embedded payload = 16.5 kbps

6.1.2 Payload increase

From the use of the intDCT domain for watermark embedding and extraction, it was found that this domain allows the reduction of perceptual impact of the scheme. The determination of the 'lowBand' parameter was performed in a heuristic manner, which could be improved with the use of genetic algorithms, for example. However, to select the intDCT coefficients in a more efficient manner, the use of the auditory masking properties of audio signals was proposed. Auditory masking allows the determination of the frequencies in the Fourier spectra where modifications are inaudible for a human listener; those frequencies in the Fourier domain can be mapped to the intDCT domain where watermarks are embedded and extracted. The use of a conventional prediction-error expansion technique allows the insertion of one bit of the watermark in each of the intDCT coefficients. However, since the fragile reversible watermarking scheme is to be used as part of the self-recovery scheme, the goal is to increase the embedding capacity while maintaining the ODG threshold. In order to increase the number of watermark bits that can be embedded in each intDCT coefficient, two approaches were identified in the literature, namely multi-bit expansion and reversible watermarking with multi-embedding.

- **Multi-bit vs. Multi-embedding.** Multi-bit expansion [Coltuc and Tudoroiu, 2012] modifies the conventional PEE strategy in a way that multiple bits can be embedded in the predicted error. From a sample value, a prediction value is calculated, and the error between them is obtained. In the conventional PEE strategy, this error is expanded by 2, and one bit can be inserted in the LSB. The multi-bit expansion approach expands the error by N, and $\log_2(N)$ bits that can be embedded. The use of multi-embedding strategy for an audio reversible watermarking scheme was first explored by [Garcia-Hernandez, 2012]. The idea behind multi-embedding is to process an audio signal in multiple stages, where in each stage the signal is processed with the encoding or decoding algorithm of the scheme, and these stages are linked in a cascade model. In each stage of the encoding algorithm with multi-embedding, one bit is inserted and a watermarked signal is produced and passed to the next stage, where a new bit is then inserted, and so on.

  Motivation.

  The goal of this experimental setup is to determine which approach allows the insertion of a higher payload with a better transparency, determined by higher ODG values. The same number of frequencies is selected for both approaches, and the same number of bits are embedded in each coefficient, then the ODG results for each strategy are compared to determine which has better transparency.

Parameters.

The number of frequencies to be selected for modification is in a range of $[50, 230]$ with increases of 30. The size of the windows are $N_w = 44,032$ and the segment size is $N_s = 512$.

Assumptions.

It is assumed that the dynamic range of the audio signals has been adapted to a range in $[0, 32768]$ to avoid underflow and overflow problems, and for the use of the intDCT transform. The multi-bit (MB) approach is performed with an expansion factor of 4 that allows the insertion of 2 watermark bits per frequency, and since for each frequency at an even index a frequency at an odd index is annexed, a total of 4 bits are inserted. The multi-embedding (ME) approach is carried out in two stages, where one bit per frequency is inserted, and a total of 2 bits per stage are inserted. Since the goal of this experimental setup is to compare the payload capacity vs. the ODG values obtained from each of the approaches, it is not necessary to limit the ODG results to a -2 threshold.

Results.

Table 6.4 presents the mean (μ), standard deviation (σ), minimum and maximum PSNR, MSE, and ODG results obtained from embedding different payloads with the two approaches considered. As it can be seen in the rows highlighted in blue, almost all of the ODG results obtained with the multi-bit (MB) approach are better than those with multi-embedding (ME). The PSNR, and MSE results for the multi-bit approach are also better than those of the other approach, since in all the cases the PSNR results are higher and the MSE results are lower than their counterparts. The statistics presented in this table were obtained from the results of all the genres in the dataset 'Ours'.

Table 6.4: Comparison between multi-bit and multi-embedding approaches.

#	Payload	Strategy	PSNR				MSE				ODG			
freqs.	(kbps)		μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
50	17.2	ME	56.4	10.0	39.4	78.5	1.09×10^{-5}	2.09×10^{-5}	1.43×10^{-8}	1.14×10^{-4}	-0.04	0.36	-1.80	0.10
50	17.2	MB	57.1	9.8	40.3	78.4	9.08×10^{-6}	1.73×10^{-5}	1.46×10^{-8}	9.42×10^{-5}	-0.06	0.38	-1.85	0.10
80	27.5	ME	50.6	11.0	34.6	75.7	4.32×10^{-5}	7.34×10^{-5}	2.69×10^{-8}	3.48×10^{-4}	-1.03	0.30	-2.06	-0.63
80	27.5	MB	51.2	10.5	35.4	74.5	3.58×10^{-5}	6.07×10^{-5}	3.55×10^{-8}	2.88×10^{-4}	-0.95	0.31	-2.13	-0.48
110	37.8	ME	44.7	13.5	11.0	75.7	1.71×10^{-4}	2.56×10^{-4}	2.69×10^{-8}	1.12×10^{-3}	-2.21	0.53	-2.83	0.30
110	37.8	MB	45.1	12.9	10.5	74.5	1.42×10^{-4}	2.12×10^{-4}	3.55×10^{-8}	9.29×10^{-4}	-2.18	0.57	-3.05	0.31
140	48.2	ME	39.4	14.4	11.0	75.7	5.24×10^{-4}	6.58×10^{-4}	2.69×10^{-8}	2.77×10^{-3}	-2.80	0.79	-3.62	0.53
140	48.2	MB	39.7	13.6	10.5	74.5	4.34×10^{-4}	5.44×10^{-4}	3.55×10^{-8}	2.28×10^{-3}	-2.78	0.83	-3.75	0.57
170	58.5	ME	35.6	13.8	11.0	75.7	1.23×10^{-3}	1.52×10^{-3}	2.69×10^{-8}	6.57×10^{-3}	-3.07	0.88	-3.81	0.79
170	58.5	MB	35.9	13.0	10.5	74.5	1.02×10^{-3}	1.26×10^{-3}	3.55×10^{-8}	5.44×10^{-3}	-3.04	0.91	-3.84	0.83
200	68.8	ME	31.2	12.6	11.0	75.7	3.23×10^{-3}	4.51×10^{-3}	2.69×10^{-8}	1.96×10^{-2}	-3.41	0.91	-3.88	0.88
200	68.8	MB	31.6	12.0	10.5	74.5	2.68×10^{-3}	3.74×10^{-3}	3.55×10^{-8}	1.62×10^{-2}	-3.38	0.93	-3.89	0.91
230	79.1	ME	22.7	9.6	10.8	75.7	1.29×10^{-2}	1.80×10^{-2}	2.69×10^{-8}	8.42×10^{-2}	-3.24	0.98	-3.91	0.91
230	79.1	MB	23.0	9.3	10.5	74.5	1.11×10^{-2}	1.49×10^{-2}	3.55×10^{-8}	7.03×10^{-2}	-3.20	0.98	-3.91	0.93

*Results obtained from all the genres in dataset ‘Ours’.

Discussion.

As it can be seen from Table 6.4, the transparency results from the multi-bit approach are better than those from multi-embedding inserting a fixed payload for both approaches, which indicates that multi-bit is a better option in terms of transparency. This suggests that the embedding capacities required by the self-recovery scheme can be fulfilled with this approach.

Conclusions.

This experimental setup demonstrated that a multi-bit approach is better in terms of transparency than a multi-embedding one. Since the fragile RWS is to be used in the self-recovery scheme, it is necessary to increase the embedding capacity while maintaining adequate transparency. From these results, multi-bit seemed to be the appropriate approach to fulfill this goal.

- **Expansion factor in multi-bit embedding.** Since it was demonstrated that the multi-bit approach produces better transparency than multi-embedding for a fixed payload, then it was explored how the expansion affects the relationship between embedding capacity vs. transparency.

Motivation.

In this experimental setup a study was made to explore how the expansion factor in the multi-bit approach affects the embedding capacity in relation to the transparency of the proposed fragile RWS.

Parameters.

The number of frequencies to be selected for modification are in a range of $[50, 170]$ increasing in steps of 30. The expansion factors tested were 2, 4, 8, and 16. The size of the windows are $N_w = 44,032$ and the segment size is $N_s = 512$.

Assumptions.

It is assumed that the dynamic range of the audio signals has been adapted to a range in $[0, 32768]$ to avoid underflow and overflow problems, and for the use of the intDCT transform. The total payload inserted is calculated based on the number of frequencies to be modified and the number of bits embedded in each frequency, that increase as the expansion factor increases.

Results.

Table 6.5 presents the mean (μ), standard deviation (σ), minimum and maximum ODG values obtained with the different expansion factors tested. A fixed number of frequencies were selected for modification, and depending on the expansion factor, a number of bits were embedded in all the frequencies, yielding the payloads in the third column. The highest average embedding capacity that meets the transparency constraint (> -2 ODG) is of 68.8 kbps, highlighted in red. From this table, it can be observed that for a given payload (27.52 kbps) a better mean ODG result is obtained by modifying less frequencies (20) but using a higher expansion factor (16), than modifying more frequencies (80) with a lower expansion factor (4), highlighted in blue. This suggests that higher expansion factors can be used to improve the transparency of the scheme given a fixed payload to be embedded. Numbers marked in bold indicate the configurations where a payload over 24 kbps is obtained and that maintains an ODG threshold over -2, which is the payload capacity required by the self-recovery scheme.

Table 6.6 presents the mean (μ), standard deviation (σ), minimum and maximum PSNR, and MSE values obtained for the same expansion factors as those from Table 6.5. The average PSNR results corroborate the phenomenon observed in the ODG. Given a fixed payload, a higher PSNR value is obtained using a higher expansion factor; these results are highlighted in blue. The greater the PSNR value measured between two audio signals, means a greater similitude between these signals.

Discussion.

As it can be observed in Table 6.5, a payload of 27.52 kbps can be obtained by embedding either 4 or 16 bits per frequency; with the later expansion factor the ODG differs in almost -1, which suggests that better transparency results are reached when using higher expansion factors. It is also observed that the highest payload (68.8 kbps) that maintains the ODG threshold is reached with a high expansion factor (16). There are several configurations where a payload over 24 kbps is obtained and that meets the ODG threshold, which are requirements for the self-recovery scheme.

Table 6.5: ODG results obtained* with different expansion factors.

#	Exp.	Payload	ODG			
freqs.		(kbps)	μ	σ	Min	Max
20	2	3.44	0.09	0.28	-1.72	0.21
20	4	6.88	0.06	0.37	-1.73	0.20
20	8	13.76	0.04	0.37	-1.75	0.20
20	16	**27.52**	**-0.01**	0.45	-1.80	0.18
50	2	8.60	0.06	0.27	-1.76	0.16
50	4	17.20	-0.05	0.37	-1.84	0.10
50	8	**34.40**	**-0.31**	0.42	-2.07	-0.02
50	16	**68.80**	**-0.91**	0.49	-2.22	-0.41
80	2	13.76	-0.31	0.24	-1.87	-0.13
80	4	**27.52**	**-0.95**	0.31	-2.13	-0.48
80	8	55.04	-2.42	0.28	-3.04	-1.82
80	16	110.08	-3.37	0.29	-3.69	-2.31
110	2	18.92	-0.69	0.26	-2.11	-0.38
110	4	37.84	-2.29	0.38	-3.06	-1.55
110	8	75.68	-3.34	0.33	-3.75	-2.42
110	16	151.36	-3.72	0.17	-3.87	-3.09
140	2	**24.08**	**-1.44**	0.41	-2.70	-1.00
140	4	48.16	-2.93	0.50	-3.75	-1.87
140	8	96.32	-3.60	0.24	-3.86	-2.86
140	16	192.64	-3.83	0.06	-3.90	-3.64
170	2	**29.24**	**-1.99**	0.63	-3.62	-1.20
170	4	58.48	-3.20	0.49	-3.84	-2.03
170	8	116.96	-3.70	0.20	-3.89	-3.06
170	16	233.92	-3.86	0.04	-3.90	-3.75

*Results obtained from all the genres in dataset 'Ours'.

Conclusions.

The results obtained with this experimental setup suggest that higher embedding capacities can be achieved by increasing the expansion factor, while maintaining a transparency over -2 ODG. With the increase in payload, the self-recovery scheme could be tested, embedding more reference bits than with previous strategies.

Table 6.6: MSE and PSNR results obtained* with different expansion factors.

#	e_f	Payload	PSNR				MSE			
freqs.		(kbps)	μ	σ	Min	Max	μ	σ	Min	Max
20	2	3.44	77.2	8.3	58.8	90.5	9.99×10^{-08}	2.26×10^{-07}	8.85×10^{-10}	1.33×10^{-06}
20	4	6.88	68.1	8.9	49.2	83.8	8.95×10^{-07}	2.04×10^{-06}	4.21×10^{-09}	1.19×10^{-05}
20	8	13.76	60.8	9.0	41.9	76.8	4.87×10^{-06}	1.11×10^{-05}	2.08×10^{-08}	6.50×10^{-05}
20	**16**	**27.52**	**54.2**	9.0	35.3	70.3	2.24×10^{-05}	5.09×10^{-05}	9.31×10^{-08}	2.99×10^{-04}
50	2	8.60	66.5	9.5	49.8	86.8	1.01×10^{-06}	1.92×10^{-06}	2.09×10^{-09}	1.05×10^{-05}
50	4	17.20	57.1	9.8	40.3	78.4	9.08×10^{-06}	1.73×10^{-05}	1.46×10^{-08}	9.42×10^{-05}
50	8	34.40	49.7	9.8	32.9	71.2	4.94×10^{-05}	9.41×10^{-05}	7.60×10^{-08}	5.13×10^{-04}
50	16	68.80	43.1	9.8	26.3	64.6	2.27×10^{-04}	4.32×10^{-04}	3.47×10^{-07}	2.36×10^{-03}
80	2	13.76	60.6	10.4	45.0	83.5	3.98×10^{-06}	6.75×10^{-06}	4.46×10^{-09}	3.20×10^{-05}
80	**4**	**27.52**	**51.2**	10.5	35.4	74.6	3.58×10^{-05}	6.07×10^{-05}	3.48×10^{-08}	2.88×10^{-04}
80	8	55.04	43.8	10.5	28.1	67.2	1.95×10^{-04}	3.31×10^{-04}	1.89×10^{-07}	1.57×10^{-03}
80	16	110.08	37.2	10.6	21.4	60.7	8.96×10^{-04}	1.52×10^{-03}	8.57×10^{-07}	7.20×10^{-03}
110	2	18.92	55.0	11.0	39.9	79.6	1.52×10^{-05}	2.35×10^{-05}	1.10×10^{-08}	1.03×10^{-04}
110	4	37.84	45.5	11.1	30.3	70.2	1.36×10^{-04}	2.12×10^{-04}	9.58×10^{-08}	9.29×10^{-04}
110	8	75.68	38.2	11.1	23.0	62.9	7.43×10^{-04}	1.15×10^{-03}	5.12×10^{-07}	5.06×10^{-03}
110	16	151.36	31.6	11.1	16.3	56.4	3.41×10^{-03}	5.30×10^{-03}	2.32×10^{-06}	2.32×10^{-02}
140	2	24.08	50.0	11.4	36.0	75.2	4.60×10^{-05}	6.03×10^{-05}	3.04×10^{-08}	2.54×10^{-04}
140	4	48.16	40.4	11.4	26.4	65.7	4.14×10^{-04}	5.43×10^{-04}	2.68×10^{-07}	2.28×10^{-03}
140	8	96.32	33.1	11.5	19.1	58.4	2.25×10^{-03}	2.95×10^{-03}	1.46×10^{-06}	1.24×10^{-02}
140	16	192.64	26.5	11.5	12.4	51.8	1.03×10^{-02}	1.36×10^{-02}	6.68×10^{-06}	5.71×10^{-02}
170	2	29.24	46.1	10.9	32.2	70.3	1.07×10^{-04}	1.40×10^{-04}	9.39×10^{-08}	6.04×10^{-04}
170	4	58.48	36.5	11.0	22.6	60.8	9.67×10^{-04}	1.26×10^{-03}	8.35×10^{-07}	5.44×10^{-03}
170	8	116.96	29.2	11.0	15.3	53.4	5.26×10^{-03}	6.84×10^{-03}	4.54×10^{-06}	2.96×10^{-02}
170	16	233.92	22.6	11.0	8.7	46.8	2.42×10^{-02}	3.14×10^{-02}	2.08×10^{-05}	1.36×10^{-01}

*Results obtained from all the genres in dataset 'Ours'.

6.2 Audio self-recovery

A previous self-recovery strategy that inserted one bit per frequency in the intDCT domain had an embedding capacity of 16.5 kbps; with the PEE strategy modified for multi-bit embedding with an expansion factor of 4, the payload capacity can be increased to 24.8 kbps. The self-recovery strategy of 16.5 kbps allowed approximate restoration of the signals after a content replacement attack. However, since perfect restoration is required for the fulfillment of the framework, the strategy of multi-bit PEE in the intDCT domain had to be tested, in order to evaluate the restoration capabilities obtained with the increased capacity, and its corresponding increase in reference bits. This approach inserts $N_s/2+64$ control bits per segment for tamper detection and signal restoration, as opposed to the $N_s/4+64$ control bits from the previous approach.

Motivation.

The goal of this experimental setup is the evaluation of the restoration capabilities of the self-recovery scheme, when 24.8 kbps of control bits are inserted using the masking threshold properties of audio segments, and the use of multi-bit PEE. It is also necessary to verify that with the insertion of this payload the transparency threshold is maintained over -2 ODG.

Parameters.

The size of the audio windows are $N_w = 44,032$ and the size of the segments are $N_s = 512$, the segment size was reduced from that of previous experimental setups to improve the resolution for tamper detection. The number of reference bits inserted per segment was $N_s/2$, and the number of check bits inserted per segment was 64, the total control bits are embedded at a rate of 24.8 kbps. Three percentages for content-replacement attacks were tested, namely 0.3%, 0.2%, and 0.1%.

Assumptions.

It is assumed that the dynamic range of the audio signals has been adapted to a range in $[0,32768]$ to avoid underflow and overflow problems, and for the use of the intDCT transform. Since the main goal of this experimental setup is to test the restoration capabilities of the self-recovery scheme, the fragile RWS stage can be omitted. The size of the segments was reduced from the previous experiments to improve the resolution in the detection of tampering. The attacks were simulated with the content replacement attack from Algorithm 4.

Results.

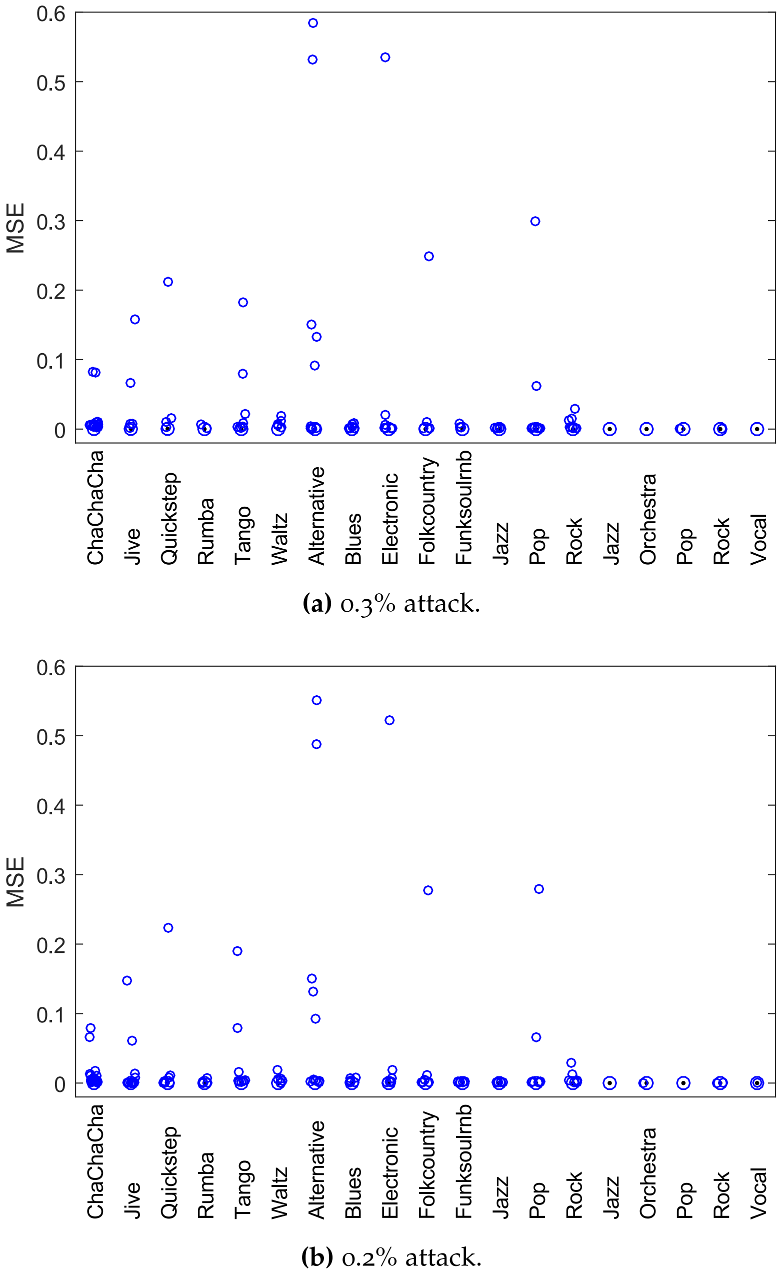
Table 6.7 presents the mean (μ), standard deviation (σ), minimum, and maximum values measured with the PSNR, MSE, and ODG between the host and watermarked audio signals. The embedded control bits are the result of the reference bits and check bits for each window of samples, and is of approximately 24.8 kbps. In this table, the numbers in bold indicate the mean ODG values over -1, which is one ODG point over the desired threshold. As it can be seen from this table, the average ODG values for most of the genres in the datasets 'MAB' and 'Ours' are over -1, indicating that the difference between the host and watermarked audio signals is indistinguishable for a human listener. And the average ODG values for all the audio signals in the three datasets are over -2 ODG, fulfilling the threshold required by the applications.

The watermarked audio signals produced by the encoding process are attacked with the simulated content replacement attack. Three percentages for the attack are used, namely 0.3%, 0.2%, and 0.1%. The PSNR, MSE, and ODG values between the host and the restored audio signals are measured to determine the quality of the restored audio signals. The distribution of MSE results for the restored audio signals of the three datasets with attacks of 0.3% and 0.2% are presented in Figures 6.1a and 6.1b, respectively. From these figures, it can be seen that most of the results are close to 0, which indicate small errors between host and restored audio signals. The PSNR distribution for the results obtained from the restored audio signals of the three datasets, for attacks of 0.3%, and 0.2% are shown in Figures 6.2a and 6.2b, respectively. Here it can be seen that the standard deviation is greater than expected, which indicates that there are cases where PSNR values are lower than 30 dB, for both 0.3%, and 0.2%. However, for both 0.3%, and 0.2% attacks, the restored PSNR values are over 30 dB for the great majority of the results, which indicate restoration with acceptable distortion.

Table 6.7: Embedding* results for the tested datasets.

Dataset	Genre	PSNR				MSE				ODG			
		μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
Ball	ChaChaCha	38.7	6.2	29.2	57.6	2.67×10^{-4}	2.62×10^{-4}	1.72×10^{-6}	1.21×10^{-3}	-1.1	0.3	-2.1	-0.4
	Jive	40.0	7.1	29.5	63.4	2.58×10^{-4}	3.14×10^{-4}	4.61×10^{-7}	1.12×10^{-3}	-1.1	0.3	-1.9	-0.5
	Quickstep	47.0	10.2	30.6	74.5	9.87×10^{-5}	1.57×10^{-4}	3.56×10^{-8}	8.67×10^{-4}	-1.0	0.4	-1.6	0.0
	Rumba	50.5	7.6	35.9	66.1	2.98×10^{-5}	5.27×10^{-5}	2.45×10^{-7}	2.55×10^{-4}	**-0.9**	0.3	-1.9	-0.4
	Tango	42.6	10.0	30.5	68.3	1.70×10^{-4}	1.75×10^{-4}	1.48×10^{-7}	8.88×10^{-4}	-1.4	0.4	-2.3	-0.6
	Waltz	48.5	11.9	35.7	72.9	6.97×10^{-5}	7.37×10^{-5}	5.13×10^{-8}	2.70×10^{-4}	-1.5	0.5	-2.2	-0.4
MAB	Alternative	50.6	10.9	28.0	74.5	8.89×10^{-5}	2.63×10^{-4}	3.53×10^{-8}	1.58×10^{-3}	**-0.6**	0.3	-1.7	0.1
	Blues	55.3	10.4	33.9	74.5	2.11×10^{-5}	5.15×10^{-5}	3.53×10^{-8}	4.03×10^{-4}	**-0.5**	0.3	-1.2	0.0
	Electronic	56.9	12.8	28.9	74.5	4.74×10^{-5}	1.62×10^{-4}	3.53×10^{-8}	1.28×10^{-3}	**-0.6**	0.4	-2.1	0.0
	Folk	54.9	11.1	31.9	74.5	2.76×10^{-5}	7.65×10^{-5}	3.51×10^{-8}	6.41×10^{-4}	**-0.6**	0.3	-1.2	0.1
	Funk	50.6	10.4	36.8	74.5	3.44×10^{-5}	4.63×10^{-5}	3.53×10^{-8}	2.10×10^{-4}	**-0.6**	0.3	-1.2	0.1
	Jazz	54.2	10.8	38.2	74.5	2.26×10^{-5}	3.66×10^{-5}	3.53×10^{-8}	1.53×10^{-4}	**-0.7**	0.3	-1.6	-0.2
	Pop	49.8	9.7	29.9	73.9	6.49×10^{-5}	1.74×10^{-4}	4.09×10^{-8}	1.02×10^{-3}	**-0.6**	0.3	-1.2	0.1
	Rock	50.7	11.7	34.1	74.5	4.31×10^{-5}	6.74×10^{-5}	3.53×10^{-8}	3.86×10^{-4}	**-0.5**	0.3	-1.2	0.1
Ours	Jazz	50.5	5.2	43.6	59.1	1.45×10^{-5}	1.27×10^{-5}	1.24×10^{-6}	4.32×10^{-5}	**-0.8**	0.2	-1.2	-0.5
	Orchestra	66.8	10.2	48.1	74.5	2.30×10^{-6}	4.94×10^{-6}	3.58×10^{-8}	1.55×10^{-5}	-1.3	0.5	-2.1	-0.7
	Pop	45.8	5.0	35.4	54.0	5.34×10^{-5}	8.52×10^{-5}	4.02×10^{-6}	2.88×10^{-4}	**-0.9**	0.2	-1.2	-0.7
	Rock	41.1	3.1	35.7	46.8	9.86×10^{-5}	7.30×10^{-5}	2.08×10^{-5}	2.66×10^{-4}	**-0.9**	0.1	-1.1	-0.7
	Vocal	51.8	4.4	44.6	58.4	1.03×10^{-5}	1.04×10^{-5}	1.43×10^{-6}	3.46×10^{-5}	**-0.8**	0.2	-1.4	-0.6

*Embedded control bits = 24.8 kbps

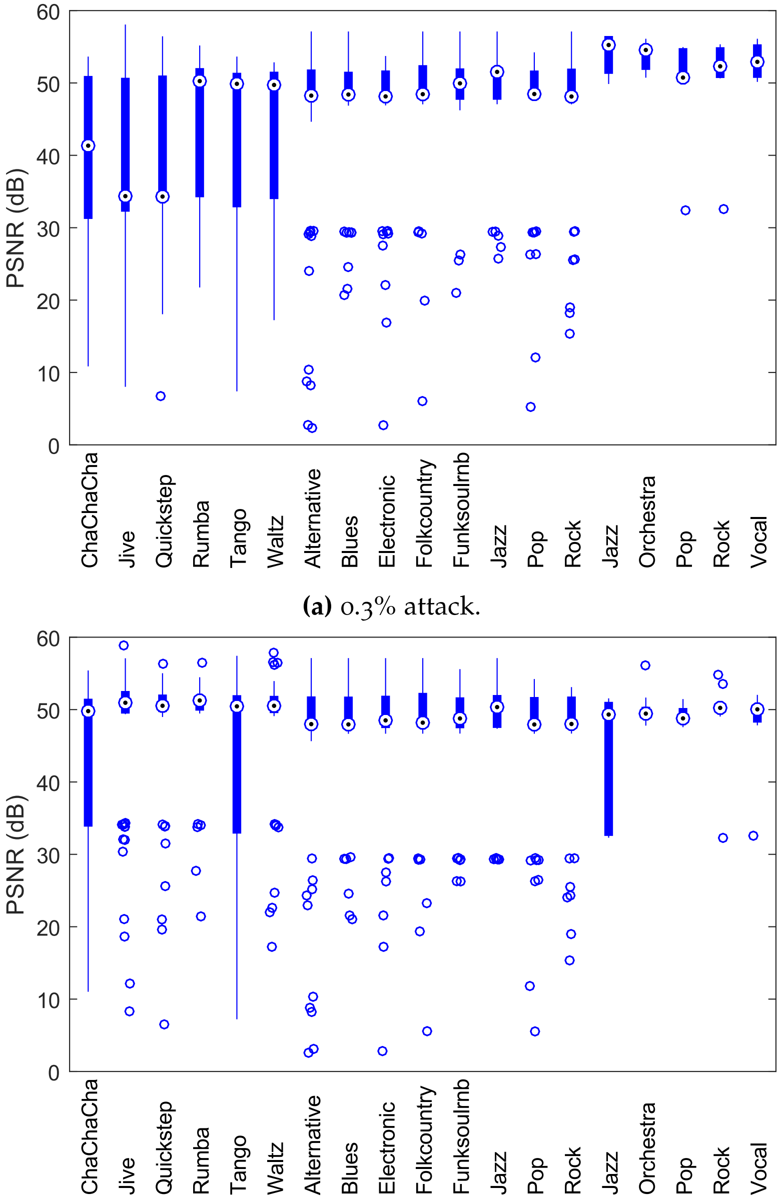


(a) 0.3% attack.

(b) 0.2% attack.

Figure 6.1: MSE results for three datasets.

Mean (μ), standard deviation (σ), minimum, and maximum values of the ODG for the attacked and restored audio signals with 0.3%, and 0.2% are presented in Table 6.10. It can be observed that the average ODG values for the attacked audio signals is close to -4 for all the genres in the datasets, indicating annoying distortion in the attacked signals. The mean ODG values for the restored signals highlighted in bold blue, indicate the ODG values over -1. They indicate that the restored signals are



(a) 0.3% attack.

(b) 0.2% attack.

Figure 6.2: PSNR (dB) results for three datasets.

very similar to the host ones, and the difference between host and restored are almost unnoticeable. As it can be seen, most of the genres for the datasets 'Ball' and 'Ours' result in signals with high similitude to the host ones for both 0.3%, and 0.2% attacks. For the dataset 'MAB', it can be observed that the ODG results for all genres are over -2, which indicate perceptible but not annoying differences between host and restored signals; this occurs for both 0.3%, and 0.2% attacks. The average ODG values in bold red indicate the cases where the ODG $\leqslant$ -2, and indicate that the differences between host and restored signals are slightly annoying. This occurs for the 'orchestra' genre, where signals are low energy ones. The attacks are very notorious because there is a great contrast between the random noise from the attack and the low energy samples from the rest of the signal. Although the scheme is capable of restoring certain samples in terms of their time domain values, these restored samples still have a notorious contrast against the non-attacked low energy regions. Despite this, most of the results are over -2 ODG, and indicate that the quality of the restored audio signals is adequate for speech restoration and music distribution applications.

Although the PSNR and ODG results for the 0.3% and 0.2% attacks might seem contradictory, what occurs is the following. The PSNR results from Figure 6.2 are not as high as expected, because the restored samples are not as similar to the original samples in terms of their time-domain values, *i.e.*, their numerical values are different. On the other hand, the ODG values from Table 6.10 indicate restoration with adequate distortion for the target applications. This means that, despite the fact that the numerical sample values are dissimilar, the perceived quality of the restored signals is adequate.

The mean (μ), standard deviation (σ), minimum, and maximum values of the PSNR, and MSE for the three datasets with a 0.1% attack are presented in Table 6.8. As it can be seen in this table, with this percentage of attack the scheme achieves perfect restoration in all of the genres of the three datasets, as indicated by the bold blue values in the minimum MSE column. An MSE value of 0 means that there is no error between host and restored audio signals, *i.e.*, perfect restoration is obtained. It can also be seen that for the dataset 'Ours', perfect restoration is achieved for all the audio signals in all the genres, except for one audio signal in the 'orchestra' genre.

The mean (μ), standard deviation (σ), minimum, and maximum values of the ODG for the attacked and restored audio signals are presented in Table 6.9. From this table, it can be seen that the average ODG values for the attacked audio signals are close to -4 for all the genres in the datasets. A value of -4 ODG indicates very annoying distortion,

Table 6.8: PSNR and MSE results of restored signals attacked with 0.1%.

Dataset	Genre	PSNR				MSE			
		μ	σ	Min	Max	μ	σ	Min	Max
Ball	ChaChaCha	28.8	8.6	10.9	53.8	2.08×10^{-3}	1.00×10^{-2}	**0.00×10^{0}**	8.09×10^{-2}
	Jive	28.6	11.2	7.1	51.7	4.47×10^{-3}	2.58×10^{-2}	**0.00×10^{0}**	1.93×10^{-1}
	Quickstep	21.9	9.3	6.4	34.2	4.23×10^{-3}	2.93×10^{-2}	**0.00×10^{0}**	2.27×10^{-1}
	Rumba	32.0	2.1	30.3	34.4	8.59×10^{-5}	2.48×10^{-4}	**0.00×10^{0}**	9.24×10^{-4}
	Tango	28.5	14.3	8.3	53.7	7.37×10^{-3}	2.78×10^{-2}	**0.00×10^{0}**	1.47×10^{-1}
	Waltz	28.1	13.1	16.4	53.3	1.08×10^{-3}	4.01×10^{-3}	**0.00×10^{0}**	2.31×10^{-2}
MAB	Alternative	23.9	19.3	2.4	48.8	1.58×10^{-2}	7.89×10^{-2}	**0.00×10^{0}**	5.70×10^{-1}
	Blues	21.3	1.1	20.1	22.2	2.64×10^{-4}	1.43×10^{-3}	**0.00×10^{0}**	9.76×10^{-3}
	Electronic	18.9	11.2	3.1	29.1	6.09×10^{-3}	5.37×10^{-2}	**0.00×10^{0}**	4.89×10^{-1}
	Folk	28.9	22.4	6.2	51.0	2.86×10^{-3}	2.62×10^{-2}	**0.00×10^{0}**	2.42×10^{-1}
	Funk	26.8	3.3	22.3	29.3	2.41×10^{-4}	9.69×10^{-4}	**0.00×10^{0}**	5.87×10^{-3}
	Jazz	34.0	11.3	27.4	47.0	8.20×10^{-5}	3.78×10^{-4}	**0.00×10^{0}**	1.82×10^{-3}
	Pop	25.7	16.8	5.4	49.5	7.29×10^{-3}	4.31×10^{-2}	**0.00×10^{0}**	2.85×10^{-1}
	Rock	27.0	14.1	15.2	51.2	1.13×10^{-3}	4.93×10^{-3}	**0.00×10^{0}**	3.03×10^{-2}
Ours	Jazz	***PR**	**PR**	**PR**	**PR**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**
	Orchestra	**PR**	**PR**	30.4	**PR**	9.02×10^{-5}	2.85×10^{-4}	**0.00×10^{0}**	9.02×10^{-4}
	Pop	**PR**	**PR**	**PR**	**PR**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**
	Rock	**PR**	**PR**	**PR**	**PR**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**
	Vocal	**PR**	**PR**	**PR**	**PR**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**

*PR = Perfect Restoration

which means that the attack is severe despite the percentage used. In this table, the mean ODG values in bold blue indicate values equivalent to perfect restoration (ODG $\geqslant 0$).

As it can be seen, these ODG results are consistent with the PSNR and MSE results from Table 6.8, which demonstrate perfect restoration capabilities. From all the audio signals evaluated in the three datasets, perfect restoration is achieved for 87.3% of the signals, and with the remaining 12.7% signals, approximate restoration is achieved for signals attacked with 0.1%.

Discussion.

The restoration capabilities of the proposed self-recovery scheme indicate that the scheme has effective restoration capabilities up to the tested 0.3% of content replace-

Table 6.9: ODG results of attacked and restored signals for 0.1% attack.

Dataset	Genre	Attacked				Restored			
		μ	σ	Min	Max	μ	σ	Min	Max
Ball	ChaChaCha	-2.8	0.6	-3.8	-1.3	-0.3	0.9	-3.6	0.2
	Jive	-2.7	0.6	-3.7	-1.6	-0.1	0.7	-3.1	0.2
	Quickstep	-3.1	0.6	-3.8	-1.3	**0.0**	0.6	-2.5	0.2
	Rumba	-3.4	0.4	-3.8	-1.9	**0.1**	0.2	-0.6	0.2
	Tango	-3.3	0.4	-3.8	-2.4	-0.3	0.9	-2.7	0.2
	Waltz	-3.5	0.3	-3.9	-2.7	**0.0**	0.5	-2.6	0.2
MAB	Alternative	-2.7	0.9	-3.9	-0.9	**0.0**	0.7	-3.8	0.2
	Blues	-3.0	0.6	-3.9	-1.4	**0.1**	0.5	-3.2	0.2
	Electronic	-3.0	0.8	-3.9	-1.0	**0.1**	0.6	-3.9	0.2
	Folk	-3.0	0.7	-3.8	-1.3	**0.2**	0.3	-1.5	0.2
	Funk	-2.7	0.8	-3.8	-0.7	**0.0**	0.7	-3.1	0.2
	Jazz	-3.1	0.7	-3.9	-1.1	**0.1**	0.4	-1.6	0.2
	Pop	-2.6	0.9	-3.8	-0.5	**0.0**	0.8	-3.8	0.2
	Rock	-2.2	0.8	-3.8	-0.7	**0.0**	0.7	-3.2	0.2
Ours	Jazz	-3.4	0.4	-3.8	-2.5	**0.2**	0.0	0.2	0.2
	Orchestra	-3.8	0.1	-3.9	-3.5	**0.1**	0.3	-0.8	0.2
	Pop	-3.6	0.2	-3.9	-3.4	**0.2**	0.0	0.2	0.2
	Rock	-2.1	0.5	-2.9	-1.5	**0.2**	0.0	0.2	0.2
	Vocal	-3.5	0.3	-3.9	-2.9	**0.2**	0.0	0.2	0.2

ment. For 0.3%, and 0.2% attacks, the quality of the restored signals is adequate for the target applications. Furthermore, for 0.1% attack, perfect restoration is achieved in 87.3% of the audio signals tested. From the results obtained, it can be appreciated that some practical applications would require a greater percentage of restoration than the current one.

Some strategies have to be further explored to increase the percentage of tampered samples that the scheme can restore, and to obtain perfect restoration for 100% of the audio signals in the datasets. To improve the restoration capabilities, the payload of the scheme could be increased to allow the insertion of more reference bits, to do so, the expansion factor could be increased, and a balance between expansion factor and transparency should be determined to increase embedding capacity and transparency. Another strategy to improve restoration is the use of periodic spreading of the reference and check bits, instead of the pseudo-random spreading. The scheme satisfies the desired transparency threshold and provides effective restoration capabilities.

Conclusions.

With a content replacement attack of 0.1%, the scheme achieves perfect restoration for 87.3% of the signals from the three tested datasets. Since the requirement to construct the framework for audio signals is perfect restoration in the self-recovery stage, and given that the results obtained show that this property had been achieved, the framework could be completed.

6.3 Framework for audio

The implementation of the framework for images proved to be effective in terms of watermark and signal robustness, these results are presented in Appendix A. A straightforward supposition from this fact was that the same framework can be followed to propose a solution for audio signals. The requirement for such a solution is a self-recovery scheme for audio with perfect restoration. The self-recovery scheme in the intDCT domain with multi-bit PEE embedding proved to achieve perfect restoration for content replacement attacks of 0.1%. With this scheme, the framework for audio can be implemented and tested, to validate that the requirements of transparency and robustness are meet.

Motivation.

The goal of this experimental setup is to evaluate the audio framework, in order to verify that the transparency threshold is maintained over -2 ODG. The restoration capabilities of the framework are also tested, to verify that for a content replacement attack of 0.1% perfect signal restoration and perfect watermark extraction are achieved.

Parameters.

The size of the audio windows is $N_w = 44,032$ and the segment size is $N_s = 512$, which was kept with that value to maintain the resolution for tamper detection. The rate of control bits embedded with the self-recovery stage is 24.8 kbps, and the payload embedded with the fragile reversible watermarking stage is 2 kbps. The attacks were of 0.1%.

Assumptions.

It is assumed that the dynamic range of the audio signals has been adapted to a range in $[0, 32768]$ to avoid underflow and overflow problems, and for the use of the intDCT transform. The attacks were simulated with the content replacement attack from Algorithm 4. The watermark inserted in the fragile reversible stage was generated randomly to simulate a watermark constructed from a secret message to be embedded into the signals.

Results.

The first part of this experimental setup is to verify that the transparency of the complete framework remains over the threshold. Table 6.11 presents the mean (μ), standard deviation (σ), minimum and maximum ODG results obtained from the two stages of the framework. A host audio signal is fed to the encoding process, where first the fragile RWS stage inserts the watermark of 2 kbps into the signal, generation a first-stage watermarked (FSW) audio signal; then this first-stage watermarked signal is passed to the self-recovery stage, where reference and check bits are calculated and inserted within, to protect the signal from content replacement attack, generating a second-stage watermarked (SSW) audio signal.

As it can be seen in Table 6.11, the mean ODG values for the second-stage watermarked signals, *i.e.* the degradation of the complete encoding process of the framework, is over -2 ODG for all the tested signals in the three datasets. The ODG values marked in bold blue indicate the higher average ODG results obtained with the encoding process, which are -1 ODG and indicate that the distortion caused is not annoying. From Table 6.11 it can also be seen that for almost all the genres from the three datasets, the average ODG values from the fragile RWS stage, marked in bold, are over 0 ODG, which means that the insertion of 2 kbps in the intDCT domain using a multi-bit PEE strategy generates watermarked signals with unnoticeable distortion.

The second part of this experimental setup is to verify the robustness of the framework. For this, the watermarked audio signals were subjected to a content replacement attack of 0.1% simulated with Algorithm 4, which generates an attacked audio signal. That attacked signal is then passed to the decoding process of the framework, where the self-recovery stage counteracts the modifications caused by the attack, generating a first-stage watermarked (FSW) audio signal. Then, the first-stage watermarked signal is passed to the fragile RWS stage, where the original samples are restored and the watermark is extracted.

Table 6.12 presents the mean (μ), standard deviation (σ), minimum and maximum PSNR values obtained in the two stages of the decoding process, *i.e.* the PSNR measured between the FSW audio signal and the host signal, and the PSNR measured between the final restored audio signal and the host audio signal. In the datasets

'BALL', and 'MAB' the PSNR results for perfect restoration are not reported, only the PSNR results with numeric values are included. For the dataset 'Ours' the PSNR for perfect restoration reported are because all the audio signals from these genres, namely jazz, rock, and vocal, had perfect restoration results. Table 6.13 presents the mean (μ), standard deviation (σ), minimum and maximum MSE values obtained from the two stages of the decoding process of the framework. As it can be seen from this table, there are cases of perfect restoration for all the datasets, which are marked in bold blue in the column of minimum MSE values; and for all the MSE values of the genres jazz, rock, and vocal of the dataset 'Ours'. These MSE results are consistent with the PSNR results from Table 6.12.

Table 6.14 presents the mean (μ), standard deviation (σ), minimum and maximum ODG values measured for the two stages of the decoding process of the framework, and the BER results obtained between the extracted and original watermarks. The average ODG results for all the datasets are above -2 ODG, which indicates restoration with acceptable perceptual quality; the results marked in bold indicate those where ODG values are over -1 ODG, which means that, although in terms of perceptual quality the restored audio signals have audible artifacts, the restoration quality is high. Since all the average ODG values are over -2 ODG, this means that the framework has signal robustness which is acceptable for most practical applications. ODG results marked in bold blue indicate those results equivalent to perfect restoration, which occurs for at least one genre in every tested dataset. The watermark extraction results from Table 6.14, given by the BER and marked in bold blue, indicate perfect extraction for all the watermarks in the restored audio signals.

Table 6.10: ODG results of attacked and restored signals for 0.3%, and 0.2% attacks.

%	Dataset	Genre	Attacked				Restored			
			µ	σ	Min	Max	µ	σ	Min	Max
0.3	Ball	ChaChaCha	-3.1	0.5	-3.7	-1.8	-1.0	0.7	-3.7	-0.2
		Jive	-3.1	0.5	-3.8	-1.6	**-0.9**	0.7	-3.5	-0.1
		Quickstep	-3.3	0.5	-3.8	-1.7	**-0.9**	0.5	-2.6	-0.1
		Rumba	-3.5	0.3	-3.9	-2.5	**-0.9**	0.5	-2.6	-0.3
		Tango	-3.4	0.3	-3.8	-2.5	-1.0	0.6	-2.5	-0.4
		Waltz	-3.5	0.3	-3.9	-2.7	-1.1	0.9	-3.6	-0.3
	MAB	Alternative	-3.0	0.6	-3.8	-1.0	-1.4	0.7	-3.8	-0.2
		Blues	-3.2	0.4	-3.8	-1.7	-1.7	0.6	-3.5	-0.5
		Electronic	-3.3	0.6	-3.9	-1.3	-1.8	0.9	-3.9	0.2
		Folk	-3.2	0.6	-3.8	-1.5	-1.5	0.8	-3.5	0.2
		Funk	-3.1	0.5	-3.9	-1.7	-1.3	0.8	-3.6	0.2
		Jazz	-3.4	0.4	-3.9	-1.9	-1.7	0.3	-3.8	0.2
		Pop	-3.0	0.6	-3.8	-1.2	-1.4	0.8	-3.8	0.2
		Rock	-2.8	0.6	-3.9	-1.6	-1.4	0.8	-3.8	0.2
	Ours	Jazz	-3.5	0.2	-3.8	-3.1	**-0.4**	0.4	-0.8	0.2
		Orchestra	-3.9	0.1	-3.9	-3.7	**-2.0**	1.6	-3.8	0.2
		Pop	-3.6	0.2	-3.9	-3.1	**-0.7**	0.3	-1.3	-0.3
		Rock	-2.5	0.5	-3.2	-1.9	**-0.5**	0.1	-0.6	-0.4
		Vocal	-3.6	0.3	-3.9	-3.1	**-0.6**	0.4	-1.1	0.2
0.2	Ball	ChaChaCha	-3.0	0.6	-3.7	-1.4	-1.0	0.7	-3.6	-0.2
		Jive	-2.9	0.6	-3.8	-1.5	**-0.9**	0.6	-3.6	-0.3
		Quickstep	-3.2	0.5	-3.8	-1.3	**-0.9**	0.5	-2.8	-0.4
		Rumba	-3.5	0.3	-3.9	-2.5	**-0.8**	0.4	-2.6	-0.4
		Tango	-3.4	0.4	-3.9	-2.4	-1.0	0.6	-2.5	-0.3
		Waltz	-3.5	0.3	-3.9	-2.6	**-0.9**	0.9	-3.7	-0.2
	MAB	Alternative	-2.9	0.8	-3.8	-1.1	-1.4	0.6	-3.8	-0.2
		Blues	-3.2	0.5	-3.8	-1.4	-1.6	0.6	-3.6	-0.5
		Electronic	-3.2	0.6	-3.9	-1.5	-1.7	0.9	-3.9	0.2
		Folk	-3.1	0.6	-3.8	-1.4	-1.4	0.7	-3.4	0.2
		Funk	-3.1	0.6	-3.9	-1.4	-1.4	0.6	-3.6	-0.3
		Jazz	-3.3	0.4	-3.9	-2.4	-1.6	0.9	-3.9	-0.6
		Pop	-3.0	0.7	-3.8	-1.1	-1.4	0.7	-3.9	0.2
		Rock	-2.8	0.6	-3.9	-1.4	-1.3	0.8	-3.8	-0.2
	Ours	Jazz	-3.5	0.3	-3.8	-2.6	**-0.7**	0.3	-1.1	-0.3
		Orchestra	-3.8	0.1	-3.9	-3.7	**-2.4**	1.3	-3.8	-0.7
		Pop	-3.7	0.1	-3.8	-3.5	**-0.7**	0.4	-1.4	-0.1
		Rock	-2.5	0.4	-3.2	-1.9	**-0.5**	0.1	-0.8	-0.2
		Vocal	-3.5	0.3	-3.8	-2.8	**-0.7**	0.3	-1.3	-0.2

Table 6.11: ODG results for encoding in the three datasets.

Dataset	Genre	*FSW audio				SSW audio			
		μ	σ	Min	Max	μ	σ	Min	Max
BALL	ChaChaCha	**0.2**	0.0	0.1	0.2	**-1.0**	0.3	-2.2	-0.4
	Jive	**0.2**	0.0	0.1	0.2	**-1.0**	0.3	-1.8	-0.3
	Quickstep	**0.2**	0.0	0.1	0.2	**-1.0**	0.3	-1.5	-0.4
	Rumba	**0.1**	0.0	0.0	0.2	-1.2	0.3	-2.0	-0.7
	Tango	**0.1**	0.1	-0.3	0.2	**-1.0**	0.3	-2.2	-0.5
	Waltz	**0.1**	0.1	-0.3	0.2	-1.2	0.4	-2.3	-0.3
MAB	Alternative	**0.2**	0.0	0.1	0.2	-1.4	0.5	-2.9	0.0
	Blues	**0.2**	0.0	0.0	0.2	-1.4	0.5	-3.2	-0.5
	Electronic	**0.1**	0.1	-0.2	0.2	-1.5	0.5	-3.0	-0.4
	Folk	**0.2**	0.0	-0.2	0.2	-1.4	0.5	-2.9	-0.3
	Funk	**0.2**	0.0	0.0	0.2	-1.4	0.6	-3.1	-0.5
	Jazz	**0.1**	0.2	-1.7	0.2	-1.4	0.6	-3.4	-0.2
	Pop	**0.2**	0.0	0.1	0.2	-1.4	0.5	-2.7	-0.3
	Rock	**0.2**	0.0	0.0	0.2	-1.4	0.5	-3.1	-0.3
Ours	Jazz	**0.1**	0.0	0.1	0.2	-1.3	0.3	-1.7	-0.9
	Orchestra	-0.3	0.7	-1.7	0.2	-1.3	0.5	-2.0	-0.6
	Pop	**0.1**	0.0	0.0	0.2	-1.6	0.2	-2.0	-1.2
	Rock	**0.2**	0.0	0.2	0.2	-1.5	0.2	-1.9	-1.2
	Vocal	**0.2**	0.0	0.1	0.2	-1.3	0.4	-1.9	-0.9

*Payload = 2 kbps

Table 6.12: PSNR restoration results for the complete framework in the three datasets.

Dataset	Genre	FSW audio				Restored audio			
		μ	σ	Min	Max	μ	σ	Min	Max
BALL	ChaChaCha	48.5	33.9	2.5	87.0	18.3	12.1	2.6	53.7
	Jive	54.6	35.6	2.9	87.0	12.3	9.6	2.9	33.7
	Quickstep	68.8	28.4	7.0	92.1	21.5	11.8	7.0	55.1
	Rumba	78.6	18.9	28.6	87.0	31.1	2.6	28.7	34.0
	Tango	63.4	30.3	8.5	87.1	26.8	25.1	8.6	151.4
	Waltz	72.7	24.2	13.2	92.1	24.1	9.9	13.2	53.2
MAB	Alternative	76.4	18.3	2.5	82.4	23.5	18.8	2.5	52.5
	Blues	80.0	10.7	19.5	82.4	31.2	14.4	19.7	47.5
	Electronic	76.6	18.5	3.3	82.4	19.3	12.9	3.4	47.0
	Folk	79.4	13.3	3.5	82.4	24.3	18.3	3.5	51.1
	Funk	78.1	14.3	21.3	82.4	25.3	3.9	21.5	29.3
	Jazz	80.5	10.3	3.9	82.4	25.4	14.6	3.9	48.4
	Pop	77.8	15.6	5.3	82.4	26.6	16.1	5.3	50.7
	Rock	78.1	14.1	3.1	82.5	29.1	14.6	3.1	53.6
Ours	Jazz	85.6	3.2	76.8	87.0	***PR**	**PR**	**PR**	**PR**
	Orchestra	80.7	17.1	32.2	87.0	32.3	0	32.3	32.3
	Pop	66.2	13.5	32.1	78.4	32.2	0	32.2	32.2
	Rock	71.7	10.7	58.0	85.6	**PR**	**PR**	**PR**	**PR**
	Vocal	77.9	6.3	65.1	86.6	**PR**	**PR**	**PR**	**PR**

*PR = Perfect restoration

Table 6.13: MSE restoration results for the complete framework in the three datasets.

Dataset	Genre	FSW audio				Restored audio			
		μ	σ	**Min**	**Max**	μ	σ	**Min**	**Max**
BALL	ChaChaCha	5.85×10^{-2}	1.26×10^{-1}	1.98×10^{-09}	5.57×10^{-1}	5.81×10^{-2}	1.25×10^{-1}	**0.00×10^{0}**	5.54×10^{-1}
	Jive	7.21×10^{-2}	1.33×10^{-1}	2.01×10^{-09}	5.10×10^{-1}	7.16×10^{-2}	1.32×10^{-1}	**0.00×10^{0}**	5.07×10^{-1}
	Quickstep	1.02×10^{-2}	3.42×10^{-2}	6.20×10^{-10}	2.00×10^{-1}	1.01×10^{-2}	3.40×10^{-2}	**0.00×10^{0}**	1.99×10^{-1}
	Rumba	1.12×10^{-4}	3.36×10^{-4}	1.97×10^{-09}	1.37×10^{-3}	1.10×10^{-4}	3.28×10^{-4}	**0.00×10^{0}**	1.34×10^{-3}
	Tango	6.74×10^{-3}	2.16×10^{-2}	1.95×10^{-09}	1.40×10^{-1}	6.67×10^{-3}	2.15×10^{-2}	**0.00×10^{0}**	1.39×10^{-1}
	Waltz	2.02×10^{-3}	6.85×10^{-3}	6.19×10^{-10}	4.77×10^{-2}	1.99×10^{-3}	6.79×10^{-3}	**0.00×10^{0}**	4.75×10^{-2}
MAB	Alternative	1.15×10^{-2}	6.67×10^{-2}	5.71×10^{-09}	5.61×10^{-1}	1.14×10^{-2}	6.63×10^{-2}	**0.00×10^{0}**	5.59×10^{-1}
	Blues	2.30×10^{-4}	1.50×10^{-3}	5.73×10^{-09}	1.11×10^{-2}	2.18×10^{-4}	1.42×10^{-3}	**0.00×10^{0}**	1.07×10^{-2}
	Electronic	7.56×10^{-3}	4.70×10^{-2}	5.71×10^{-09}	4.68×10^{-1}	7.41×10^{-3}	4.61×10^{-2}	**0.00×10^{0}**	4.59×10^{-1}
	Folk	4.73×10^{-3}	3.86×10^{-2}	5.72×10^{-09}	4.52×10^{-1}	4.66×10^{-3}	3.82×10^{-2}	**0.00×10^{0}**	4.48×10^{-1}
	Funk	2.59×10^{-4}	1.19×10^{-3}	5.72×10^{-09}	7.34×10^{-3}	2.46×10^{-4}	1.13×10^{-3}	**0.00×10^{0}**	7.00×10^{-3}
	Jazz	2.07×10^{-3}	2.49×10^{-2}	5.69×10^{-09}	4.05×10^{-1}	2.05×10^{-3}	2.47×10^{-2}	**0.00×10^{0}**	4.04×10^{-1}
	Pop	4.78×10^{-3}	3.42×10^{-2}	5.69×10^{-09}	2.96×10^{-1}	4.70×10^{-3}	3.38×10^{-2}	**0.00×10^{0}**	2.94×10^{-1}
	Rock	2.43×10^{-3}	2.73×10^{-2}	5.69×10^{-09}	4.87×10^{-1}	2.40×10^{-3}	2.71×10^{-2}	**0.00×10^{0}**	4.85×10^{-1}
Ours	Jazz	4.11×10^{-9}	5.99×10^{-9}	2.01×10^{-09}	2.11×10^{-8}	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**
	Orchestra	6.01×10^{-5}	1.90×10^{-4}	2.00×10^{-09}	6.01×10^{-4}	5.87×10^{-5}	1.86×10^{-4}	**0.00×10^{0}**	5.87×10^{-4}
	Pop	6.18×10^{-5}	1.95×10^{-4}	1.45×10^{-08}	6.15×10^{-4}	6.08×10^{-5}	1.92×10^{-4}	**0.00×10^{0}**	6.08×10^{-4}
	Rock	3.83×10^{-7}	5.27×10^{-7}	2.75×10^{-09}	1.59×10^{-6}	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**
	Vocal	4.69×10^{-8}	9.36×10^{-8}	2.19×10^{-09}	3.11×10^{-7}	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**	**0.00×10^{0}**

Table 6.14: ODG and BER for the complete framework in the three datasets.

Dataset	Genre	FSW audio				Restored audio							
		ODG				ODG				*BER			
		μ	σ	Min	Max	μ	σ	Min	Max	μ	σ	Min	Max
BALL	ChaChaCha	-1.3	1.0	-3.9	-0.2	-1.2	1.0	-3.9	-0.1	**0.0**	0.1	0.0	0.3
	Jive	-1.2	0.8	-3.7	-0.4	-1.0	0.8	-3.5	-0.3	**0.0**	0.1	0.0	0.3
	Quickstep	-1.0	0.9	-3.5	0.2	**-0.8**	0.8	-3.4	**0.2**	**0.0**	0.0	0.0	0.1
	Rumba	-0.9	0.4	-3.0	-0.5	**-0.5**	0.2	-0.9	-0.1	**0.0**	0.0	0.0	0.0
	Tango	-1.2	0.8	-3.5	0.2	**-0.8**	0.6	-2.7	**0.2**	**0.0**	0.0	0.0	0.1
	Waltz	-1.0	0.7	-3.5	0.2	**-0.6**	0.5	-3.2	**0.2**	**0.0**	0.0	0.0	0.0
MAB	Alternative	-0.9	0.6	-3.8	0.2	**-0.8**	0.6	-3.8	**0.2**	**0.0**	0.0	0.0	0.3
	Blues	-0.9	0.5	-3.6	-0.2	**-0.9**	0.5	-3.5	-0.4	**0.0**	0.0	0.0	0.0
	Electronic	-1.0	0.7	-3.9	-0.3	-1.0	0.6	-3.9	-0.4	**0.0**	0.0	0.0	0.2
	Folk	-0.8	0.4	-2.8	0.0	**-0.8**	0.4	-2.8	**0.1**	**0.0**	0.0	0.0	0.3
	Funk	-1.0	0.7	-3.4	-0.5	-1.0	0.7	-3.4	-0.5	**0.0**	0.0	0.0	0.0
	Jazz	-0.9	0.5	-3.2	-0.4	**-0.9**	0.5	-3.1	-0.4	**0.0**	0.0	0.0	0.2
	Pop	-0.8	0.4	-3.8	-0.4	**-0.8**	0.4	-3.8	-0.4	**0.0**	0.0	0.0	0.2
	Rock	-0.8	0.5	-3.8	-0.3	**-0.7**	0.5	-3.7	-0.3	**0.0**	0.0	0.0	0.3
Ours	Jazz	-0.7	0.1	-0.8	-0.5	**-0.4**	0.1	-0.6	-0.3	**0.0**	0.0	0.0	0.0
	Orchestra	-0.9	0.5	-2.1	-0.3	**-0.8**	0.4	-2.0	-0.6	**0.0**	0.0	0.0	0.0
	Pop	-1.0	0.8	-3.3	-0.5	**-0.7**	0.5	-1.8	-0.2	**0.0**	0.0	0.0	0.0
	Rock	-0.4	0.2	-0.6	-0.1	**-0.2**	0.2	-0.6	**0.0**	**0.0**	0.0	0.0	0.0
	Vocal	-0.7	0.2	-0.9	-0.3	**-0.5**	0.2	-0.7	-0.3	**0.0**	0.0	0.0	0.0

*Payload = 2 kbps

Discussion.

The extraction results suggest that the watermarks can be extracted even when the signals restored with the framework are not exactly the same as the ones watermarked in the encoding process. This phenomenon in the BER results suggests that the use of the intDCT domain not only improves the transparency of the reversible watermarking scheme, but it also increases the robustness of the watermark. A fragile reversible watermarking scheme would not be able to extract the watermark with low BER values, if the signals are different from those in the encoding process. However, the capacity of the proposed reversible watermarking scheme to extract the watermarks even when the signals differ, suggests that the scheme is semi-fragile instead of fragile. The watermark robustness of the framework has to be further explored to determine the robustness limits against content replacement attacks and other attacks.

Conclusions.

The transparency results for the encoding process of the framework indicate that the proposed strategies allow the insertion of the watermarks and the control bits required for restoration in an imperceptible manner, and that the transparency threshold is maintained. The restoration results obtained from the decoding process of the framework indicate that there is perfect restoration for audio signals in all the tested datasets, although there are cases where perfect restoration is not achieved. Perfect restoration of the complete framework, *i.e.* the reversibility property of the framework is dependent on the restoration capabilities of the self-recovery stage. When the self-recovery stage is capable of completely restoring the modifications caused by the content replacement attack, the whole construction can guarantee reversibility. However, in the cases where reversibility is not achieved, the perceptual quality of the signals restored with the framework are over -2 ODG for all the tested signals, which means that the restoration capabilities are acceptable for most practical applications in terms of audio quality.

The watermark extraction results suggest that the proposed reversible watermarking scheme is semi-fragile and tolerates certain distortions caused by a content replacement attack. Robustness of the proposed reversible watermarking scheme has to be further explored to identify its limitations.

6.3.1 Effect of embedded payload on watermark and signal robustness

The previous results obtained with the framework for audio, showed that it maintains adequate transparency, and that it has watermark and signal robustness when the payload inserted in the RWS stage is 2 kbps. After that, an analysis was made to identify the influence that the embedded payload has on the extraction of the watermark, and the restoration quality of the signals.

Motivation.

The goal of this experimental setup is to analyze the watermark and signal robustness of the framework when various payloads are inserted in the RWS stage.

Parameters.

The size of the audio windows is $N_w = 44,032$ and the segment size is $N_s = 512$, the rate of control bits embedded is 24.8 kbps, the content replacement attack applied to the watermarked signals is 0.1%, and the payloads tested are 0.5, 1, 1.5, 2, 5, 7, 10, and 13 kbps. The dataset used for these experiments is ´Ours'.

Assumptions.

It is assumed that the dynamic range of the audio signals has been adapted to a range in $[0, 32768]$ to avoid underflow and overflow problems, and for the use of the intDCT transform. The attacks were simulated with the content replacement attack from Algorithm 4. The watermark inserted in the fragile reversible stage was generated randomly to simulate a watermark constructed from a secret message to be embedded into the signals.

Results.

The complete framework was applied to all the audio signals from the dataset ´Ours', and the restored signals obtained from the decoding process of the framework were compared against the host audio signals, varying the payload rates embedded in the encoding process. Figure 6.3 presents the different distributions of the PSNR values obtained for all the signals in the dataset, at the various payloads; it does not include PSNR results of perfect restoration. When the payload is under 2 kbps, the PSNR results do not vary much and have a mean value of approximately 32 dB. As the payload increases, the PSNR results show a decrease in quality, as it is expected since more information is inserted. It can be observed that these results have a greater variance when payloads at higher rates are inserted, which means that restoration results are not always stable at high embedding rates.

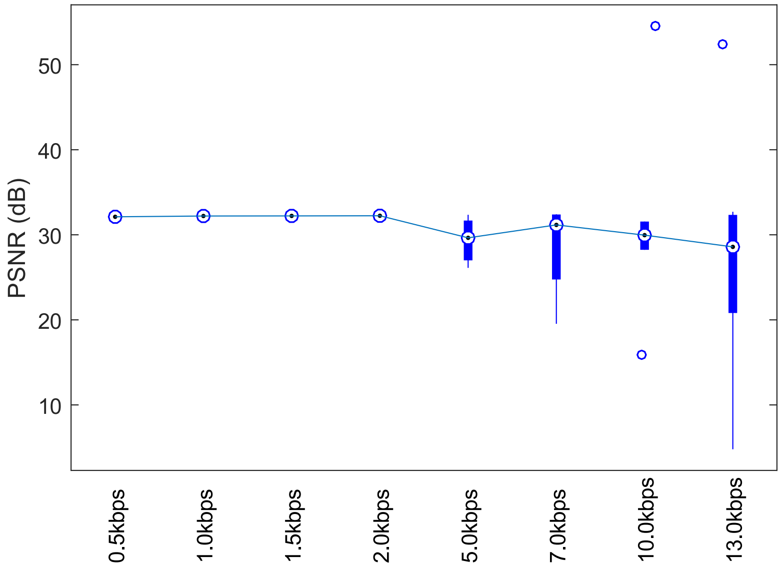


Figure 6.3: Distributions of PSNR results obtained for the various payloads.

The MSE results obtained from the restored signals compared against the host ones, at the various payloads are presented in Figure 6.4. In this figure, it can be seen that the behavior is similar to that of the PSNR. When lower payloads are embedded, the errors between the restored and host signals are lower, and more cases of perfect restoration occur, as it is expected. For payloads greater than 5 kbps, the error increases and the variance is greater, as in the case of PSNR.

The ODG results obtained from the restored signals are presented in Figure 6.5. These results indicate that the restoration capabilities of the framework, in terms of perceptual quality behaves in a constant manner, even at high embedding rates. For all the payloads tested, most of the ODG results are over -2, with a few exceptions presented in the figure as outliers; this indicates that the restoration obtained has acceptable quality. The mean values for all the payloads are over -1 ODG, which indicate not annoying distortions in the restored signals. This means that although the framework does not guarantee reversibility at high embedding rates, the restoration quality for payloads up to 13 kbps is acceptable for most practical applications.

The BER measured between the extracted and original watermarks obtained for all the payloads tested is presented in Figure 6.6. As it can be seen, perfect extraction is achieved for payloads under 5 kbps, and as the embedding rate is increased, the error on the extraction of the watermark increases as well. However, even at a payload of 13

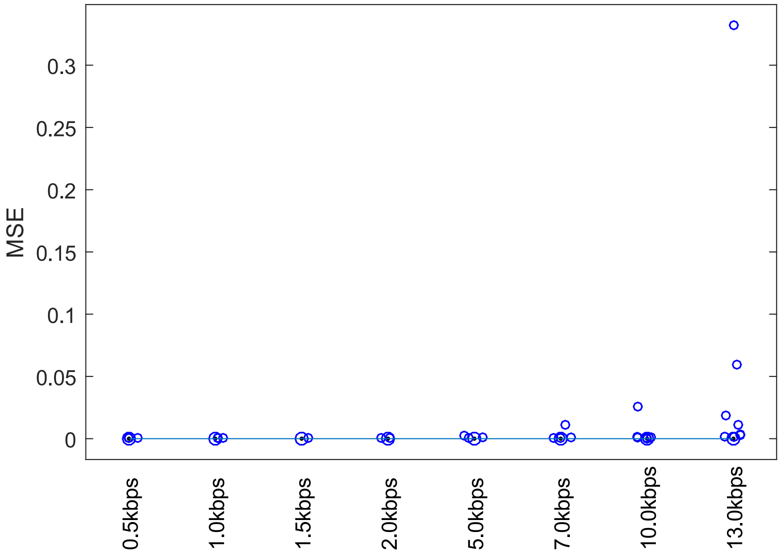


Figure 6.4: Distributions of MSE results obtained for the various payloads.

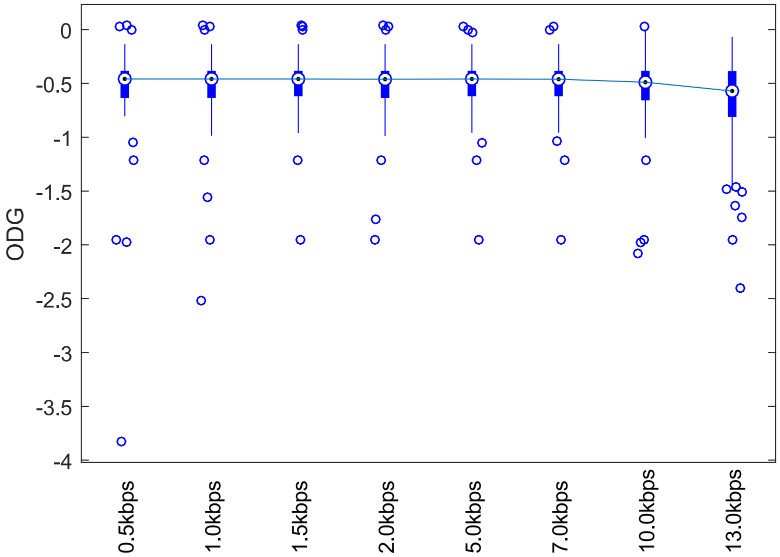


Figure 6.5: Distributions of ODG results obtained for the various payloads.

kbps, the BER is under 0.18, which is an acceptable error rate.

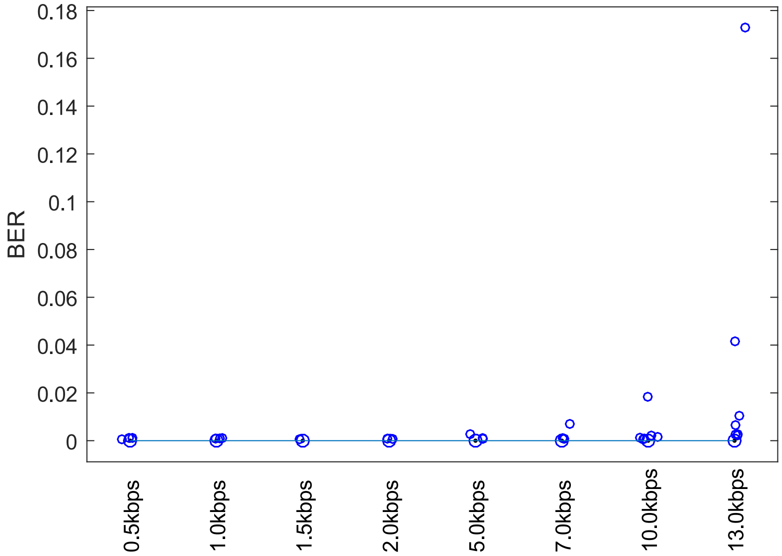


Figure 6.6: Distributions of BER results obtained for the various payloads.

Discussion.

The results obtained from this experimental setup, show that the framework can achieve perfect restoration when a payload under 2 kbps is inserted in the encoding process. Although PSNR, and MSE results at payloads over 5 kbps indicate a greater difference between restored and host signals, the perceptual quality of these restored signals is sufficient for most practical applications. For payloads under 2 kbps the framework has perfect extraction of the watermarks, even when only approximate restoration is achieved; at embedding rates over 5 kbps the framework can extract the watermark with an acceptable distortion. Furthermore, with a payload of 13 kbps, practical applications can use error correction codes to counteract the errors at the extraction, and still provide enough embedding capacity for a message to be embedded.

Conclusions.

These results have demonstrated that with payloads $\leqslant$ 2 kbps the complete framework can achieve perfect restoration when the self-recovery stage can perfectly counteract the modifications caused by the content replacement attack. In the cases where perfect restoration is not achieved, the perceptual quality of the restored signals is acceptable for most practical applications. When payloads $\geqslant$ 5 kbps are inserted in the encoding process, the distortion in the restored signals increases; however, the perceptual quality of these signals is still acceptable for practical applications. The watermark robustness of the framework is acceptable for embedding rates up to 13 kbps, which presents an adequate trade-off between errors at extraction and embedding capacity. This signifies that strategies such as the use of error correction codes can be used to counteract errors in the extraction of the watermarks while providing sufficient embedding capacity for practical applications.

A clear drawback of the proposed framework for audio are the underflow and overflow problems, the current solution adjusts the dynamic range of the signals in a pre-processing stage to avoid this problem, instead of the construction of a location map (LM) as in most RWS. The location map strategy would require the insertion of a lossless compressed version of the location map along with the payload. An initial estimation on the size of the location map is presented. Since watermark embedding is performed in the intDCT domain, a the location map that corresponds to those intDCT coefficients would have to be constructed. Because each intDCT coefficient has a repercussion in all the time-domain samples of the watermarked signal, an iterative process has to verify when an intDCT coefficient produces an underflow or overflow in time-domain samples; when this occurs, the location map position that corresponds to the problematic intDCT coefficient has to be marked.

For each audio segment, 512 intDCT coefficients are obtained, from those only the last 160 coefficients are used to insert watermark bits and the first 352 coefficients are left intact; therefore, the location map for each segment only needs 160 bits. In the best case scenario, none of the intDCT coefficients cause underflow or overflow problems, and the size of the location map is $|LM| = 0$. In the worst case scenario, all the intDCT coefficients are problematic, and $|LM| = 160$ for each segment. Therefore, in the worst case scenario, the required bit-rate to insert the LM is approximately 13.8 kbps. Results from the previous section showed that a payload up to 13 kbps can be inserted while preserving adequate signal restoration and watermark extraction capabilities. The insertion of the LM in the worst case scenario would significantly reduced the embedding payload, since almost all the embedding space would be used by the LM.

The next chapter presents a summary of this doctoral research along with the contributions, the conclusions and future work derived from this work are stated, the limitations of the work are discussed, as well as the hypothesis of the research.

Chapter 7

Conclusions and future work

In this chapter, a summary of the proposed schemes is presented as well as the contributions obtained from the doctoral research. The conclusions drawn from this work are stated, and a discussion on the hypothesis of this work is given. Finally, the limitations of the reversible watermarking scheme, the self-recovery scheme, and the framework are given, and the future work derived from this work is stated.

7.1 Summary and contributions

In this work, a general framework for reversible watermarking with watermark and signal robustness has been proposed, and it can be implemented for a variety of signals since it is an abstract construction. It is a contribution that advances the state of the art, since such schemes did not exist in the literature, as far as we know. Robust reversible watermarking schemes provide the capacity to extract the watermarks after the watermarked signals have been attacked. Self-recovery schemes on the other hand allow the restoration of regions from a signal after it has been attacked. However, these schemes do not have the capability of inserting useful payload. The proposed framework combines the characteristics of these two type of watermarking scheme to propose a new type.

Experimental results demonstrate the effectiveness of the framework through an implementation for images [Menendez-Ortiz et al., 2014], which proved to allow perfect restoration of the host images and perfect extraction of the watermarks. An implementation of the framework for audio signals corroborated these restoration and extraction capabilities with audio signals, and it also proved to maintain adequate transparency for practical audio applications.

In order to implement the framework for audio, a self-recovery scheme for audio signals with perfect restoration capabilities had to be proposed. As far as we know, self-recovery schemes for audio signals did not exist in the literature. In this work, a self-recovery scheme for audio with approximate restoration was proposed and published [Menendez-Ortiz et al., 2016]. However, to complete the framework perfect restoration was necessary, so improvements to the strategy were proposed, and it is the scheme described in Chapter 5.

To be able to insert the payload required for perfect restoration, a reversible watermarking scheme for audio had to be proposed. The requirements of such a scheme are the insertion of watermarks at high payloads, while maintaining adequate transparency for practical audio applications. The scheme proposed to fulfill these requirements is the one described in Chapter 4.

With the reversible watermarking scheme and the self-recovery scheme, the framework for audio, described in Chapter 3 could be completed. The contributions of this work to the state of the art are the general framework and its implementation for images and audio, also the proposed reversible watermarking scheme for audio, the self-recovery scheme for audio with approximate restoration which was the first scheme of its kind proposed for this type of media, and the self-recovery scheme for audio with perfect restoration.

7.2 Discussion on hypothesis

The hypothesis of this work is that a reversible watermarking scheme for audio signals with watermark and signal robustness can be proposed following a framework with an encoding and decoding process, each further divided in two stages: a fragile reversible one, and a self-recovery one.

The experimental results demonstrate the effectiveness of the proposed self-recovery scheme in terms of its restoration capabilities. With that scheme, the framework for audio was completed, and the experimental results of the framework for audio demonstrated that this construction provides reversibility through the perfect restoration of the signals, and it also has perfect extraction of the watermarks after a content replacement attack. Therefore it is demonstrated that a reversible watermarking scheme for audio with watermark and signal robustness can be designed following the proposed framework.

7.3 Limitations of the work

A strategy to deal with underflow and overflow problems has to be devised to be incorporated into the proposed reversible watermarking scheme, such as the use of a location map, as in most reversible watermarking schemes. The embedding capacity of the reversible watermarking scheme is determined by the number of frequencies that fall under the masking threshold; therefore, the embedding capacity is signal dependent. Experimental results suggest that the scheme is semi-fragile, but further experiments have to be done to determine the robustness of the scheme.

For the self-recovery scheme, a strategy to solve underflow and overflow also has to be devised. However, because of the embedding capacity condition of this scheme, a location map is not a suitable solution since it would increase the payload size even further; a new strategy has to be devised that allows the solution of the problem while preserving the embedding capacity required. The restoration of the scheme is determined by the amount of reference bits that can be inserted, so the restoration capabilities are given by the embedding capacity of the scheme. The embedding capacity depends on the frequency components on the audio segments. Audio signals with reduced frequency information will not have the capacity to carry the reference bits required for restoration. Since restoration depends on the reference bits and on non-attacked regions of the signal, the attacks supported by the scheme are limited to content replacement. If cropping is to be resisted, a synchronization strategy has to be included in the scheme.

Because of the chaining of a fragile RWS stage with a self-recovery stage, the robustness of the whole framework depends on the robustness of the later. Therefore, the framework only resists content replacement. If a synchronization strategy is added to the self-recovery scheme, the framework could also resist cropping attacks. Since the self-recovery scheme requires embedding high amount of control bits, the embedding capacity for useful payload is reduced in order to maintain a transparency threshold. The reversibility after attacks depends on the perfect restoration obtained by the self-recovery stage. However, even when perfect restoration is not obtained, the watermarks can be extracted with low error probability. As with the self-recovery scheme, the framework is signal dependent for restoration, and for embedding capacity. Because both self-recovery and fragile RWS use the masking threshold for frequency selection, the framework is not suited for signals with reduced frequency components.

7.4 Future work

As future work, the strategy of multi-bit expansion with higher expansion factor has to be explored, to increase the embedding capacity of both the reversible watermarking scheme and the self-recovery scheme. For the self-recovery scheme, this can be translated into an improvement in the restoration capabilities, since more control bits can be inserted. A synchronization strategy can be added to increase the robustness and include cropping attacks. Also, a solution for underflow and overflow has to be devised, so it can be used as part of the self-recovery scheme. The reversible watermarking scheme can use a location map to solve this problem, however this is not the case of the self-recovery scheme.

Appendices

Appendix A

Experimental results of framework for images

This appendix presents the experimental results obtained from the implementation of the framework for images detailed in Chapter 3.

In the early stages of the research, a framework for reversible audio watermarking with watermark and signal robustness was proposed. In order to validate the effectiveness of the framework, it had to be implemented and tested to evaluate it. Since self-recovery schemes from the literature existed only for images, the implementation of the framework was done first for this type of media.

Motivation.

The goal of this experimental setup was to evaluate the effectiveness of the proposed framework for images, in terms of its watermark and signal robustness against content replacement. It is also necessary to evaluate the transparency of the framework measured with the PSNR between host and watermarked images. The desired PSNR values for well reconstructed images is of 30 dB [Taubman and Marcellin, 2004].

Parameters.

The tested images are 512×512 grayscale images, common in the image processing community. The block sizes used in the self-recovery stage are of 8×8, and the watermark inserted is a binary image proposed by our research group of size 210×210, and is presented in Figure A.1.

Assumptions.

The tested images are nine miscellaneous grayscale images taken from [Weber, 1993], and [Mayer, 2009], namely Barbara, boat, gold-hill, boat on a lake, Lena, mandrill, peppers, washsat, and Zelda; all images are 512×512 8-bit grayscale images with a dynamic range in $[0, 255]$. The content replacement attack is performed manually,

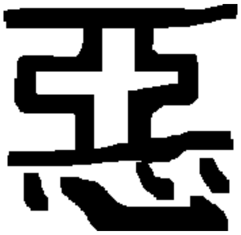


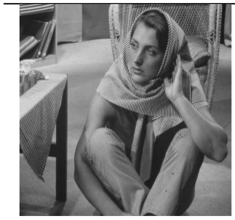
Figure A.1: Binary image inserted as a watermark.

where a region of the watermarked image smaller than 3.2% of the total pixels is substituted by a region of the same size from a different image.

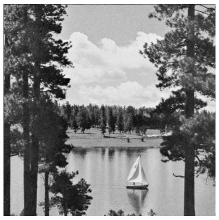
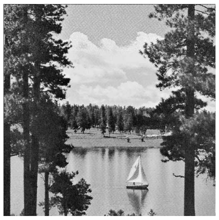
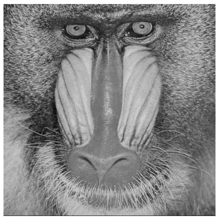
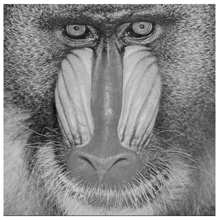
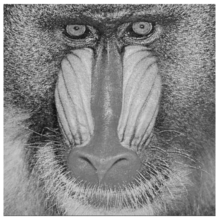
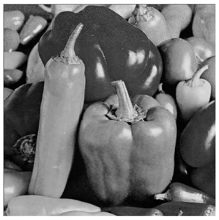
Results.

The first part of this experimental setup is to evaluate the transparency of the framework. To do so, the watermark image is embedded into the test images, and the PSNR obtained in the two stages of the framework are evaluated. The PSNR$_{\mathrm{FSW}}$ value is measured between the host image and the first-stage watermarked (FSW) image, which is produced by the fragile RWS stage; the PSNR$_{\mathrm{SSW}}$ value is measured between the first-stage watermarked image and the second-stage watermarked (SSW) image, which is produced by the self-recovery stage; and the PSNR$_{\mathrm{F}}$ value is calculated between the host image and the second-stage watermarked image, which is the global distortion of the framework. The results are shown in Table A.1, where it can be observed that the expected PSNR results of 30 dB are obtained only for two images, namely washsat, and zelda, marked in bold blue in the table. The average distortion for the encoding process of the framework is of 26.63 dB, which is lower than the expected 30 dB. However, this transparency could be acceptable for applications with higher tolerance for distortion, such as watermarking for compressed images [Emmanuel et al., 2005, Subramanyam et al., 2012].

Table A.1: PSNR between host and watermarked images after encoding.

Host image	**FSW image**	**PSNR$_{\mathrm{FSW}}$**	**SSW image**	**PSNR$_{\mathrm{SSW}}$**	**PSNR$_{\mathrm{F}}$**
		41.23		23.60	23.32

Continues on next page

Host image	FSW image	PSNR$_{FSW}$	SSW image	PSNR$_{SSW}$	PSNR$_{F}$
		43.55		27.29	26.99
		43.25		27.89	27.53
		40.98		25.01	24.56
		42.22		28.98	28.37
		36.76		20.79	20.15
		41.74		28.45	27.77
		41.84		31.27	**30.19**

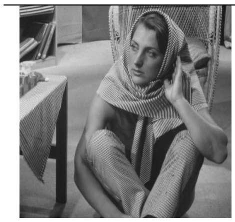
Continues on next page

Host image	FSW image	PSNR$_{\text{FSW}}$	SSW image	PSNR$_{\text{SSW}}$	PSNR$_{\text{F}}$
		42.38		31.88	**30.81**
	Average	41.55	—	27.24	**26.63**

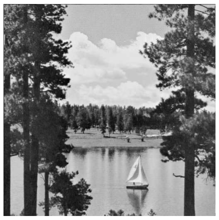
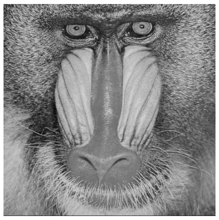
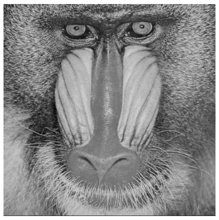
To test the robustness of the proposed scheme, the watermarked images were subjected to a content replacement attack, where some pixels of the image were changed by a region of pixels from another image. The content replacement attack was selected to perform these tests to corroborate that the proposed scheme has the same robustness as the one by Zhang and Wang [Zhang and Wang, 2008]. The distortions caused by this attack were measured in terms of PSNR values and the results are presented in Table A.2.

Finally, the decoding process of the framework is applied to the attacked images, where the self-recovery scheme counteracts the modifications caused by the attack, producing a first-stage watermarked (FSW) image. The resulting image is fed to the fragile RWS stage, where a final restored image is produced; the fragile stage also extracts the watermark image, which is compared to the embedded one. Two PSNR values from these steps are collected, the PSNR$_{\text{FSW}}$ is measured between the restored FSW image and the host image; the PSNR$_{\text{FR}}$ is measured between the final restored image and the host one. The differences between an original watermark and an extracted one are measured with BER. The collected PSNR and BER values are presented in Table A.3.

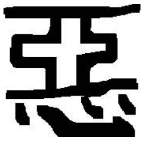
Table A.3: PSNR (dB) and BER (%) values obtained after decoding.

FSW image	PSNR$_{\text{FSW}}$	Restored image	PSNR$_{\text{FR}}$	Original watermark	Extracted watermark	BER
	41.23		***PR**			**0**

Continues on next page

FSW image	PSNR_{FSW}	Restored image	PSNR_{FR}	Original watermark	Extracted watermark	BER
	43.55		PR			0
	43.25		PR			0
	40.98		PR			0
	42.22		PR			0
	36.76		PR			0
	47.74		PR			0
	41.84		PR			0

Continues on next page

FSW image	PSNR$_{\text{FSW}}$	Restored image	PSNR$_{\text{FR}}$	Original watermark	Extracted watermark	BER
	42.38		**PR**			**0**

*PR = Perfect restoration

From Table A.3, it can be seen that the framework achieves perfect restoration after the fragile RWS stage is applied to restore the images, and that the watermarks extracted have no errors; both perfect restoration and perfect extraction results are highlighted in bold blue in the table.

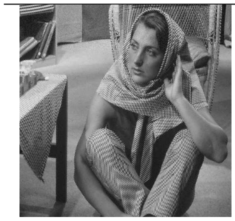
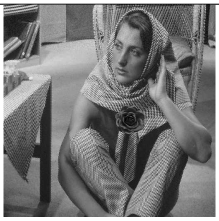
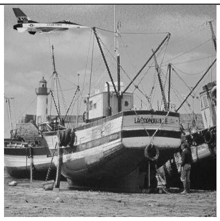
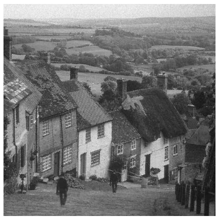
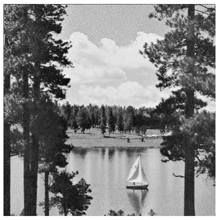
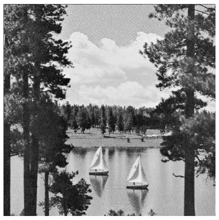
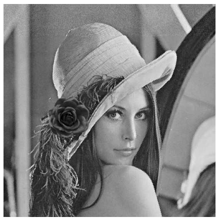
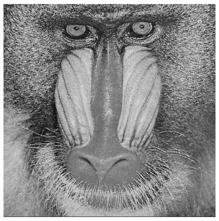
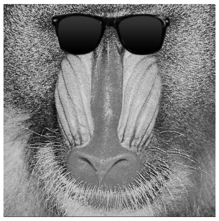
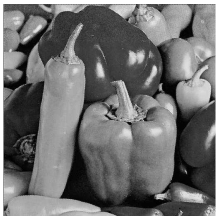
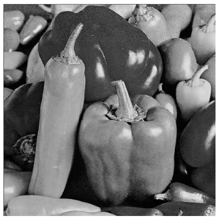
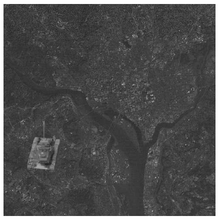
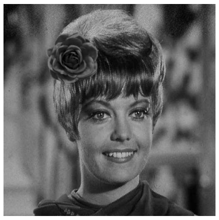
Discussion.

The transparency of the whole framework can be improved to reach the threshold of 30 dB which is required for most practical applications; however, the results obtained are acceptable for applications that deal with compressed images. From this results, it was demonstrated that watermark and image robustness could be achieved following the proposed framework.

Conclusions.

These results validate the effectiveness of the proposed framework. Because the framework fulfills the requirements of this research, which are reversibility, and watermark and signal robustness, it was assumed that the same framework could be followed to propose a solution for audio signals.

Table A.2: PSNR for attacked images after content replacement.

Watermarked image	Attacked image	PSNR	Watermarked image	Attacked image	PSNR
		22.60			24.64
		24.91			23.57
		25.69			17.52
		25.66			27.95
		25.05			

Appendix B

Description of datasets

This appendix presents a description of the datasets used for the experimental part of this research. To the best of our knowledge, in the literature there is not a collection of audio signals for testing watermarking schemes particularly. However, there are several audio datasets for machine learning and music information retrieval applications.

Although the objective of these research lines is different from the objectives of watermarking schemes, the datasets selected for the experimental part of this work contain excerpts of songs from various musical genres in non compressed format. It is necessary to test audio signals from multiple genres to have signals with various acoustic properties, and test the performance of the schemes under a wide range of characteristics of the signals. The datasets selected from the literature are the Ballroom dataset, used in music information retrieval [Gouyon, 2006]; the Music Audio Benchmark dataset constructed by the University of Dortmund [Homburg et al., 2005]; and a dataset by our research team constructed for fast testing.

B.1 Ballroom dataset

This dataset is commonly used for musical genre classification. It has 698 musical excerpts with a duration of approximately 30 seconds with a sampling frequency of 44.1 KHz, the audio signals are monaural with a quantization of 16 bits per sample in WAV format. This dataset is divided in 10 musical genres, distributed as in Table B.1.

The signals from genres Rumba-americana, Rumba-international, and Rumba-misc were combined in one genre for experimental testing, as well as genres Viennese-waltz, and Waltz. The signals from genre Samba were omitted from the tests because the audio formats of these excerpts were not compatible with the audio properties required by Matlab.

Table B.1: Distribution of genre files in Ballroom dataset.

Genre	# of files
Chachacha	111
Jive	60
Quickstep	82
Rumba-americana	7
Rumba-international	51
Rumba-misc	40
Samba	86
Tango	86
Viennese-waltz	65
Waltz	110

B.2 Music Audio Benchmark dataset

The Music Audio Benchmark (MAB) dataset is a collection constructed by the University of Dortmund for its use in machine learning and data mining applications. It has 1,886 musical excerpts with a duration of 10 seconds with a sampling frequency of 44.1 KHz. The signals were originally in MP3 format, encoded at 128 kbps. These files were converted to WAV format manually using the Audacity software [Audacity ®, 2015], the sampling frequency of 44.1 KHz was maintained, and the quantization bits were set to 16 bits per sample. This dataset is divided in 9 genres, distributed as in Table B.2.

Table B.2: Distribution of genre files in MAB dataset.

Genre	# of files
Alternative	145
Blues	120
Electronic	113
Folkcountry	222
Funksoulrnb	47
Jazz	319
Pop	116
Raphiphop	300
Rock	504

B.3 Dataset 'Ours'

A dataset constructed by our research team was used for fast testing of the various strategies explored while proposing the schemes. This dataset contains 50 excerpts from music obtained from commercial CDs, the signals have a duration of 20 seconds in WAV format, with a sampling frequency of 44.1 KHz and a quantization of 16 bits per sample. It is divided in 5 genres, and distributed as in Table B.3.

Table B.3: Distribution of genre files in dataset 'Ours'.

Genre	# of files
Jazz	10
Orchestra	10
Pop	10
Rock	10
Vocal	10

Appendix C

intDCT implementation

This appendix presents a detailed explanation of the implementation done to calculate the forward and inverse intDCT transform. It is based on the works by [Huang et al., 2004] and [Huang et al., 2006]. The fast algorithm proposed to calculate the intDCT-IV is an approximation of the DCT-IV.

The forward DCT-IV transform of an N-point audio signal $x[n]$ is given by Eq. (C.1), and its inverse transform is given by Eq. (C.3) as follows:

$$\mathbf{X}[m] = C_N^{IV} \cdot \mathbf{x}[n], \qquad m = n = 0, 1, \cdots, N-1 \tag{C.1}$$

where $\mathbf{X}$ represents the DCT coefficients of $\mathbf{x}$. C_N^{IV} is the transform matrix, defined as:

$$C_N^{IV} = \sqrt{\frac{2}{N}} \left[\cos \left(\frac{(m+\frac{1}{2})(n+\frac{1}{2})\pi}{N} \right) \right], \tag{C.2}$$

where $m = 0, 1, \cdots, N-1$ and $n = 0, 1, \cdots, N-1$. Because C_N^{IV} is an orthogonal matrix, the inverse DCT transform is given by:

$$\mathbf{x}[n] = C_N^{IV} \cdot \mathbf{X}[m]. \tag{C.3}$$

The algorithm by [Huang et al., 2004] decomposes matrix C_N^{IV} in six matrices in the following way:

$$C_N^{IV} = R_1 \cdot R_2 \cdot S \cdot T_1 \cdot T_2 \cdot P, \tag{C.4}$$

where P is a permutation matrix given by [Huang et al., 2006] that reorders the components the elements of vector $\mathbf{x}$ by separating even indexes from odd indexes, *i.e.*

$$\begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(3) \\ \vdots \\ \mathbf{x}(N-1) \\ \mathbf{x}(2) \\ \mathbf{x}(4) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix} = P \begin{bmatrix} \mathbf{x}(1) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}. \tag{C.5}$$

Matrices R_1, R_2, S, T_1, and T_2 are defined as:

$$R_1 = \begin{bmatrix} I_{N/2} & 0 \\ H_1 & I_{N/2} \end{bmatrix}, \tag{C.6}$$

$$R_2 = \begin{bmatrix} I_{N/2} & H_2 \\ 0 & I_{N/2} \end{bmatrix}, \tag{C.7}$$

$$S = \begin{bmatrix} I_{N/2} & 0 \\ H_3 + K_1 & I_{N/2} \end{bmatrix}, \tag{C.8}$$

$$T_1 = \begin{bmatrix} -D_{N/2} & K_2 \\ 0 & I_{N/2} \end{bmatrix}, \tag{C.9}$$

$$T_2 = \begin{bmatrix} I_{N/2} & 0 \\ K_3 & I_{N/2} \end{bmatrix}. \tag{C.10}$$

Matrix $I_{N/2}$ is the identity matrix of order $N/2$, and the critical sub-matrices K_1, K_2,

K_3, H_1, H_2, and H_3 are given by:

$$K_1 = -(C^{IV}_{N/2} \cdot D_{N/2} + \sqrt{2} \cdot I_{N/2}) \cdot C^{IV}_{N/2}, \tag{C.11}$$

$$K_2 = \frac{C^{IV}_{N/2}}{\sqrt{2}}, \tag{C.12}$$

$$K_3 = \sqrt{2} \cdot C^{IV}_{N/2} \cdot D_{N/2} + I_{N/2}, \tag{C.13}$$

$$H_1 = H_3 = \begin{bmatrix} & & & -\tan\frac{(N-1)\pi}{8N} \\ & & \iddots & \\ & -\tan\frac{3\pi}{8N} & & \\ -\tan\frac{\pi}{8N} & & & \end{bmatrix}, \tag{C.14}$$

$$H_2 = \begin{bmatrix} & & & \sin\frac{\pi}{4N} \\ & & \sin\frac{3\pi}{4N} & \\ & \iddots & & \\ \sin\frac{(N-1)\pi}{4N} & & & \end{bmatrix}. \tag{C.15}$$

$C^{IV}_{N/2}$ is the DCT-IV matrix of order $N/2$, and $D_{N/2}$ is a diagonal matrix of order $N/2$ defined by:

$$D_{N/2} = \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & -1 \end{bmatrix}. \tag{C.16}$$

From eq. C.4, it can be seen that the integer DCT-IV transform can be performed through five lifting stages. In each of these stages, a rounding operation is applied, producing the integer results. The algorithm for the forward integer DCT-IV transform is given in Algorithm 11, where $\lfloor . \rfloor$ is the rounding operator. Let $\mathbf{x}$ and $\mathbf{X}$ be the input and output vectors, the first stage from eq. C.4 is

$$\mathbf{X} = T_2 \cdot \mathbf{x}. \tag{C.17}$$

If $\mathbf{x}$ and $\mathbf{X}$ are divided in two halves, eq. C.17 can be re-written as:

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} I_{N/2} & 0 \\ K_3 & I_{N/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}. \tag{C.18}$$

The forward integer transform from the first stage is then:

$$\mathbf{X}_1 = \mathbf{x}_1 \tag{C.19}$$

$$\mathbf{X}_2 = \lfloor K_3 \cdot \mathbf{x}_1 \rfloor + \mathbf{x}_2. \tag{C.20}$$

The rest of the lifting stages are listed in Algorithm 11.

```
Input  : Time-domain signal ($\mathbf{x}$)
Output: IntDCT signal ($\mathbf{X}$)
$N \leftarrow |\mathbf{x}|$
Calculate DCT-IV transform matrix $C^{IV}_{N/2}$
Calculate critical sub-matrices $K_1$, $K_2$, $K_3$, $H1$, $H_2$, and $H_3$
$\mathbf{X}_0 \leftarrow \mathbf{x} \cdot P$
                                        /* Divide permuted vector in two */
$\mathbf{x}_1 \leftarrow \mathbf{X}_0(1 : N/2)$
$\mathbf{x}_2 \leftarrow \mathbf{X}_0(N/2 + 1 : N)$
                                                         /* 1st lifting */
$\mathbf{X}_{11} \leftarrow \mathbf{x}_1$
$\mathbf{X}_{12} \leftarrow \lfloor K_3 \cdot \mathbf{x}_1 \rfloor + \mathbf{x}_2$
                                                         /* 2nd lifting */
$\mathbf{X}_{21} \leftarrow -D_{N/2} \cdot \mathbf{X}_{11} + \lfloor K_2 \cdot \mathbf{X}_{12} \rfloor$
$\mathbf{X}_{22} \leftarrow \mathbf{X}_{12}$
                                                         /* 3rd lifting */
$\mathbf{X}_{31} \leftarrow \mathbf{X}_{21}$
$\mathbf{X}_{32} \leftarrow \lfloor (H_3 + K_1) \cdot \mathbf{X}_{21} \rfloor + \mathbf{X}_{22}$
                                                         /* 4th lifting */
$\mathbf{X}_{41} \leftarrow \mathbf{X}_{31} + \lfloor H_2 \cdot \mathbf{X}_{32} \rfloor$
$\mathbf{X}_{42} \leftarrow \mathbf{X}_{32}$
                                                         /* 5th lifting */
$\mathbf{X}_{51} \leftarrow \mathbf{X}_{41}$
$\mathbf{X}_{52} \leftarrow \lfloor H_1 \cdot \mathbf{X}_{41} \rfloor + \mathbf{X}_{42}$
                                  /* Concatenating transform coefficients */
$X \leftarrow [\mathbf{X}_{51}; \mathbf{X}_{52}]$
```

Algorithm 11: Forward integer DCT-IV transform.

The inverse integer DCT-IV transform from the first stage is then:

$$\mathbf{x}_1 = \mathbf{X}_1 \tag{C.21}$$

$$\mathbf{x}_2 = \mathbf{X}_2 - \lfloor K_3 \cdot \mathbf{X}_1 \rfloor \tag{C.22}$$

The inverse transform is performed in reverse order from the forward transform, and the final coefficients are reordered according to a permutation vector $\mathbf{P}_{eo} \neq P$. This vector is not defined by [Huang et al., 2004] or [Huang et al., 2006] and it had to be found through an analysis of the transform matrix C_N^{IV}. The calculation of the permutation vector $\mathbf{P}_{eo}$ is given in Algorithm 12.

```
Input   : Input vector ($\mathbf{y}$)
Output: Permutation vector ($\mathbf{P}_{eo}$)
$N \leftarrow |\mathbf{y}|$
for $i = 1 : N/2$ do
    $\mathbf{P}_{eo}((i-1) \times 2 + 1) \leftarrow i$
    $\mathbf{P}_{eo}((i-1) \times 2 + 2) \leftarrow i + N/2$
end
```

Algorithm 12: Calculation of permutation vector $\mathbf{P}_{eo}$.

The five lifting stages of the inverse integer DCT-IV transform are given in Algorithm 13.

```
Input   : IntDCT signal ($\mathbf{X}$)
Output: Time-domain signal ($\mathbf{x}$)
$N \leftarrow |\mathbf{X}|$
Calculate DCT-IV transform matrix $C^{IV}_{N/2}$
Calculate critical sub-matrices $K_1$, $K_2$, $K_3$, $H1$, $H_2$, and $H_3$
                                        /* Divide vector in two */
$\mathbf{X}_{51} \leftarrow \mathbf{X}(1 : N/2)$
$\mathbf{X}_{52} \leftarrow \mathbf{X}(N/2 + 1 : N)$
                                        /* 5th inverse lifting */
$\mathbf{X}_{41} \leftarrow \mathbf{X}_{51}$
$\mathbf{X}_{42} \leftarrow \mathbf{X}_{52} - \lfloor H_1 \cdot \mathbf{X}_{51} \rfloor$
                                        /* 4th inverse lifting */
$\mathbf{X}_{31} \leftarrow \mathbf{X}_{41} - \lfloor H_2 \cdot \mathbf{X}_{42} \rfloor$
$\mathbf{X}_{32} \leftarrow \mathbf{X}_{42}$
                                        /* 3rd inverse lifting */
$\mathbf{X}_{21} \leftarrow \mathbf{X}_{31}$
$\mathbf{X}_{22} \leftarrow \mathbf{X}_{32} - \lfloor (H_3 + K_1) \cdot \mathbf{X}_{31} \rfloor$
                                        /* 2nd inverse lifting */
$\mathbf{X}_{11} \leftarrow \lfloor D_{N/2} \cdot K_2 \cdot \mathbf{X}_{22} \rfloor + \lfloor -D_{N/2} \cdot \mathbf{X}_{21} \rfloor$
$\mathbf{X}_{12} \leftarrow \mathbf{X}_{22}$
                                        /* 1st inverse lifting */
$\mathbf{x}_1 \leftarrow \mathbf{X}_{11}$
$\mathbf{x}_2 \leftarrow \mathbf{X}_{12} - \lfloor K_3 \cdot \mathbf{X}_{11} \rfloor$
                         /* Concatenating transform coefficients */
$\mathbf{x}_{\text{lift}} \leftarrow [\mathbf{x}_1; \mathbf{x}_2]$
                                   /* Reordering coefficients */
$\mathbf{x} \leftarrow \mathbf{x}_{\text{lift}}(\mathbf{P}_{eo})$
```

Algorithm 13: Inverse integer DCT-IV transform.

References

S.S. Agaian, D. Akopian, O. Caglayan, and S.A. D'Souza. Lossless Adaptive Digital Audio Steganography. In *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005*, pages 903–906, October 2005.

T. Ahsan, T. Mohammad, M.S. Alam, and Ui-Pil Chong. Digital watermarking based image authentication and restoration by quantization of integer wavelet transform coefficients. In *International Conference on Informatics, Electronics Vision (ICIEV), 2012*, pages 1163–1167, 2012. doi: 10.1109/ICIEV.2012.6317509.

Lingling An, Xinbo Gao, Xuelong Li, Dacheng Tao, Cheng Deng, and Jie Li. Robust reversible watermarking via clustering and enhanced pixel-wise masking. *IEEE Transactions on Image Processing*, 21(8):3598–3611, 2012a. ISSN 1057-7149. doi: 10.1109/TIP.2012.2191564.

Lingling An, Xinbo Gao, Yuan Yuan, and Dacheng Tao. Robust lossless data hiding using clustering and statistical quantity histogram. *Neurocomputing*, 77(1):1–11, 2012b. ISSN 0925-2312. doi: 10.1016/j.neucom.2011.06.012. URL http://www.sciencedirect.com/science/article/pii/S0925231211003614.

Lingling An, Xinbo Gao, Yuan Yuan, Dacheng Tao, Cheng Deng, and Feng Ji. Content-adaptive reliable robust lossless data embedding. *Neurocomputing*, 79 (0):1–11, 2012c. ISSN 0925-2312. doi: 10.1016/j.neucom.2011.08.019. URL http://www.sciencedirect.com/science/article/pii/S0925231211005157.

Audacity ®. Audacity. Online, 2015. URL http://www.audacityteam.org.

Brett Bradley and Adnan M. Alattar. High-capacity invertible data-hiding algorithm for digital audio. In *SPIE Proceedings 5681, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 789–800, 2005. doi: 10.1117/12.586042. URL http://dx.doi.org/10.1117/12.586042.

S. Bravo-Solorio, C.-T. Li, and A.K. Nandi. Watermarking method with exact self-propagating restoration capabilities. In *IEEE International Workshop on Information*

Forensics and Security (WIFS), 2012, pages 217–222, 2012a. doi: 10.1109/WIFS.2012.6412652.

S. Bravo-Solorio, C.-T. Li, and A.K. Nandi. Watermarking with low embedding distortion and self-propagating restoration capabilities. In *19th IEEE International Conference on Image Processing (ICIP), 2012*, pages 2197–2200, 2012b. doi: 10.1109/ICIP.2012.6467330.

Roberto Caldelli, Franco Bartolini, Vito Cappellini, Alessandro Piva, and Mauro Barni. A New Self-Recovery Technique for Image Authentication. In Narciso Garcia, Luis Salgado, and Jose M. Martinez, editors, *Visual Content Processing and Representation*, volume 2849 of *Lecture Notes in Computer Science*, pages 164–171. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-20081-9. doi: 10.1007/978-3-540-39798-4_22.

Mehmet U Celik, Gaurav Sharma, A Murat Tekalp, and Eli S Saber. Video authentication with self-recovery. In *Electronic Imaging 2002*, pages 531–541. International Society for Optics and Photonics, 2002. doi: 10.1117/12.465311. URL http://dx.doi.org/10.1117/12.465311.

Chin-Chen Chang, Pei-Yu Lin, and Jieh-Shan Yeh. Preserving robustness and removability for digital watermarks using subsampling and difference correlation. *Information Sciences*, 179(13):2283–2293, 2009. ISSN 0020-0255. doi: 10.1016/j.ins.2009.03.003.

Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. A secure and improved self-embedding algorithm to combat digital document forgery. *Signal Processing*, 89(12):2324–2332, 2009. ISSN 0165-1684. doi: 10.1016/j.sigpro.2009.02.001. URL http://www.sciencedirect.com/science/article/pii/S0165168409000425.

Fan Chen, Hongjie He, and Hongxia Wang. A Fragile Watermarking Scheme for Audio Detection and Recovery. In *Congress on Image and Signal Processing, 2008. CISP '08.*, volume 5, pages 135–138, 2008. doi: 10.1109/CISP.2008.298.

Quan Chen, Shijun Xiang, and Xinrong Luo. Reversible Watermarking for Audio Authentication Based on Integer DCT and Expansion Embedding. In YunQ. Shi, Hyoung-Joong Kim, and Fernando Pérez-González, editors, *Digital Forensics and Watermaking*, volume 7809 of *Lecture Notes in Computer Science*, pages 395–409. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40098-8. doi: 10.1007/978-3-642-40099-5_33. URL http://dx.doi.org/10.1007/978-3-642-40099-5_33.

E Chrysochos, V Fotopoulos, AN Skodras, and M Xenos. Reversible image watermarking based on histogram modification. In *11th Panhellenic Conference on Informatics (PCI 2007)*, pages 93–104, 2007.

G. Coatrieux, J. Montagner, H. Huang, and C. Roux. Mixed Reversible and RONI Watermarking for Medical Image Reliability Protection. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007*, pages 5653–5656, 2007. doi: 10.1109/IEMBS.2007.4353629.

D. Coltuc. Improved embedding for prediction-based reversible watermarking. *IEEE Transactions on Information Forensics and Security*, 6(3):873–882, Sept 2011. ISSN 1556-6013. doi: 10.1109/TIFS.2011.2145372.

D. Coltuc and J.-M. Chassery. Very Fast Watermarking by Reversible Contrast Mapping. *IEEE Signal Processing Letters*, 14(4):255–258, 2007. ISSN 1070-9908. doi: 10.1109/LSP.2006.884895.

D. Coltuc and A. Tudoroiu. Multibit versus multilevel embedding in high capacity difference expansion reversible watermarking. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012*, pages 1791–1795, August 2012.

Clara Cruz, Rogelio Reyes, Mariko Nakano, and Hector Perez. Semi-fragile watermarking based content image authentication scheme. *Revista Facultad de Ingeniería Universidad de Antioquia*, pages 160–169, 12 2010. ISSN 0120-6230.

C. De Vleeschouwer, J.-F. Delaigle, and B. Macq. Circular interpretation of bijective transformations in lossless watermarking for media asset management. *Multimedia, IEEE Transactions on*, 5(1):97–105, 2003. ISSN 1520-9210. doi: 10.1109/TMM.2003.809729.

I. C. Dragoi and D. Coltuc. Local-prediction-based difference expansion reversible watermarking. *IEEE Transactions on Image Processing*, 23(4):1779–1790, April 2014. ISSN 1057-7149. doi: 10.1109/TIP.2014.2307482.

S. Emmanuel, H.C. Kiang, and A. Das. A Reversible Watermarking Scheme for JPEG-2000 Compressed Images. In *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005.*, pages 69–72, July 2005.

J. Fridrich and M. Goljan. Images with self-correcting capabilities. In *International Conference on Image Processing, 1999. ICIP 99.*, volume 3, pages 792–796, 1999a. doi: 10.1109/ICIP.1999.817228.

Jiri Fridrich and Miroslav Goljan. Protection of digital images using self embedding. In *Symposium on Content Security and Data Hiding in Digital Media*. Newark, NJ, USA, May 1999b.

Tie-Gang Gao and Qiao-Lun Gu. Reversible watermarking algorithm based on wavelet lifting scheme. In *International Conference on Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07*, volume 4, pages 1771–1775, 2007. doi: 10.1109/ICWAPR.2007.4421740.

Xinbo Gao, Lingling An, Yuan Yuan, Dacheng Tao, and Xuelong Li. Lossless Data Embedding Using Generalized Statistical Quantity Histogram. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1061–1070, 2011. ISSN 1051-8215. doi: 10.1109/TCSVT.2011.2130410.

J.J. Garcia-Hernandez. Exploring Reversible Digital Watermarking in Audio Signals Using Additive Interpolation-error Expansion. In *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012*, pages 142–145, July 2012.

Emilia Gómez, P. Cano, L. Gomes, E. Batlle, and M. Bonnet. Mixed watermarking-fingerprinting approach for integrity verification of audio recordings. In *Proceedings of the International Telecommunications Symposium*, 2002.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson Education, 3rd edition, July 2011. ISBN 0133002322, 9780133002324.

Fabien Gouyon. Ballroom dataset. Online, March 2006. URL http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html.

QiaoLun Gu, Guanglin Han, Tiegang Gao, and Zengqiang Chen. A Novel Adaptive Reversible Watermarking Algorithm Based on Wavelet Lifting Scheme. In *International Conference on Information Engineering and Computer Science, 2009. ICIECS 2009*, pages 1–4, 2009. doi: 10.1109/ICIECS.2009.5366941.

Xinlu Gui, Xiaolong Li, and Bin Yang. A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. *Signal Processing*, 98:370 – 380, 2014. ISSN 0165-1684. doi: http://dx.doi.org/10.1016/j.sigpro.2013.12.005. URL http://www.sciencedirect.com/science/article/pii/S0165168413004933.

F. Hartung and F. Ramme. Digital rights management and watermarking of multimedia content for m-commerce applications. *Communications Magazine, IEEE*, 38(11):78–84, 2000. doi: 10.1109/35.883493. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=883493.

Y. M Y Hasan and A.M. Hassan. Tamper Detection with Self-Correction Hybrid Spatial-DCT Domains Image Authentication Technique. In *IEEE International Symposium on Signal Processing and Information Technology, 2007*, pages 369–374, 2007. doi: 10.1109/ISSPIT.2007.4458195.

A. M. Hassan, A. Al-Hamadi, Y. M Y Hasan, M. A A Wahab, A. Panning, and B. Michaelis. Variable block-size image authentication with localization and self-recovery. In *17th IEEE International Conference on Image Processing (ICIP), 2010*, pages 3665–3668, 2010a. doi: 10.1109/ICIP.2010.5652114.

A.M. Hassan, A. Al-Hamadi, B. Michaelis, Y. M Y Hasan, and M. A A Wahab. Secure Self-Recovery Image Authentication Using Randomly-Sized Blocks. In *20th International Conference on Pattern Recognition (ICPR), 2010*, pages 1445–1448, 2010b. doi: 10.1109/ICPR.2010.357.

Ammar M. Hassan, Ayoub Al-Hamadi, Yassin M. Y. Hasan, Mohamed A. A. Wahab, and Bernd Michaelis. Secure Block-Based Video Authentication with Localization and Self-Recovery. *World Academy of Science, Engineering and Technology*, 2009(33): 69–74, September 2009.

Hong-Jie He, Jia-Shu Zhang, and Heng-Ming Tai. Self-recovery Fragile Watermarking Using Block-Neighborhood Tampering Characterization. In Stefan Katzenbeisser and Ahmad-Reza Sadeghi, editors, *Information Hiding*, volume 5806 of *Lecture Notes in Computer Science*, pages 132–145. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04430-4. doi: 10.1007/978-3-642-04431-1_10.

HongJie He, JiaShu Zhang, and Fan Chen. A self-recovery fragile watermarking scheme for image authentication with superior localization. *Science in China Series F: Information Sciences*, 51(10):1487–1507, 2008. ISSN 1009-2757. doi: 10.1007/s11432-008-0094-1. URL http://dx.doi.org/10.1007/s11432-008-0094-1.

Hongjie He, Fan Chen, Heng-Ming Tai, T. Kalker, and Jiashu Zhang. Performance Analysis of a Block-Neighborhood-Based Self-Recovery Fragile Watermarking Scheme. *IEEE Transactions on Information Forensics and Security*, 7(1):185–196, 2012. ISSN 1556-6013. doi: 10.1109/TIFS.2011.2162950.

Miguel Angel Hernandez-Morales. Esquema Robusto de Marca de Agua Digital Reversible en Imágenes. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2012.

Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A Benchmark Dataset for Audio Classification and Clustering. In *ISMIR*, volume 2005, pages 528–31, 2005. URL http://www-ai.cs.uni-dortmund.de/audio.html.

Chris W Honsinger, Paul W Jones, Majid Rabbani, and James C Stoffel. Lossless recovery of an original image containing embedded data. US Patent, August 2001. US Patent 6,278,791.

Haibin Huang, S. Rahardja, Rongshan Yu, and Xiao Lin. A fast algorithm of integer MDCT for lossless audio coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).* , volume 4, pages IV–177–IV–180, May 2004.

Haibin Huang, S. Rahardja, Rongshan Yu, and Xiao Lin. Integer MDCT with enhanced approximation of the DCT-IV. *IEEE Transactions on Signal Processing*, 54(3):1156–1159, March 2006. ISSN 1053-587X. doi: 10.1109/TSP.2005.862942.

Xuping Huang, I. Echizen, and A. Nishimura. A New Approach of Reversible Acoustic Steganography for Tampering Detection. In *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010*, pages 538–542, October 2010.

Xuping Huang, Akira Nishimura, and Isao Echizen. A Reversible Acoustic Steganography for Integrity Verification. In Hyoung-Joong Kim, YunQing Shi, and Mauro Barni, editors, *Digital Watermarking*, volume 6526 of *Lecture Notes in Computer Science*, pages 305–316. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18404-8. doi: 10.1007/978-3-642-18405-5_25. URL http://dx.doi.org/10.1007/978-3-642-18405-5_25.

KuoLung Hung and Chin-Chen Chang. Recoverable Tamper Proofing Technique for Image Authentication Using Irregular Sampling Coding. In Bin Xiao, LaurenceT. Yang, Jianhua Ma, Christian Muller-Schloer, and Yu Hua, editors, *Autonomic and Trusted Computing*, volume 4610 of *Lecture Notes in Computer Science*, pages 333–343. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73546-5. doi: 10.1007/978-3-540-73547-2_35.

Yongjin Huo, Shijun Xiang, Shangyi Liu, Xinrong Luo, and Zhongliang Bai. Reversible audio watermarking algorithm using non-causal prediction. *Wuhan University*

Journal of Natural Sciences, 18(5):455–460, 2013. ISSN 1007-1202. doi: 10.1007/s11859-013-0956-2. URL http://dx.doi.org/10.1007/s11859-013-0956-2.

M. Iwata, T. Hori, A. Shiozaki, and A. Ogihara. Digital watermarking method for tamper detection and recovery of JPEG images. In *International Symposium on Information Theory and its Applications (ISITA), 2010*, pages 309–314, 2010. doi: 10.1109/ISITA.2010.5649171.

Xuemei Jiang and Quan Liu. Semi-fragile watermarking algorithm for image tampers localization and recovery. *Journal of Electronics (China)*, 25(3):343–351, 2008. ISSN 0217-9822. doi: 10.1007/s11767-006-6209-1. URL http://dx.doi.org/10.1007/s11767-006-6209-1.

A Karantonis and J Ellinas. Self embedding watermarking for preventing image content from tampering. In *International Conference in Information Technology (eRA4)*, 2009.

Asifullah Khan, Ayesha Siddiqa, Summuyya Munib, and Sana Ambreen Malik. A recent survey of reversible watermarking techniques. *Information Sciences*, pages 1–22, 2014. ISSN 0020-0255. doi: 10.1016/j.ins.2014.03.118. URL http://www.sciencedirect.com/science/article/pii/S0020025514004150.

Kyung-Su Kim, Min-Jeong Lee, Young-Ho Suh, and Heung-Kyu Lee. Robust lossless data hiding based on block gravity center for selective authentication. In *IEEE International Conference on Multimedia and Expo, 2009. ICME 2009*, pages 1022–1025, 2009. doi: 10.1109/ICME.2009.5202671.

P. Korus and A. Dziech. Reconfigurable self-embedding with high quality restoration under extensive tampering. In *19th IEEE International Conference on Image Processing (ICIP), 2012*, pages 2193–2196, 2012. doi: 10.1109/ICIP.2012.6467329.

P. Korus and A. Dziech. Efficient Method for Content Reconstruction With Self-Embedding. *IEEE Transactions on Image Processing*, 22(3):1134–1147, 2013. ISSN 1057-7149. doi: 10.1109/TIP.2012.2227769.

P. Korus, W. Szmuc, and A. Dziech. A scheme for censorship of sensitive image content with high-quality reconstruction ability. In *IEEE International Conference on Multimedia and Expo (ICME), 2010*, pages 1073–1078, 2010. doi: 10.1109/ICME.2010.5583410.

Pawel Korus, Lucjan Janowski, and Piotr Romaniak. Automatic quality control of digital image content reconstruction schemes. In *IEEE International Conference on Multimedia and Expo (ICME), 2011*, pages 1–6, 2011. doi: 10.1109/ICME.2011.6011872.

Chunlei Li, Yunhong Wang, Bin Ma, and Zhaoxiang Zhang. A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure. *Computers & Electrical Engineering*, 37(6):927–940, 2011. ISSN 0045-7906. doi: 10.1016/j.compeleceng.2011.09.007. URL http://www.sciencedirect.com/science/article/pii/S0045790611001327.

X. Li, B. Li, B. Yang, and T. Zeng. General framework to histogram-shifting-based reversible data hiding. *IEEE Transactions on Image Processing*, 22(6):2181–2191, June 2013. ISSN 1057-7149. doi: 10.1109/TIP.2013.2246179.

X. Li, W. Zhang, X. Gui, and B. Yang. Efficient reversible data hiding based on multiple histograms modification. *IEEE Transactions on Information Forensics and Security*, 10 (9):2016–2027, Sept 2015. ISSN 1556-6013. doi: 10.1109/TIFS.2015.2444354.

Phen Lan Lin, Chung-Kai Hsieh, and Po-Whei Huang. A hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognition*, 38(12):2519–2529, 2005. ISSN 0031-3203. doi: 10.1016/j.patcog.2005.02.007. URL http://www.sciencedirect.com/science/article/pii/S0031320305000890.

Yiqing Lin and Waleed H. Abdulla. *Audio Watermark: A Comprehensive Foundation Using MATLAB*. Springer International Publishing, 1 edition, 2015. ISBN 978-3-319-07973-8. doi: 10.1007/978-3-319-07974-5.

Chun-Shien Lu. *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. IGI Publishing, Hershey, PA, USA, 2004. ISBN 1591401925.

Gregory Mayer. Image repository. Online, 2009. URL http://links.uwaterloo.ca/Repository.html.

J.A. Mendoza-Noriega, B.M. Kurkoski, M. Nakano-Miyatake, and H. Perez-Meana. Halftoning-based self-embedding watermarking for image authentication and recovery. In *53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2010*, pages 612–615, 2010. doi: 10.1109/MWSCAS.2010.5548902.

Jose Antonio Mendoza-Noriega, M Kurkoski, Mariko Nakano Miyatake, and Hector Perez Meana. Image authentication and recovery using BCH error-correcting codes. *International Journal of Computers*, 5(1):26–33, 2011.

Alejandra Menendez-Ortiz, Claudia Feregrino-Uribe, and J. J. Garcia-Hernandez. Reversible image watermarking scheme with perfect watermark and host restoration after a content replacement attack . In *The 2014 International Conference on Security and Management (SAM'14)*, volume 13 of *The 2014 International Conference on Security and Management (SAM'14)*, pages 385–391, July 2014.

Alejandra Menendez-Ortiz, Claudia Feregrino-Uribe, Jose Juan Garcia-Hernandez, and Zobeida Jezabel Guzman-Zavaleta. Self-recovery scheme for audio restoration after a content replacement attack. *Multimedia Tools and Applications*, pages 1–28, 2016. ISSN 1573-7721. doi: 10.1007/s11042-016-3783-6. URL http://dx.doi.org/10.1007/s11042-016-3783-6.

B.G. Mobasseri. A spatial digital video watermark that survives MPEG. In *International Conference on Information Technology: Coding and Computing, 2000.*, pages 68–73, 2000. doi: 10.1109/ITCC.2000.844185.

Bijan G. Mobasseri and Aaron T. Evans. Content-dependent video authentication by self-watermarking in color space. In *SPIE Proceedings in Security and Watermarking of Multimedia Contents III*, volume 4314, pages 35–44. International Society of Optics and Photonics, August 2001. doi: 10.1117/12.435437. URL http://dx.doi.org/10.1117/12.435437.

Heather Newton. Music censorship: An overview. In *Points of view: Music censorship (2011)*, volume 1. EBSCOhost, November 2012. URL http://www.billingsschools.org/cms/lib3/MT01001765/Centricity/Domain/289/Grade%2010%20-%20Argumentative%20-%20Music%20Censorship%20articles.pdf.

Zhicheng Ni, Y.Q. Shi, N. Ansari, Wei Su, Q. Sun, and Xiao Lin. Robust lossless image data hiding. In *IEEE International Conference on Multimedia and Expo, 2004. ICME '04*, volume 3, pages 2199–2202, 2004. doi: 10.1109/ICME.2004.1394706.

Zhicheng Ni, Y.Q. Shi, N. Ansari, Wei Su, Q. Sun, and Xiao Lin. Robust lossless image data hiding designed for semi-fragile image authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(4):497–509, 2008. ISSN 1051-8215. doi: 10.1109/TCSVT.2008.918761.

A. Nishimura. Reversible Audio Data Hiding Using Linear Prediction and Error Expansion. In *Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011*, pages 318–321, October 2011.

A. Nishimura. Controlling Quality and Payload in Reversible Data Hiding Based on Modified Error Expansion for Segmental Audio Waveforms. In *Eighth International*

Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012, pages 110–113, July 2012a.

R. Nishimura. Audio Watermarking Using Spatial Masking and Ambisonics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2461–2469, November 2012b. ISSN 1558-7916.

Mark Owen. *Practical Signal Processing*. Cambridge University Press, 2007.

Fei Peng, Xiaolong Li, and Bin Yang. Adaptive reversible data hiding scheme based on integer transform. *Signal Processing*, 92(1):54 – 62, 2012. ISSN 0165-1684. doi: http://dx.doi.org/10.1016/j.sigpro.2011.06.006. URL http://www.sciencedirect.com/science/article/pii/S0165168411001964.

Zhenxing Qian and Guorui Feng. Inpainting Assisted Self Recovery With Decreased Embedding Data. *IEEE Signal Processing Letters*, 17(11):929–932, 2010. ISSN 1070-9908. doi: 10.1109/LSP.2010.2072991.

Zhenxing Qian and Tong Qiao. Image Self-Embedding with Large-Area Restoration Capability. In *International Conference on Multimedia Information Networking and Security (MINES), 2010*, pages 649–652, 2010. doi: 10.1109/MINES.2010.141.

Chuan Qin, Chin-Chen Chang, and Pei-Yu Chen. Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism. *Signal Processing*, 92(4):1137–1150, 2012. ISSN 0165-1684. doi: 10.1016/j.sigpro.2011.11.013. URL http://www.sciencedirect.com/science/article/pii/S0165168411003987.

M.J. Saberian, M.A. Akhaee, and F. Marvasti. An invertible quantization based watermarking approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, pages 1677–1680, 2008. doi: 10.1109/ICASSP.2008.4517950.

V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi. Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):989–999, July 2009. ISSN 1051-8215. doi: 10.1109/TCSVT.2009.2020257.

Vasiliy Sachnev, HyoungJoong Kim, Sundaram Suresh, and YunQing Shi. Reversible Watermarking Algorithm with Distortion Compensation. *EURASIP Journal on Advances in Signal Processing*, 2010(1):316820, 2010. ISSN 1687-6180. doi: 10.1155/2010. URL http://asp.eurasipjournals.com/content/2010/1/316820.

Yanjiao Shi, Miao Qi, Yinghua Lu, Jun Kong, and Danying Li. Object based self-embedding watermarking for video authentication. In *International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, pages 519–522, 2011. doi: 10.1109/TMEE.2011.6199255.

Yanjiao Shi, Miao Qi, Yugen Yi, Ming Zhang, and Jun Kong. Object based dual watermarking for video authentication. *Optik - International Journal for Light and Electron Optics*, 124(19):3827–3834, 2013. ISSN 0030-4026. doi: 10.1016/j.ijleo.2012.11.078. URL http://www.sciencedirect.com/science/article/pii/S0030402613001113.

Martin Steinebach and Jana Dittmann. Watermarking-based digital audio data authentication. *EURASIP Journal on Advances in Signal Processing*, 2003:1001–1015, January 2003. ISSN 1110-8657. doi: 10.1155/S1110865703304081. URL http://dx.doi.org/10.1155/S1110865703304081.

A. V. Subramanyam, S. Emmanuel, and M.S. Kankanhalli. Robust Watermarking of Compressed and Encrypted JPEG2000 Images. *IEEE Transactions on Multimedia*, 14 (3):703–716, June 2012. ISSN 1520-9210. doi: 10.1109/TMM.2011.2181342.

David S. Taubman and Michael W. Marcellin. *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer Academic Publishers, 2004. ISBN 978-1-4613-5245-7.

Thilo Thiede, William C. Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G. Beerends, and Catherine Colomes. PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality. *Journal of the Audio Engineering Society*, 48 (1/2):3–29, 2000. URL http://www.aes.org/e-lib/browse.cfm?elib=12078.

D.M. Thodi and J.J. Rodriguez. Reversible Watermarking by Prediction-Error Expansion. In *6th IEEE Southwest Symposium on Image Analysis and Interpretation, 2004*, pages 21–25, march 2004. doi: 10.1109/IAI.2004.1300937.

H.-H. Tsai, H.-C. Tseng, and Y.-S. Lai. Robust lossless image watermarking based on α-trimmed mean algorithm and support vector machine. *Journal of Systems and Software*, 83(6):1015–1028, 2010. ISSN 0164-1212. doi: 10.1016/j.jss.2009.12.026. URL http://www.sciencedirect.com/science/article/pii/S0164121209003392. Software Architecture and Mobility.

Michiel van der Veen, Fons Bruekers, Arno van Leest, and Stephane Cavin. High capacity reversible watermarking for audio. In *SPIE Proceedings 5020, Security and Watermarking of Multimedia Contents V*, volume 5020, pages 1–11, 2003. doi: 10.1117/12.476858. URL http://dx.doi.org/10.1117/12.476858.

Ming-Shi Wang and Wei-Che Chen. A majority-voting based watermarking scheme for color image tamper detection and recovery. *Computer Standards & Interfaces*, 29(5):561–570, 2007. ISSN 0920-5489. doi: 10.1016/j.csi.2006.11.009. URL http://www.sciencedirect.com/science/article/pii/S0920548906001346.

Shuenn-Shyang Wang and Sung-Lin Tsai. Automatic image authentication and recovery using fractal code embedding and image inpainting. *Pattern Recognition*, 41(2): 701–712, 2008. ISSN 0031-3203. doi: 10.1016/j.patcog.2007.05.012. URL http://www.sciencedirect.com/science/article/pii/S0031320307002518.

G. Allan Weber. USC-SIPI image database: Version 4. Technical report, University of Southern California, 1993. URL http://sipi.usc.edu/database/database.php.

C. J. Weinstein. Opportunities for advanced speech processing in military computer-based systems. *Proceedings of the IEEE*, 79(11):1626–1641, Nov 1991. ISSN 0018-9219.

Hsien-Chu Wu and Chin-Chen Chang. Detection and restoration of tampered JPEG compressed images. *Journal of Systems and Software*, 64(2):151–161, 2002. ISSN 0164-1212. doi: 10.1016/S0164-1212(02)00033-X. URL http://www.sciencedirect.com/science/article/pii/S016412120200033X.

Xiaoyun Wu. Reversible Semi-fragile Watermarking Based on Histogram Shifting of Integer Wavelet Coefficients. In *Inaugural IEEE International Conference on Digital Ecosystems and Technologies, 2007. IEEE DEST '07*, pages 501–505, 2007. doi: 10.1109/DEST.2007.372028.

Diqun Yan and Rangding Wang. Reversible Data Hiding for Audio Based on Prediction Error Expansion. In *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 249–252, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3278-3. doi: 10.1109/IIH-MSP.2008.27.

Ching-Yu Yang, Chih-Hung Lin, and Wu-Chih Hu. Reversible Watermarking by Coefficient Adjustment Method. In *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pages 39–42, 2010. doi: 10.1109/IIHMSP.2010.17.

Xian-Ting Zeng, Ling-Di Ping, and Xue-Zeng Pan. A lossless robust data hiding scheme. *Pattern Recognition*, 43(4):1656–1667, 2010. ISSN 0031-3203. doi: 10.1016/j.patcog.2009.09.016. URL http://www.sciencedirect.com/science/article/pii/S0031320309003604.

Xinpeng Zhang and Shuozhong Wang. Fragile Watermarking With Error-Free Restoration Capability. *IEEE Transactions on Multimedia*, 10(8):1490–1499, 2008. ISSN 1520-9210. doi: 10.1109/TMM.2008.2007334.

Xinpeng Zhang, Zhenxing Qian, Yanli Ren, and Guorui Feng. Watermarking With Flexible Self-Recovery Quality Based on Compressive Sensing and Compositive Reconstruction. *IEEE Transactions on Information Forensics and Security*, 6(4):1223–1232, 2011a. ISSN 1556-6013. doi: 10.1109/TIFS.2011.2159208.

Xinpeng Zhang, Shuozhong Wang, Zhenxing Qian, and Guorui Feng. Reference Sharing Mechanism for Watermark Self-Embedding. *IEEE Transactions on Image Processing*, 20(2):485–495, 2011b. ISSN 1057-7149. doi: 10.1109/TIP.2010.2066981.

X. Zhao, A.T.S. Ho, H. Treharne, V. Pankajakshan, C. Culnane, and W. Jiang. A Novel Semi-Fragile Image Watermarking, Authentication and Self-Restoration Technique Using the Slant Transform. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007.*, volume 1, pages 283–286, 2007. doi: 10.1109/IIH-MSP.2007.50.

Xunzhan Zhu, Anthony T.S. Ho, and Pina Marziliano. A new semi-fragile image watermarking with robust tampering restoration using irregular sampling. *Signal Processing: Image Communication*, 22(5):515–528, 2007. ISSN 0923-5965. doi: 10.1016/j.image.2007.03.004. URL http://www.sciencedirect.com/science/article/pii/S0923596507000434.

D. Zou, Y.Q. Shi, and Zhicheng Ni. A semi-fragile lossless digital watermarking scheme based on integer wavelet transform. In *IEEE 6th Workshop on Multimedia Signal Processing, 2004*, pages 195–198, 2004. doi: 10.1109/MMSP.2004.1436526.

D. Zou, Y.Q. Shi, Zhicheng Ni, and Wei Su. A Semi-Fragile Lossless Digital Watermarking Scheme Based on Integer Wavelet Transform. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(10):1294–1300, 2006. ISSN 1051-8215. doi: 10.1109/TCSVT.2006.881857.