UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Applications of perceptual sparse representation (Spikegram) for copyright protection of audio signals

Applications de la représentation parcimonieuse perceptuelle par graphe de décharges (Spikegramme) pour la protection du droit d'auteur des signaux sonores

Thèse de doctorat
Specialité : génie électrique

Yousof ERFANI

Jury: Ramin PICHEVAR (Co-directeur)
Jean ROUAT (Co-directeur)
Roch LEFEBVRE
Martin BOUCHARD (examinateur externe)

Sherbrooke (Québec) Canada August 2016

To the love of my life, Shadi

RÉSUMÉ

Chaque année, le piratage mondial de la musique coûte plusieurs milliards de dollars en pertes économiques, pertes d'emplois et pertes de gains des travailleurs ainsi que la perte de millions de dollars en recettes fiscales. La plupart du piratage de la musique est dû à la croissance rapide et à la facilité des technologies actuelles pour la copie, le partage, la manipulation et la distribution de données musicales [Domingo, 2015], [Siwek, 2007]. Le tatouage des signaux sonores a été proposé pour protéger les droit des auteurs et pour permettre la localisation des instants où le signal sonore a été falsifié. Dans cette thèse, nous proposons d'utiliser la représentation parcimonieuse bio-inspirée par graphe de décharges (spikegramme), pour concevoir une nouvelle méthode permettant la localisation de la falsification dans les signaux sonores. Aussi, une nouvelle méthode de protection du droit d'auteur. Finalement, une nouvelle attaque perceptuelle, en utilisant le spikegramme, pour attaquer des systèmes de tatouage sonore.

Nous proposons tout d'abord une technique de localisation des falsifications ('tampering') des signaux sonores. Pour cela nous combinons une méthode à spectre étendu modifié ('modified spread spectrum', MSS) avec une représentation parcimonieuse. Nous utilisons une technique de poursuite perceptive adaptée (perceptual marching pursuit, PMP [Hossein Najaf-Zadeh, 2008]) pour générer une représentation parcimonieuse (spikegramme) du signal sonore d'entrée qui est invariante au décalage temporel [E. C. Smith, 2006] et qui prend en compte les phénomènes de masquage tels qu'ils sont observés en audition. Un code d'authentification est inséré à l'intérieur des coefficients de la représentation en spikegramme. Puis ceux-ci sont combinés aux seuils de masquage. Le signal tatoué est resynthétisé à partir des coefficients modifiés, et le signal ainsi obtenu est transmis au décodeur. Au décodeur, pour identifier un segment falsifié du signal sonore, les codes d'authentification de tous les segments intacts sont analysés. Si les codes ne peuvent être détectés correctement, on sait qu'alors le segment aura été falsifié. Nous proposons de tatouer selon le principe à spectre étendu (appelé MSS) afin d'obtenir une grande capacité en nombre de bits de tatouage introduits. Dans les situations où il y a désynchronisation entre le codeur et le décodeur, notre méthode permet quand même de détecter des pièces falsifiées. Par rapport à l'état de l'art, notre approche a le taux d'erreur le plus bas pour ce qui est de détecter les pièces falsifiées. Nous avons utilisé le test de l'opinion moyenne ('MOS') pour mesurer la qualité des systèmes tatoués. Nous évaluons la méthode de tatouage semi-fragile par le taux d'erreur (nombre de bits erronés divisé par tous les bits soumis) suite à plusieurs attaques. Les résultats confirment la supériorité de notre approche pour la localisation des pièces falsifiées dans les signaux sonores tout en préservant la qualité des signaux.

Ensuite nous proposons une nouvelle technique pour la protection des signaux sonores. Cette technique est basée sur la représentation par spikegrammes des signaux sonores et utilise deux dictionnaires (TDA pour Two-Dictionary Approach). Le spikegramme est utilisé pour coder le signal hôte en utilisant un dictionnaire de filtres gammatones. Pour le tatouage, nous utilisons deux dictionnaires différents qui sont sélectionnés en fonction du bit d'entrée à tatouer et du contenu du signal. Notre approche trouve les gammatones

appropriés (appelés noyaux de tatouage) sur la base de la valeur du bit à tatouer, et incorpore les bits de tatouage dans la phase des gammatones du tatouage. De plus, il est montré que la TDA est libre d'erreur dans le cas d'aucune situation d'attaque. Il est démontré que la décorrélation des noyaux de tatouage permet la conception d'une méthode de tatouage sonore très robuste.
Les expériences ont montré la meilleure robustesse pour la méthode proposée lorsque le signal tatoué est corrompu par une compression MP3 à 32 kbits par seconde avec une charge utile de 56.5 bps par rapport à plusieurs techniques récentes. De plus nous avons étudié la robustesse du tatouage lorsque les nouveaux codec USAC (Unified Audion and Speech Coding) à 24kbps sont utilisés. La charge utile est alors comprise entre 5 et 15 bps.

Finalement, nous utilisons les spikegrammes pour proposer trois nouvelles méthodes d'attaques. Nous les comparons aux méthodes récentes d'attaques telles que 32 kbps MP3 et 24 kbps USAC. Ces attaques comprennent l'attaque par PMP, l'attaque par bruit inaudible et l'attaque de remplacement parcimonieuse. Dans le cas de l'attaque par PMP, le signal de tatouage est représenté et resynthétisé avec un spikegramme. Dans le cas de l'attaque par bruit inaudible, celui-ci est généré et ajouté aux coefficients du spikegramme. Dans le cas de l'attaque de remplacement parcimonieuse, dans chaque segment du signal, les caractéristiques spectro-temporelles du signal (les décharges temporelles ;'time spikes') se trouvent en utilisant le spikegramme et les spikes temporelles et similaires sont remplacés par une autre.

Pour comparer l'efficacité des attaques proposées, nous les comparons au décodeur du tatouage à spectre étendu. Il est démontré que l'attaque par remplacement parcimonieux réduit la corrélation normalisée du décodeur de spectre étendu avec un plus grand facteur par rapport à la situation où le décodeur de spectre étendu est attaqué par la transformation MP3 (32 kbps) et 24 kbps USAC.

Mots-clés : Tatouage, Représentation parcimonieuse, Banc de filtres gammatones, Localisation d'attaque, Dissimulation de données, Masquage auditif, Attaque de remplacement.

ABSTRACT

Every year global music piracy is making billion dollars of economic, job, workers' earnings losses and also million dollars loss in tax revenues. Most of the music piracy is because of rapid growth and easiness of current technologies for copying, sharing, manipulating and distributing musical data [Domingo, 2015], [Siwek, 2007]. Audio watermarking has been proposed as one approach for copyright protection and tamper localization of audio signals to prevent music piracy. In this thesis, we use the spikegram- which is a bio-inspired sparse representation- to propose a novel approach to design an audio tamper localization method as well as an audio copyright protection method and also a new perceptual attack against any audio watermarking system.

First, we propose a tampering localization method for audio signal, based on a Modified Spread Spectrum (MSS) approach. Perceptual Matching Pursuit (PMP) is used to compute the spikegram (which is a sparse and time-shift invariant representation of audio signals) as well as 2-D masking thresholds. Then, an authentication code (which includes an Identity Number, ID) is inserted inside the sparse coefficients. For high quality watermarking, the watermark data are multiplied with masking thresholds. The time domain watermarked signal is re-synthesized from the modified coefficients and the signal is sent to the decoder. To localize a tampered segment of the audio signal, at the decoder, the ID's associated to intact segments are detected correctly, while the ID associated to a tampered segment is mis-detected or not detected. To achieve high capacity, we propose a modified version of the improved spread spectrum watermarking called MSS (Modified Spread Spectrum). We performed a mean opinion test to measure the quality of the proposed watermarking system. Also, the bit error rates for the presented tamper localization method are computed under several attacks. In comparison to conventional methods, the proposed tamper localization method has the smallest number of mis-detected tampered frames, when only one frame is tampered. In addition, the mean opinion test experiments confirms that the proposed method preserves the high quality of input audio signals.

Moreover, we introduce a new audio watermarking technique based on a kernel-based representation of audio signals. A perceptive sparse representation (spikegram) is combined with a dictionary of gammatone kernels to construct a robust representation of sounds. Compared to traditional phase embedding methods where the phase of signal's Fourier coefficients are modified, in this method, the watermark bit stream is inserted by modifying the phase of gammatone kernels. Moreover, the watermark is automatically embedded only into kernels with high amplitudes where all masked (non-meaningful) gammatones have been already removed. Two embedding methods are proposed, one based on the watermark embedding into the sign of gammatones (one dictionary method) and another one based on watermark embedding into both sign and phase of gammatone kernels (two-dictionary method). The robustness of the proposed method is shown against 32 kbps MP3 with an embedding rate of 56.5 bps while the state of the art payload for 32 kbps MP3 robust watermarking is lower than 50.3 bps. Also, we showed that the proposed method is robust against unified speech and audio codec (24 kbps USAC, Linear predictive and Fourier domain modes) with an average payload of $5 - 15$ bps. Moreover, it is shown that the proposed method is robust against a variety of signal processing transforms while preserving quality.

Finally, three perceptual attacks are proposed in the perceptual sparse domain using spikegram. These attacks are called PMP, inaudible noise adding and the sparse replacement attacks. In PMP attack, the host signals are represented and re-synthesized with spikegram. In inaudible noise attack, the inaudible noise is generated and added to the spikegram coefficients. In sparse replacement attack, each specific frame of the spikegram representation - when possible - is replaced with a combination of similar frames located in other parts of the spikegram. It is shown than the PMP and inaudible noise attacks have roughly the same efficiency as the 32 kbps MP3 attack, while the replacement attack reduces the normalized correlation of the spread spectrum decoder with a greater factor than when attacking with 32 kbps MP3 or 24 kbps unified speech and audio coding (USAC).

Keywords: Watermarking, Sparse representation, Gammatone filter bank, Tamper localization, Data hiding, Auditory masking, Replacement attack.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisors, Ramin Pichevar and Jean Rouat for all the support and encouragement they gave me. I appreciate all their contributions of time, ideas, lectures and funding during the course of this thesis. I thank Martin Bouchard, Roch Lefebvre and Eric Plourde who accepted to be the examiners of my thesis, for their valuable comments which improved the overall quality of the thesis.

During this research, I was supported financially by the National Science and Engineering Research Council of Canada (NSERC), Communication Research Center of Canada (CRC) and Université de Sherbrooke.

Many thanks to Hossein Najaf-Zadeh for fruitful discussions and for providing the perceptual matching pursuit code used in this project.

Special thanks to Frédérick Lavoie and Roch Lefebvre for managing the listening test and to the participants of the listening test. Many thanks to Philippe Gournay for providing the USAC codec.

Thanks to Malcom Slaney for providing the gammatone filter bank source code used in this project, and to Christopher J. Rozell for the locally competitive algorithm (LCA) code.

I also thank the members of the NECOTIS research group that have been a source of good advice and collaboration. I have had the pleasure to work with or alongside of them. I am grateful to my wife, Shadi, for her love, support and patience throughout my PhD.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronyme	Definition
AW	Audio Watermarking
BER	Bit Error Rate
CD	Compact Disk
CF	Center Frequency
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
HAS	Human Auditory System
ITH	Iterative Hard Thresholding
ID	Identity Number
ISS	Improved Spread Spectrum
LFSR	Linear Feedback Shift Register
LCA	Linear Competitive Algorithm
LPF	Low Pass Filtering
MP	Matching Pursuit
ML	Maximum Likelihood
MSE	Mean Square Error
MSS	Modified Spread Spectrum
PN	Pseudo Noise
PMP	Perceptual Matching Pursuit
QIM	Quantization Index Modulation
STFT	Short Time Fourier Transform
SS	Spread Spectrum
SNR	Signal to Noise Ration
SDMI	Secure Digital Music Initiative
SSW	Spread Spectrum Watermarking
SPL	Sound Pressure Level
TDA	Two Dictionaries Approach
USAC	Unified Speech and Audio Coder
VOIP	Voice Over IP

CHAPTER 1

INTRODUCTION

With the Internet growth, unauthorized copying and distribution of digital media (audio, image, video) has never been easier. Consequently, the music industry claims a multi-billion dollar annual revenue loss due to piracy [Domingo, 2015],[Siwek, 2007] which is likely to increase due to cloud and peer-to-peer file sharing Web communities. This includes 58.0 billion dollars of economic loss, 373,375 jobs loss, 16.3 billion dollars in workers earning loss and 2.6 billion dollars in tax revenues annually.

Technological advances on data processing can help providing ways of enforcing copyright on the Internet. For copyright protection, traditional data protection methods such as scrambling or cryptography cannot be used. The reason is that only the authorized key holders can decrypt the encrypted data and once the data is decrypted to the original form, they can always be re-recorded and then freely distributed since there is no way to track their reproduction or retransmission [Yiqing Lin, 2014].

A promising solution to this problem is to insert a mark into the media signal with a secret, robust and imperceptible watermark. The media player at the client side can detect this mark and consequently enforce a corresponding E-commerce policy.

Generally, embedding of a secure mark into a media (so that it can not be removed easily while preserving the media quality) is called watermarking. Since, copyright protection of multimedia signals is still in demand for many applications from industry and governments, it can be a promising research direction.

1.1 Audio watermarking

Digital watermarking has been proposed as a method to enforce the intellectual property rights and protect digital media from tampering. It involves a process of embedding into a host signal a visible (only in the case of image and video) or perceptually invisible (transparent) digital signature. This digital signature might carry a message about the host signal to mark its ownership or may contain the identity information of the purchaser of the host signal. An authorized receiver (detector) which has a key should be able to extract the watermark with high accuracy. Digital watermarking can be used in various applications, including digital rights management and tamper proofing. Although perceptually transparent, the existence of the watermark is indicated when watermarked media is passed through an appropriate watermark detector [Yiqing Lin, 2014]. Figure 1.1 gives an overview to the general audio watermarking system which comprises the following blocks [Wu, 2015]:

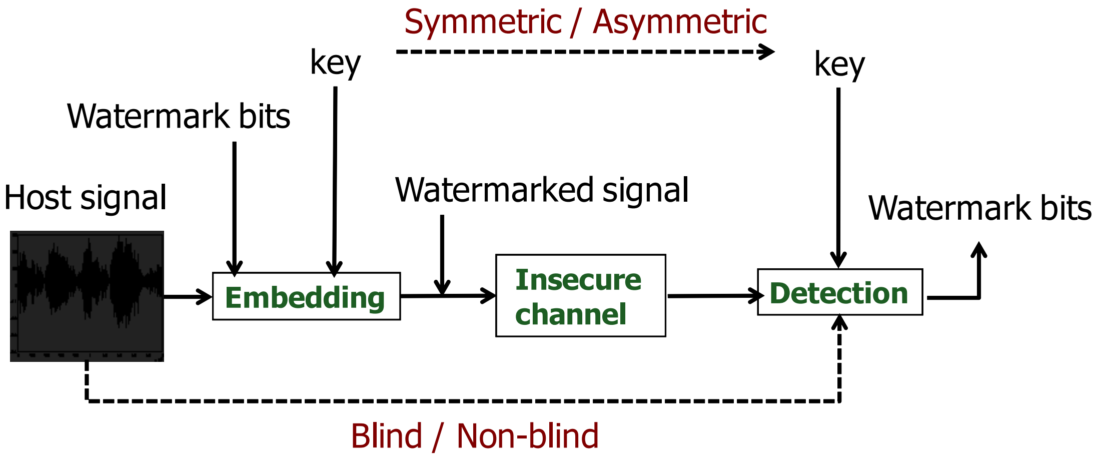

Figure 1.1 Representing a watermarking system via a communication channel. At the embedding block, the watermark bits are inserted into the signal by using a key stream and the watermarked signal is passed through an insecure channel before reception at the detector. The detector block uses a key stream again to confirm the presence of a watermark or extract the inserted watermark bits. When the decoder and the encoder use the same key, the watermarking is of symmetric type otherwise it is of asymmetric type. If the original non watermarked signal is not required for the decoding process, the watermarking is blind, otherwise it is non-blind.

1. **The watermark embedder:** The watermark embedder has three inputs: one is the **watermark message** which usually consists of a binary data sequence, inserted into the host signal. The second input is a **secret key** which can be shared with the decoder. The third input is the **host signal** (image, video clip, audio sequence). The output of the watermark embedder is the **watermarked signal**, in which the watermark message is hidden, and (hopefully) should not be perceptually discriminated from the host signal.
2. **The insecure channel:** The watermarked signal passes an insecure channel before arriving at the receiver. The name insecure indicates the presence of either intentional or unintentional attacks-transforms which may result in removing the watermark from the signal or destroying it.
3. **The watermark decoder:** There are two types of decoders for the watermarking systems: zero-bit and multi-bit watermarking. In zero-bit watermarking, the decoder

should confirm whether there is a mark in the host signal or not. While in multi-bit watermarking, the decoder should extract a bit stream from the watermarked signal [Cox *et al.*, 2007]. In both cases, the decoder inputs include the watermarked signal (which has passed the insecure channel) and a key stream.

It is good to mention that for the watermarking system of Figure 1.1, the host signal acts as a carrier of the watermark. However, as the goal is to extract the watermark from the watermarked host signal, sometimes, the host signal can be the main cause of erroneous detection at the detector.

1.2 Research motivations

Audio watermarking systems are in great demand for the goal of copyright protection and ownership authentication [Domingo, 2015].

Although there have been many researches on audio watermarking methods and attacks, there are still difficulties in designing high quality- high payload watermarking systems for copyright and tampering localization applications. The main reasons are as follows:

- The quality of audio systems changes with even very small additional watermark. The human auditory system (HAS) perceives over a range of power greater than one billion to one and a range of frequencies greater than one thousand to one [Kale *et al.*, 2010]. Sensitivity to additive random noise is also acute. Thus this sensitivity to additional noise is a big challenge against designing high payload watermarking systems for copyright protection and tamper localization applications. Hence, there is a need for examining novel high resolution time-frequency representations for audio signals that find the best positions in the time-frequency domain to insert inaudible watermarks. Also, better time-frequency representations help to shape the inaudible watermark noise under better masking curves.
- The amount of data that can be embedded transparently into an audio sequence is considerably lower than the amount of data that can be hidden in video sequences. On the other hand, audio signals have perceptual properties that are specific. However, most current audio watermarking systems for copyright protection and tamper localization use the same old methodologies borrowed from image watermarking. Hence there is a need to propose new frameworks specific to audio watermarking based on state of the art tools and representations.
- One important application of audio watermarking is tamper verification in which the whole signal is classified as tampered or not [Hua *et al.*, 2016],[Ho and Li, 2015].

There are methods that find the position of the tampering in the audio signals based on block-based representations. However, still there is no efficient methods to find the location of short-time tampering while smallest modifications (shorter than half a second) on a speech signal might drastically change the meaning of the signal.

- Another challenge for audio watermarking is that benchmarks to asses the robustness are based on conventional attacks that do not reflect the attack efficiency of more recent coding techniques. Hence there is a need to verify the efficiency of watermarking systems against new attacks and propose new attack benchmarks against audio watermarking systems.

1.3 Applications and Research objectives

The general applications for this project include:

- audio tamper localization: where any modifications on the signal by attackers are recognized and localized.
- audio fingerprinting: where each audio signal (e.g. music) is characterized with a specific ID (e.g. an ASCII code), and this ID can be embedded into the signal using the proposed methods.
- copyright protection: where the information of the owner of the audio signal is embedded into and extracted from the signal.
- designing a covert channel: where the hidden information can be transferred using the proposed watermarking methods.
- recently emerged second screen applications [Arnold *et al.*, 2014]: where the watermark is inserted into the audio section of the video being played on a TV screen. This watermark can be extracted by cell phone devices for advertisement applications
- designing new attacks on audio watermarking using perceptual sparse representation and comparing them with the state of the art attacks and coding transforms.

There are also some specific research goals including

- investigating the efficiency of masking thresholds obtained from sparse representation [Pichevar *et al.*, 2011], for audio watermarking.
- proposing strategies to embed watermark into the phase of kernels representing the audio signals.
- investigating the efficiency of gammatone kernels for phase modulation watermarking.

— designing a fast projection based decoder for audio watermarking based on a non-linear, non-orthogonal sparse representation (with matching pursuit and a highly correlated gammatone dictionary).

— improving the spread spectrum technique [Xu *et al.*, 2016] for watermarking applications.

1.4 Evaluation and experimental setup

For each audio watermarking application, there are different sets of measures. In this thesis, we focus on the watermarking methods with the following properties:

1. **Symmetric:** where the decoder shares the same key as the embedder for extracting the watermark bits from the signal, otherwise the method is called asymmetric-key watermarking.

2. **Blind:** where the decoder does not require the presence of the original signal for watermark extraction. Otherwise the system is called non-blind. In non-blind watermarking, the presence of the original signal is also required at the decoder, thus the required bandwidth is higher than the required bandwidth with blind watermarking.

3. **Transform Domain:** where the watermark insertion is performed on the time-frequency representation of the signal. In this thesis, we insert watermarks in the spikegram which is a perceptual sparse representation of signal.

4. **Robust and semi-fragile:** There can be three different definitions for the security of a watermarking system: If the goal is to protect the watermark inside the signal against any intentional or unintentional task, the watermarking system is called robust. If the goal is to design a watermarking system which is very sensitive to small changes on the watermarked signal (so that the watermark is removed or undetectable after the watermarked signal passes any little intentional or unintentional changes), then the watermarking system is called fragile. Fragile watermarks are commonly used for tamper detection (integrity proof).

   A watermark is called semi-fragile, if it resists benign transformations but fails detection after malignant transformations. Semi-fragile watermarks are commonly used to detect malignant transformations. In this thesis, both semi-fragile (in chapter 3) and robust (in chapter 4) watermarking systems are designed.

5. **Irreversible:** If the decoder is able to find both watermark and the original signal, then the watermarking is called reversible watermarking. While if the detector is only able to verify the presence of watermark or the value of watermark stream, then the watermarking is of irreversible type.

   One application of reversible watermarking is for **telemedicine** [Singh *et al.*, 2015], where the patient is at home and the information regarding his health is watermarked and sent to the hospital. In this case, the decoder is able to remove the mark and find the original signal. In this thesis, only irreversible watermarking methods are designed.

Specifically, to evaluate the proposed watermarking methods, we use the following criteria and experimental setups.

1. **Bit Error Rate (BER):** It is computed as the number of erroneous detected bits at the decoder divided by the total number of detected bits. This is done for all signals used in the experiments and the average BER is calculated for each specific application and under different attacks.
2. **Unobtrusiveness:** The watermark should be perceptually invisible, or its presence should not interfere with the multimedia data being protected. In this thesis, we use MOS [ITU, 1996] and ABC/HR [ITU, 1997] tests for the quality evaluation of the audio signals before and after the watermark insertion.
3. **Payload:** In the case of multi-bit watermarking, the watermarking system should have a high embedding bit rate. For the case of zero-bit watermarking, the decoder verifies the presence of a watermark and does not extract a bit stream.

   There is a trade off between the payload, quality and robustness. Usually, the larger the insertion, the greater is the degradation of the signal's quality and and the lower is the robustness.

   Experimentally, the payload is calculated as the number of inserted bits in each second of the audio signal.

4. **Robustness tests:** To test the robustness of the proposed methods, we perform the following attacks on the watermarked signals:

   (a) **Common signal processing attacks:** These attacks include re-sampling (44.1 kHz, 22 kHz, 11.025 kHz), re-quantization (12 bits, 8 bits) and MP3 compression

(64 kHz, 32 kHz);

(b) **Common Synchronization attacks (for audio):** There are two kinds of synchronization attacks: In the first type the goal is to misalign the starting points of watermarked blocks in the decoder in relation to watermarking blocks in the encoder. In this thesis, for evaluating the robustness of the proposed methods against this type of attack, we insert samples into or remove random samples from watermarked audio segments. For example, a garbage clip can be added to the beginning of the audio signal. The second type of synchronization attack is time-scaling or pitch-scaling, which can done by malicious attackers. In this thesis, the time and pitch scaling attacks are performed on the watermarked signals for the scaling range between 90-110 percent. Note that, based on International Federation of Phonographic Industry [Katzenbeisser and Petitcolas, 2000], to consider time rescaling as an attack, the acceptable scaling factor range is between 90% and 110%.

(c) **Subterfuge attacks; collusion and forgery:** The watermark should be robust to collusion by multiple individuals, who each of which possess a watermarked copy of the data. It must be impossible for colluders to combine their signals to generate a different valid watermark with the intention of framing a third party. For this goal, in this project, we design key based watermarking systems. The linear feedback shift register (LFSR)[Klein, 2013a] is used to generate keys. In this case, every watermark is recognized by its own key. Thus the new manipulated watermark by colluders is not assigned with a key, hence is not detected at the decoder. Note that the whole algorithm is known to the attackers and the watermarking key is only shared between the encoder and the decoder.

1.5 Contribution

In this work, three applications are developed based on the proposed designs of audio watermarking in sparse domain. First, an audio tamper localization method is developed based on a semi-fragile audio watermarking. A new version of spread spectrum watermarking is proposed, the sparse auditory masking is applied, and an encoding-decoding method for tamper localization is proposed.

Second, a robust audio watermarking is proposed in the sparse domain based on the phase

modulation of the gammatone kernels in the sparse representation. This method uses two dictionaries, based on the input watermark bit and the signal content.
Lastly, a perceptual attack is developed in the sparse domain. This attack is based on the substitution of the perceptually similar content of the audio signal obtained using the sparse representation.

1.6 Originality of the research

In this thesis, we use the spikegram which is a perceptual sparse representation for designing audio watermarking algorithms and attacks. In the field of sparse representation, one method has been proposed for image watermarking [Sheikh and Baraniuk, 2007], in which the input signal is represented with sparse coefficients and the watermark is whitened by a whitening matrix. The obtained watermark is added to the sparse coefficients of the signal. At the receiver side, the watermark and the host signal are found using the compressive sensing $L1$ algorithm [Candes and Tao, 2005]. Because of the instability of compressive sensing algorithm [Candes and Tao, 2005] against noise, the mentioned algorithm lacks robustness against noise adding attacks.
Another method has been proposed in [Parvaix *et al.*, 2008], in which they used the concept of molecular matching pursuit [Daudet, 2006] to represent the signal sparsely. They find the masking thresholds using the psychoacoustic model for the resulting spikes and add the mark as a shaped spread spectrum random noise based on the masking threshold. The main inefficiency of their algorithm is that it is a **non-blind watermarking** system and the original signal is also required at the decoder. This method requires a channel with higher capacity compared to blind watermarking methods.

1.7 Scientific publications

The major findings of this thesis are included in the following scientific papers:

- Y. Erfani, R. Pichevar, J. Rouat, "Audio watermarking using spikegram and a two-dictionary approach", submitted to IEEE Transactions on Forensics, March 2016.
- Y. Erfani, R. Pichevar, J. Rouat, "Audio tampering localization using masking-aware ISS watermarking in the sparse domain", submitted to IEEE Transactions on Audio, Speech and Language Processing, April 2016.

- Y. Erfani, R. Pichevar and J. Rouat, “Audio tampering localization using modified ISS watermarking in sparse-domain,” Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE, Austin, TX, 2013, pp. 249-252.

1.8 Outline of the thesis

The outline of the thesis is as follows:

Chapter 2 introduces the spikegram which is a perceptual sparse time frequency representation. The spikegram is created using the perceptual matching pursuit [Pichevar *et al.*, 2011] by applying the gammatone dictionary (obtained from gammatone filter bank [Slaney, 1998a]). Finally, interesting characteristics of spikegram sparse representations for audio watermarking are shown via experimental results.

In chapter 3, a novel audio tampering localization is proposed in the sparse domain. The perceptual matching pursuit is explored as a tool for sparse representation and an embedding method called modified spread spectrum is introduced. The experimental results show the efficiency of the proposed tamper localization against traditional methods of tamper localization.

Chapter 4 presents a new framework for audio watermarking called two-dictionary method (TDA). The designs of the encoder and the decoder are presented. The efficiency of the method is shown compared to the most recent methods in audio watermarking via experimental results.

Chapter 5 demonstrates a new attack called perceptual sparse replacement attack on audio watermarking systems. The strength of this attack is evaluated against 32 kbps MP3 and 24 kbps USAC (unified speech and audio coding) on the signals watermarked using the spread spectrum method.

The thesis is concluded in chapter 6.

CHAPTER 2

RESEARCH BACKGROUND

2.1 Introduction

In this chapter, we briefly discuss the state of the art of spectro-temporal representations used for audio watermarking. Then, a sparse representation is introduced and the spikegram is generated from the sparse representation. Finally, a list of desirable characteristics of the spikegram for audio watermarking are mentioned.

2.2 State of the art of spectro-temporal representations for audio watermarking

So far, several methods have been presented on audio watermarking in time, spectral or spectro-temporal domains including spread spectrum watermarking [Xiang *et al.*, 2015], [Kuribayashi, 2014], echo hiding [Hua *et al.*, 2015], quantized index modulation (QIM) watermarking [Wang *et al.*, 2014] and patchwork algorithm [Xiang *et al.*, 2014b]. Time domain methods are easy to implement, have less computational cost but they are less robust against signal processing transforms performed in the spectral domain.
So far the following methods have been used to transform the signal into the spectro-temporal domain for the goal of audio watermarking:

1. **Block-based methods:**
   There are block-based spectro-temporal domain algorithms which segment the signal and find the spectral coefficients using the fast Fourier transform (FFT) or discrete cosine transform (DCT) for each block and then add watermark bits into each block. In one work, FFT transform of the signal has been used to compute time and frequency masking thresholds. These masking thresholds are used to shape a spread spectrum stream for audio watermarking [Xiang *et al.*, 2014a].

2. **Wavelet-based methods:** Many spectro-temporal domain audio watermarking systems have been proposed recently especially in the wavelet domain. These sys-

tems benefit from the multi resolution property of the wavelet representation. These systems usually embed the mark in the low frequency wavelet levels of the signal, where the mark can not be removed easily [Ratanasanya *et al.*, 2005], [Fallahpour and Megías, 2010], [Chen *et al.*, 2015], [Lei *et al.*, 2013].

2.3 Sparse representation

Sparse representation is a bio-inspired method that mimics the way the neurons transfer the information from sensory systems of mammals to the cortex [Pichevar *et al.*, 2011], [Smith and Lewicki, 2006]. It is shown that although in the mammalian brain, there are many neurons available to carry information from the sensory inputs, only a very few sparse number of neurons are selected to characterize the stimulus [Hromádka *et al.*, 2008], [Smith and Lewicki, 2006]. In sparse coding, the audio signal is represented with a dictionary of bases while small number of these bases have high value coefficients. This is similar to the transmission of a stimulus when only a small number of neurons fire among a huge pool of neurons, to transmit (or encode) a stimulus. [Chui and Montefusco, 2014], [Pichevar *et al.*, 2010b], [Smith and Lewicki, 2006], [Rozell *et al.*, 2008] [Blumensath and Davies, 2008].

In sparse representation, an efficient dictionary of bases has to be built where each base is localized around a center frequency and a time sample. Then, the acoustical signal $x[n]$ can be represented as a weighted sum of these bases as below:

$$x[n] = \sum_{i=1}^{M} \alpha_i g_{c_i}[n - l_i] \tag{2.1}$$

where α_i is a sparse coefficient, $g_{c_i}[n - l_i]$ is a base located at time sample l_i and channel number c_i. To have high sparsity, a large percentage of coefficients in (2.1) have to be zero. One challenge is to learn the dictionary (find the basis) that maximize the sparsity of the coefficients.

The amount of sparsity depends on the type of dictionary and the optimization algorithm we use to find the coefficients. The dictionary matrix Φ includes all bases of the dictionary as it columns. We can associate all bases in the dictionary to specific points in the time-channel plane. Thus, the sparse representation algorithm takes the original signal $x[n]$ ($\boldsymbol{x}$ in vector format) and the dictionary matrix Φ as the input and gives the sparse coefficient vector $\boldsymbol{\alpha}$ as the output. The re-synthesized input signal is the multiplication of the dictionary matrix by the sparse coefficient vector, i.e $\boldsymbol{x} = \Phi \times \boldsymbol{\alpha}$ (see Fig.2.1).

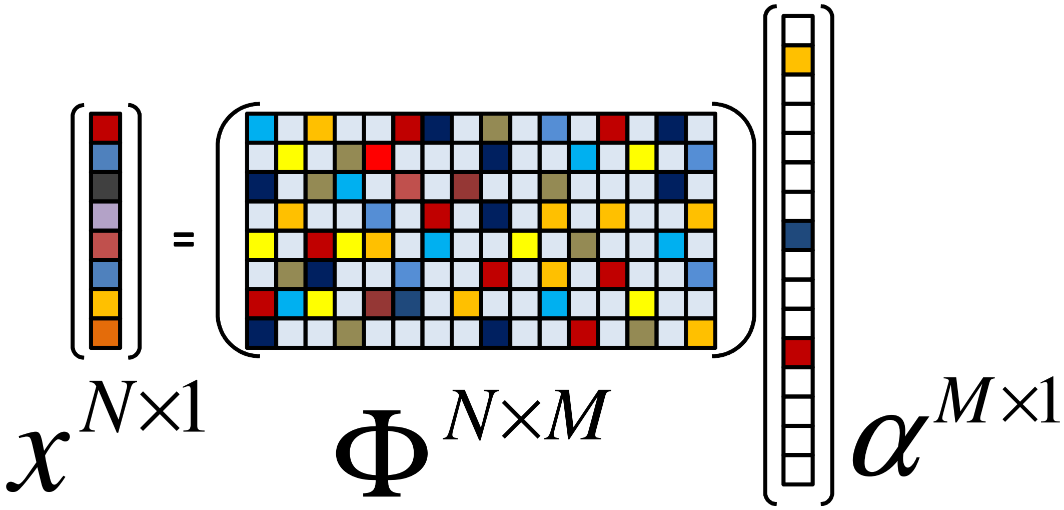

Figure 2.1 The sparse representation of a signal $\boldsymbol{x}$ over a dictionary $\Phi \in R^{N\times M}$ to achieve a sparse coefficient vector $\boldsymbol{\alpha}$. The dictionary Φ can be undercomplete ($M < N$), complete ($M = N$) or overcomplete ($M > N$). The sparsity of the coefficient vector $\boldsymbol{\alpha}$ varies based on the selection of the dictionary.

The kernel based representation in (2.1), can be modeled as the signal is transfered by some neurons (kernels). In this case, the neuron's receptive field is the kernel wave and when the amplitude is zero, the neuron is not spiking. The main difference between the representation in (2.1) and traditional ones, is that first it is a kernel based representation and also we use a non linear optimization method along with a bio-inspired dictionary of kernels to find a sparse coefficient vector. Thus, we have highest energy for small number of kernels [Smith and Lewicki, 2006]. After finding the sparse coefficients, we will have a few bases (similar to the receptive filed of neurons) from the dictionary matrix with high value coefficients (or equivalently firing neurons), while so many other coefficients are around zero (not firing). Therefore, we can show their amplitude in color, in a time-channel representation, based on the position of the bases in the sampled dictionary and consider the result as a spikegram representation. The obtained representation is called spikegram because it shows whether a base (similar to the characteristic of the receptive filed on a neuron) is active (spiking) or not. Thus, the position of non zero kernels in the spikegram visualizes the active and inactive spikes (neurons) (see Fig.2.2).

This representation can have its own specific characteristics which might make it superior to other representation methods for audio watermarking applications. It is time shift invariant, has high resolution and is compact.

To design spikegram for audio and speech signals, the gammatone/gammachirp dictionary has been suggested as the dictionary which mimics at best the auditory filter bank of mammals [Strahl and Mertins, 2009]. Also, the gammatone/gammachirp has been

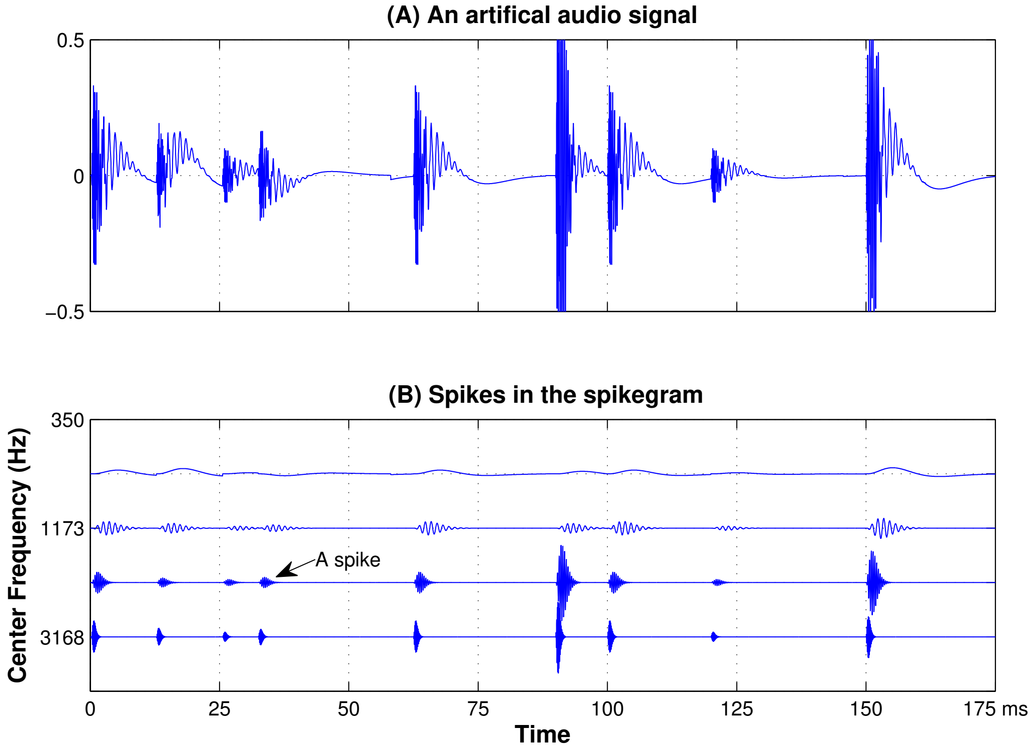

Figure 2.2 A simple model which shows the place of spikes in the spikegram. The signal in (A) is represented by spikegram in (B). Each spike is a gammatone kernel with a specific center frequency located at a specific time (time sample). High amplitude spikes have more contributions in generating the audio signal.

confirmed as a dictionary which is adapted to the content of natural sounds. This means that the best learned kernels, which represent the natural sound sparsely, are gammatones [Smith and Lewicki, 2006].

For sparse representation of audio signals, several methods are proposed in the literature including matching pursuit [Mallat and Zhang, 1993], [Chui and Montefusco, 2014], perceptual matching pursuit (PMP) [Pichevar *et al.*, 2011], [Najaf-Zadeh *et al.*, 2008], a bio-inspired algorithm called locally competitive algorithm between neurons of the receptive field (LCA) [Rozell *et al.*, 2008], [Pichevar *et al.*, 2011] and also iterative hard thresholding algorithm (ITH) [Blumensath and Davies, 2008].

In [Siahpoush *et al.*, 2015], the efficiency of spikegram, as a bio-inspired representation of acoustical signals has been shown when performing neural decoding. In that work, an approximate spikegram representation of the auditory stimuli is reconstructed from the neural activity recorded from the inferior colliculus of the Guinea pig.

2.4 Building a sample spikegram from an overcomplete sparse representation

In this chapter, to explain the reasons of choosing sparse representation for audio, we design a sample sparse representation for audio signals using the gammatone dictionary and perceptual matching pursuit as the optimization algorithm. Firstly, we briefly explain the gammatone dictionary and the PMP to show, how a spikegram can be generated and then we explain its characteristics.

2.5 Building a gammatone dictionary

A gammatone filter equation [Slaney, 1998a] has a gamma part and a tone part as

$$g[n] = an^{m-1}e^{-2\pi ln}cos[2\pi(f_c/f_s)n + \theta] \tag{2.2}$$

in which, $an^{m-1}e^{-2\pi ln}$ is the gamma part and the rest of the equation is the tone part. The gamma part controls the time envelope of the kernel. Also, n is the time index, m and l are used for tuning the gamma part of the equation. f_s is the sampling frequency, θ is the phase, f_c is the center frequency of the gammatone. The term a is the normalization factor to set the energy of each gamatone to one. A gammatone filterbank includes a set of M gammatone kernels with different center frequencies where their bandwidth altogether cover the hearing frequency range of human. Gammatone filterbank is bio-inspired and has been shown to be adapted to the natural sounds [Smith and Lewicki, 2005]. Gammatone kernels are shown to be efficient for sparse representation [Pichevar *et al.*, 2010b]. In this thesis, each gammatone filter is considered as a spectro-temporal kernel for sparse representation. To represent an audio signal with a spikegram, we generate N_c gammatone channels with different center frequencies [Slaney, 1998a]. These gammatone atoms overlap in frequency and cover the frequency range from 20 Hz up to the half of the sampling rate of the input signal (Fig.2.3).

The gammatone dictionary includes the repetition of the gammatone filter bank along the time axis. Hence the gammatone dictionary includes gammatone bases at different time samples and channel numbers (center frequencies). To generate the 2-D dictionary (time-channel plane) Φ, a base g (here gammatone [Slaney, 1998a]) is modulated to be located at N_c center frequencies (channels) c_i and T_s time shifts τ_i. ($M = N_cT_s$). In this case, the spikegram is represented with a 2-D image which shows the sparse coefficients

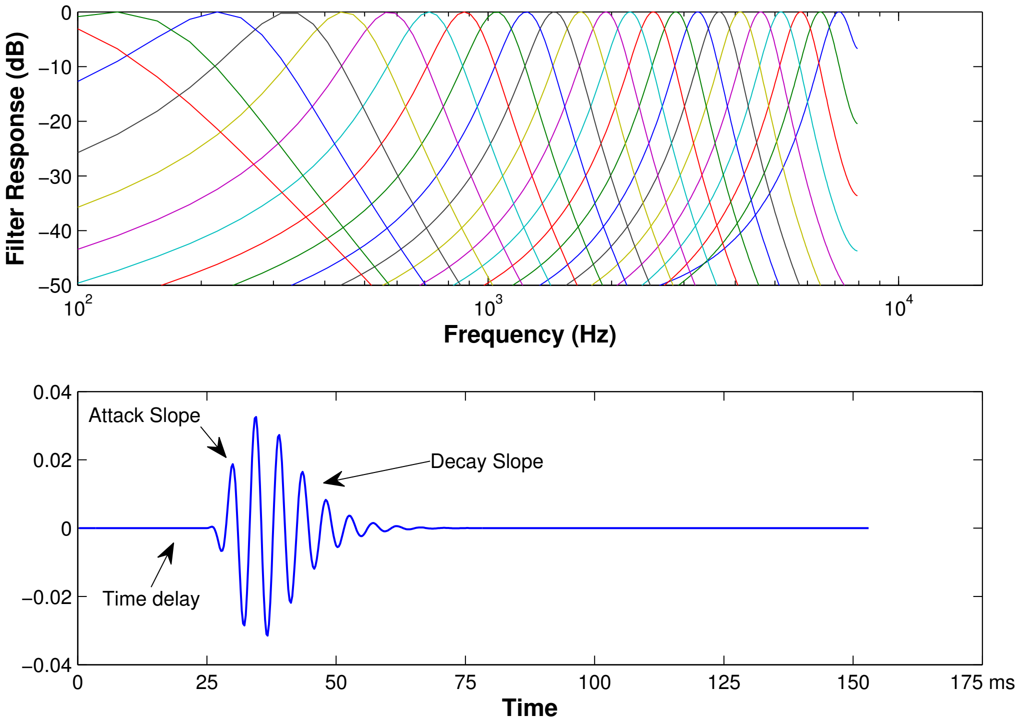

Figure 2.3 **Top:** An N channel gammatone filter bank which covers the hearing frequency range of the human. **Bottom:** A gammatone kernel in time: It has a time delay, an attack and a decay slope [Pichevar *et al.*, 2010b].

associated to the gammatone bases at each time sample-channel point (Fig.2.2).
Each point of the spikegram plane represents a specific center frequency and a specific time delay for a gammatone kernel (atom) (Fig.2.2). At each point, the amplitude of each gammatone indicates the contribution of each atom of the spikegram in generating the audio signal. For coefficients with greater values, we have greater amplitudes for their associated gammatone kernels. Here, as the number of bases is greater than the signal's length, thus we have an over-complete dictionary.

2.6 Perceptual matching pursuit (PMP) algorithm

Perceptual matching pursuit (PMP) [Pichevar *et al.*, 2011], [Najaf-Zadeh *et al.*, 2008] is a perceptual sparse representation method. In PMP, the gammatone dictionary is generated as described in section 2.5. PMP is an iterative method similar to Matching Pursuit (MP) [Mallat and Zhang, 1993; Chui and Montefusco, 2014]. At each iteration i, PMP finds a coefficient α_i and a masking threshold m_i for a gammatone basis $g_{c_i}[n-l_i]$ at time position

l_i, channel number c_i in the 2-D time-channel plane. Hence the signal $x[n]$ is represented as

$$x[n] = \sum_{i=1}^{M} \alpha_i g_{c_i}[n - l_i] \tag{2.3}$$

Compared to MP, in PMP masking thresholds $\{m_i, i{=}1{:}\mathrm{M}\}$ are progressively generated and updated along the signal decomposition. At each iteration i of PMP, the projections of the signal onto the dictionary are computed and a kernel $g_{c_i}[n{-}l_i]$ with the maximum projection is selected as the basis for the current iteration (see Fig.2.4). The sparse coefficient α_i is set to the maximum projection. Also, a masking threshold m_i is computed for the current kernel and all other projections are compared against that masking threshold (the right part of (2.4)) and those values below that masking threshold are set to zero. In *PMP*, at each iteration i, a sensation level ($SL_k(i)$ in dB) is computed, where,

$$SL_k(i) = 10Log_{10}(\frac{(\alpha_i G_k)^2}{QT_k}) \tag{2.4}$$

k is the critical band number (the channel number c_i) for the gammatone kernel found at iteration i. G_k is the peak value of the Fourier transform of the normalized base in critical band k and QT_k is the elevated threshold in quiet for the same critical band [Najaf-Zadeh *et al.*, 2008]. A selected base is considered audible (*audible kernel*) if it induces a sensation level ($SL_k(i)$ in dB) greater than its associated masking threshold m_i.

Finally, PMP generates a residual signal $r[n]$ to be considered as the input signal for the next iteration. By setting the residual signal at the first iteration equal to the input signal $x[n]$, then the residual signal at iteration i equals the residual signal of the previous iteration minus $\alpha_i g_{c_i}[n - l_i]$.

Thus, the PMP finds only audible bases that their sensation levels are above their masking thresholds and neglects the rest. Hence, PMP algorithm finds a progressive masking thresholds for all gammatones in the representation which can be used for watermarking applications.

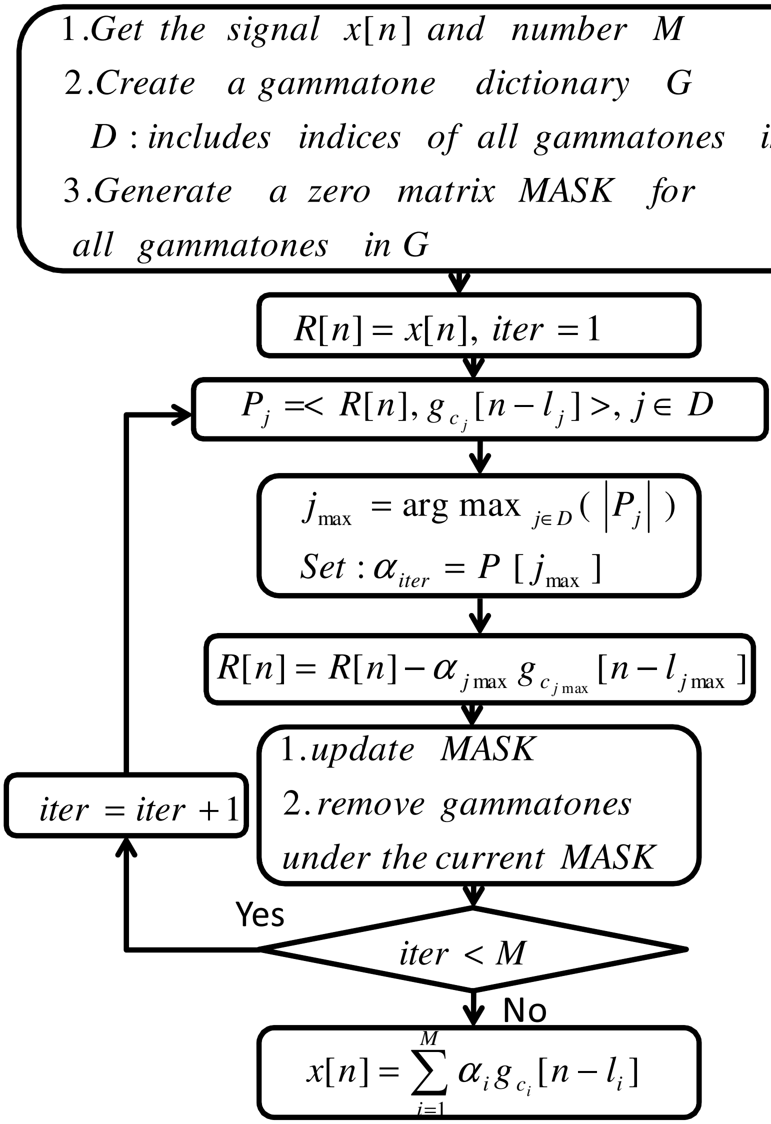
1.Get the signal $x[n]$ and number M
2.Create a gammatone dictionary G
D : includes indices of all gammatones in G
3.Generate a zero matrix MASK for all gammatones in G
$R[n] = x[n]$, $iter = 1$
$P_j =< R[n], g_{c_j}[n - l_j] >, j \in D$
$j_{\max} = \arg\max_{j \in D} (|P_j|)$
Set : $\alpha_{iter} = P\ [j_{\max}]$
$R[n] = R[n] - \alpha_{j\max}\, g_{c_{j\max}}[n - l_{j\max}]$
1.update MASK
2.remove gammatones under the current MASK
$iter = iter + 1$
Yes
$iter < M$
No
$x[n] = \sum_{i=1}^{M} \alpha_i g_{c_i}[n - l_i]$

Figure 2.4 The block diagram of the perceptual matching pursuit. First, it receives the input signal $x[n]$. It generates a dictionary G of M gammatone kernels. G is a matrix, in which each column j includes a gammatone $g_{c_j}[n - l_j]$ located at the time sample l_j and channel number c_j. D= $\{1, ..M\}$ is a set incluing all the gammatone indices in the dictionary. Also, a matrix $MASK$ with the size of the dictionary G is initialized to zero elements and is called the masking matrix. At the first iteration $iter = 1$, the residual signal $r[n]$ is set to the input audio signal $x[n]$. Then, correlations between the residual signal and all gammatones in the dictionary are computed and the gammatone with the maximum correlation with the residual signal is chosen as the selected gammatone of the current iteration. The sparse coefficient of the current iteration α_{iter} is set to the maximum correlation. Then the residual signal is updated. The masking matrix is updated and any gammatone in G under its associated masking threshold in $MASK$ is ignored (set to zero) for the upcomming iteration. The algorithm continues until reaching the maximun number of iterations.

2.7 Representing a speech signal using the spikegram

The spikegram is defined as the image plot of the module of the sparse coefficients in the 2-D time-channel domain. In this section, the PMP is used to generate the spikegram of a sample speech signal. For having a better insight about the PMP, the PMP spikegram is compared with the spectrogram and also with a spikegram obtained from another sparse representation algorithm called locally competitive algorithm (LCA). LCA is a bio-inspired algorithm which applies biological neuron models for the sparse representation [Rozell *et al.*, 2008]. In the rest of the thesis, as PMP can be used to compute masking thresholds, only this algorithm is used for making spikegrams.

In Fig.2.5, the plot at the top of figure shows the color bar for displaying the spectrogram and spikegram. The second plot from the top shows the original time domain speech signal of the excerpt “Une Fourchette” sampled at 44.1 kHz. This signal is uttered by a female speaking in French. The third plot from the top shows the spectrogram of the signal using Hamming window with 128 samples length and 50% overlapping. The third and the fourth plots from the top are respectively the spikegram of the signal using LCA and PMP. For the LCA, the algorithm is run for 100 iterations. For both PMP and LCA, a 25-channel gammatone filter bank is used and the number of time shifts equal as 1/10 of the length of signal.

To compute the STFT, we tried different window lengths. The spikegram is a multi-resolution representation, it can show the time-frequency contents of the signal with high resolutions. The spikegram uses a filter bank of gammatone kernels which are adapted to the signal content. Also, the sparse representation algorithm is adaptive and find the best coefficients and kernels for representing the signal.

As is seen in Fig.2.5, the spikegram obtained by LCA is sparse and clearly demonstrates the spectro-temporal content of the signal. This is because in the cost function of the LCA, the regularization term is the l^1 norm which is similar to coefficient estimation using maximum a posteriori (MAP) with Laplacian prior. This is because Laplacian prior on the coefficients, compared to no prior or Gaussian prior, will results in a sparser coefficient estimation [Fevotte *et al.*, 2006].

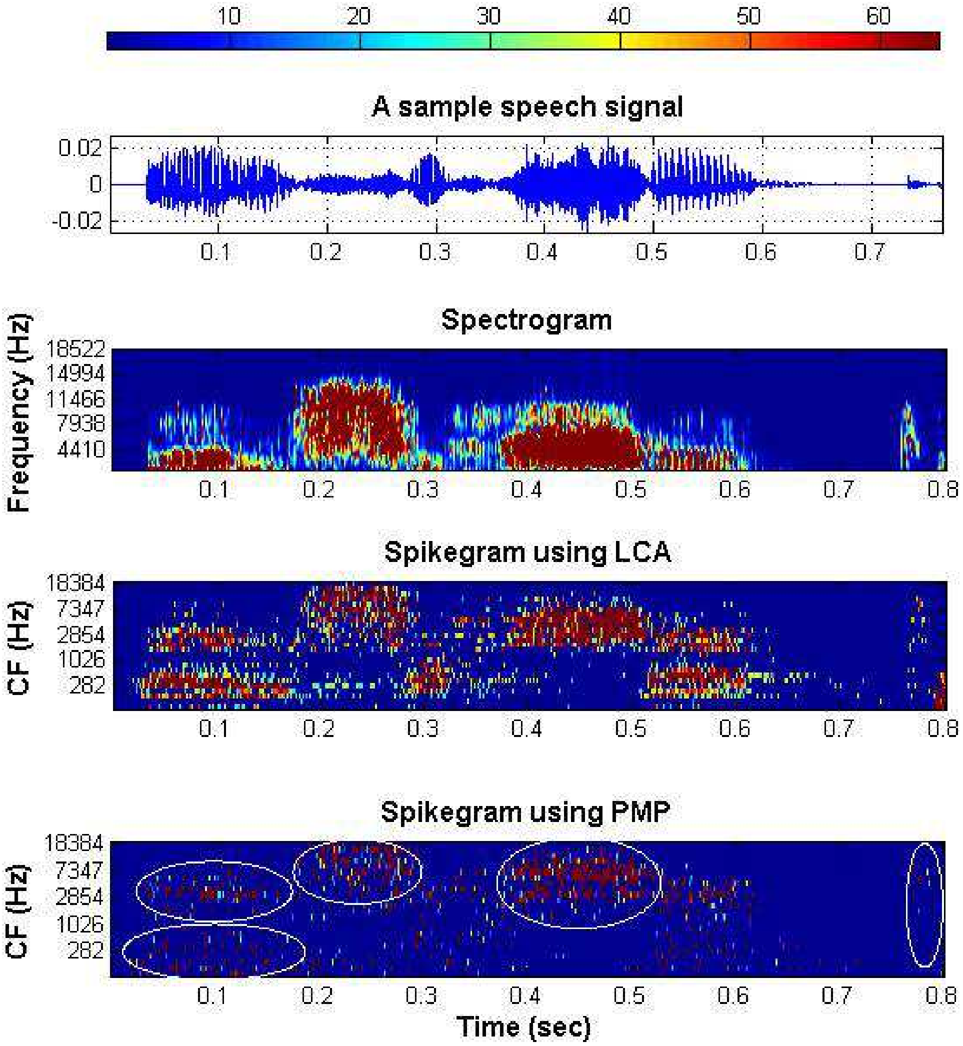

Figure 2.5 **Top:** the color bar for the spectrogram and spikegram. **Second from the top:** the original time domain signal for the speech utterance "une fourchette" sampled at 44.1 kHz. **Third from the top:** the spectrogram using Hamming window with 128 bits length with 25 frequency bins. **Fourth from the top:** the spikegram using LCA when running 100 iterations, **Bottom:** the spikegram when using PMP with non-zero coefficients as much as 20% of the signal's length. For both LCA and PMP, the dictionary includes 25 channels of gammatone kernel and the number of time shifts equals 10% of the signal's length. The variable CF denotes the center frequency of the gammatone channel. The high quality of LCA and PMP spikegram representations with the mentioned parameters are shown in [Pichevar *et al.*, 2010a], [Najaf-Zadeh *et al.*, 2008]

Moreover, the spikegram obtained by PMP is sparser and the absolute value of sparse coefficients are greater compared to the spikegram obtained by LCA. One reason is because of the efficiency of PMP in sparse representation and in removing the coefficients under the masking thresholds. The sparse representation property of PMP can be very essential for robust audio watermarking, where we want to insert watermark into the value or the phase of the most reliable and meaningful coefficients which contribute to the quality of audio signal and are not removed by watermarking attacks. In Fig.2.6, the histograms of coefficients obtained by PMP and LCA are plotted. As is seen, PMP generates more sparse high value coefficients than LCA. These high value coefficients are reliable for robust watermark insertion.

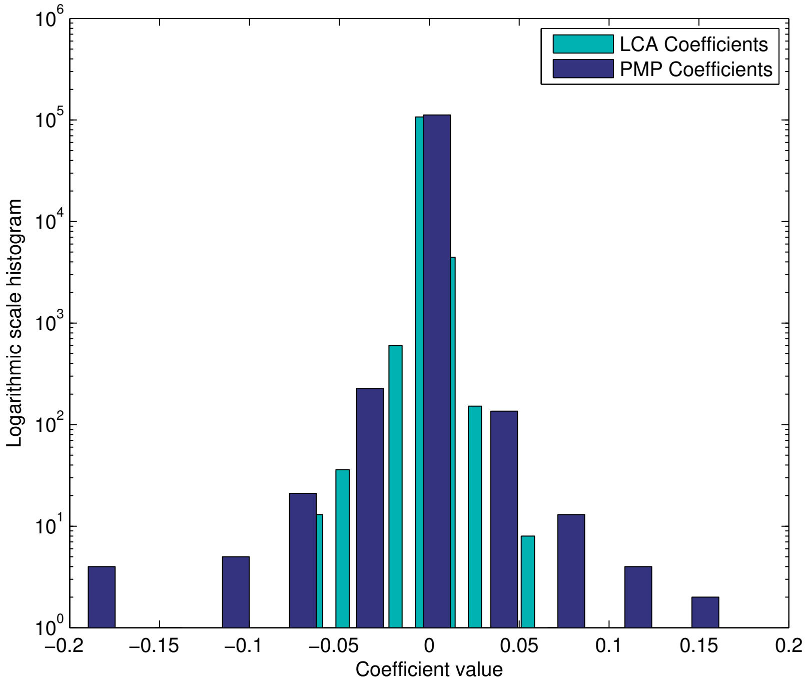

Figure 2.6 The histogram of spikegram coefficients obtained using LCA and PMP for the speech utterance “une fourchette” sampled at 44.1 kHz. As is seen, PMP bears more sparse and high values coefficients. These coefficients are suitable for watermark insertion. Note that to better compare the two histograms, the histogram of LCA coefficients is plotted with narrower bar widths.

2.8 Why exploring PMP spikegram for watermarking?

Briefly speaking, the reasons of choosing spikegram as a sparse representation for watermarking, include:

1. **Spikegram is a high resolution representation:** The spikegram represents the signal sparsely. Hence for preserving the energy equality between the representation and the original signal, the representation should include a few large coefficients. As mentioned previously, the sparse representation property of spikegram is similar to sparse activities of neurons in the auditory system. Also, the spikegram uses the gammatone which is a bio-inspired kernel which results in a representation that is bio-inspired and auditory based. As an example, in Fig. 2.5, the vowels are shown as horizontal colorful features and the transients as vertical colorful features. As is seen, a spikegram can specify the features of a signal with a great resolution, this is because of the high value sparse coefficients associated to these features.

2. **Spikegram is time-shift invariant and the whole signal is represented at once.** In spikegram representation, as time-shifting the signal causes the same amount of time-shifts on the underlying kernels, hence the representation is time shift invariant [Smith and Lewicki, 2005], [Smith and Lewicki, 2006]. This is because we use kernel based representation and at each time sample we have one kernel per channel. This property can be useful for audio watermarking to make the representation robust against de-synchronization attack which is one of the most powerful attacks against audio watermarking systems.

3. **A more robust watermarking can be achieved:** One traditional problem in audio watermarking is the insertion of watermark into non-meaningful spectro-temporal content of the audio signal. This means that many coefficients in the representation might be fragile against signal processing modifications and watermarking attacks where spectro-temporal content of the signal might be removed by masking thresholds, filtering or denoising. Using the spikegram obtained by PMP, firstly auditory based masking thresholds for all spectro-temporal coefficients in the representation are generated. Then all the spectro-temporal content of the signal under the masking thresholds are removed. Also, the remained coefficients are sparse (with high values) and are reliable for robust watermark insertion.

4. **Spikegram as a perceptual attack:** It is shown that the gammatone atoms are the building blocks of the natural sound [Smith and Lewicki, 2006]. Hence by sparse representation, an additive white Gaussian noise (AWGN) is not represented well and with high value coefficients on gammatone bases, as there are weak correlations between gammatone bases and white Gaussian noise. Thus spikegram with gammatone dictionary, for an audio signal contaminated with AWGN, generates few coefficients with large values associated to the signal and many low value coefficients associated to the noise. This means that the low value coefficients associated to the additive white Gaussian noise are spread on the whole spikegram [Razaviyayn *et al.*, 2014]. Hence this makes the spikegram suitable for thresholding denoising.

   In Fig.2.7, the time domain signal with the sampling frequency of 44.1 kHz, contaminated with a white Gaussian noise with SNR =15 dB has been plotted along with its spikegram, the denoised spikegram and the denoised time domain signal. Denoising is performed by hard thresholding the spikegram coefficients. As is seen, the spikegram representation itself has a denoising property and removes noise contents from the signal. This is because the spikegram uses PMP for the signal representation where many small value coefficients (associated to the noise) under the masking thresholds are removed. In this example, a coefficient, for which the abstract value is under the threshold of 0.002 is set to zero (This threshold is obtained empirically and by trial and error to have the highest SNR after denoising.). Note that, by denoising using the spikegram, many small value coefficients are removed [Donoho, 1995]. This is mainly because the gammatone kernels are matched more to the natural sound than noise [Smith and Lewicki, 2006], and the PMP removes many coefficients associated to the noise which are located under the perceptual masks [Pichevar *et al.*, 2010a]. Hence, to design a robust audio watermarking system, it is necessary to insert watermark bits into sparse high value coefficients. Therefore, this again shows the efficiency of sparse representation for robust audio watermarking. Furthermore, in Fig.2.7, the Gaussian noise is appears more on the small value coefficients and not too much on the high value coefficients. In many additive watermarking systems such as spread spectrum and quantization index modulation, the watermark is inserted as a small amplitude noise into several coefficients. This means that spikegram as an attack is able to remove this additional noise created by watermarking.

   In chapter 5, spikegram is used to design an audio watermarking attack called perceptual replacement attack.

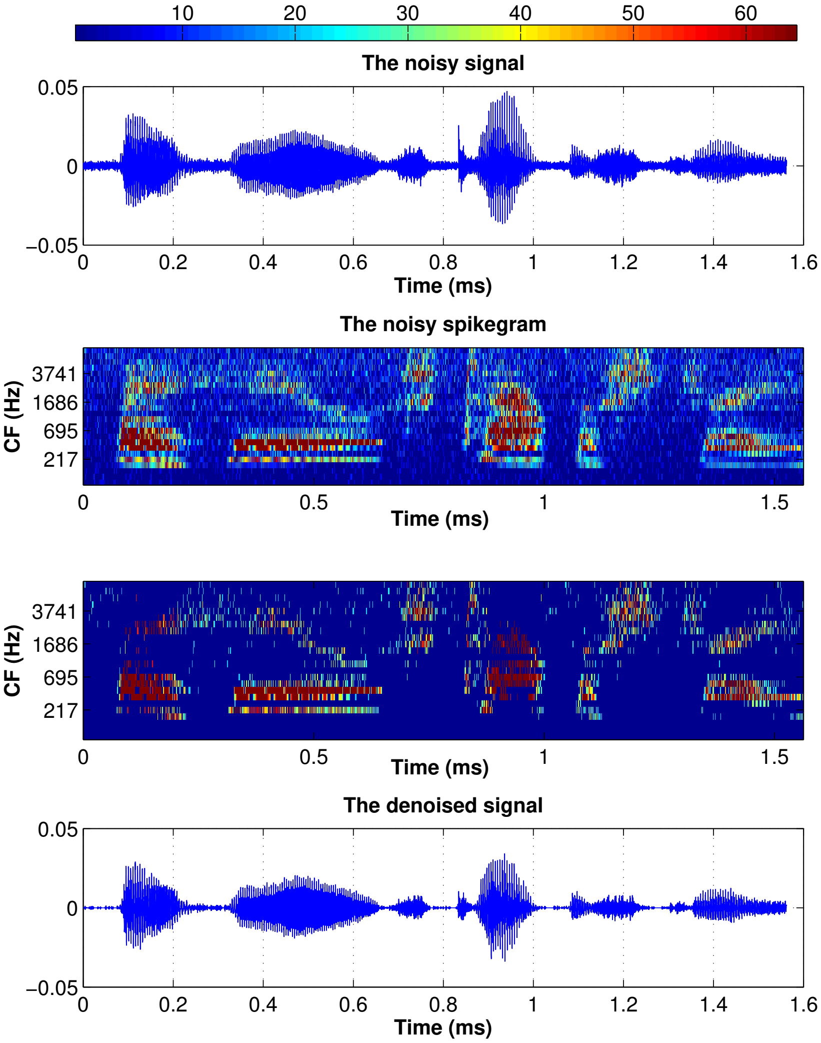

Figure 2.7 **Top:** the color bar for spikegram. **Second from the top:** the noisy signals for the speech excerpt “a huge tapestry” sampled at 44.1 kHz, contaminated with a white Gaussian noise with SNR =15 dB. **Third from the top:** spikegram of the noisy signal. **Fourth from the top:** denoised spikegram, by thresholding the coefficients below 0.002. **Bottom:** re-synthesized denoised signal using the spikegram thresholding. The signal representation is performed by running the perceptual matching pursuit with the number of iterations equal to 20 percent of the signal’s length using 25 channels of gammatone dictionary. The number of time shifts equals 1/10 of signal’s length. CF denotes the center frequency of the gammatone channel.

2.9 Summary

In this chapter, we explained the spikegram using the perceptual matching pursuit (PMP). The PMP generates the masking thresholds for all the gammatones in the representation. As the masking thresholds obtained by PMP can be used for audio watermarking, in the rest of the thesis, PMP is used for making spikegrams.

In the next chapter, we design a tamper localization method using a novel audio watermarking method in the spikegram domain.

CHAPTER 3

AUDIO AUTHENTICATION USING SPIKEGRAM

3.1 Avant-propos

Auteurs et affiliation:

— Yousof Erfani: étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

— Ramin Pichevar: professeur associé, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

— Jean Rouat: professeur titulaire, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

Date de soumission: 6 Mars 2016

Revue: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Titre français: Localisation des falsifications dans les signaux sonores en utilisant le tatouage de spectre étendu modifié basé sur masquage auditif dans le domaine parcimonieux

Résumé français:

L'une des principales applications de tatouage sonore est la vérification de l'authentification et la détection de l'altération des signaux sonores. La Localisation des plus petits changements dans les signaux sonores est essentielle pour l'authentification de ces signaux. Les méthodes actuelles d'authentification sonore ne sont pas capables de localiser les segments de courte taille des signaux sonores (plus petit que 250 msec) en présence d'attaques malveillantes telles que la suppression, le remplacement et la compression. Dans cet article, une méthode de localisation d'altération pour les signaux sonores est présentée dans le domaine parcimonieux perceptif en utilisant une version modifiée de spectre étalé (MSS). La méthode proposée a la capacité de localiser un court segment du signal sonore altéré par le remplacement, la suppression ou mise à l'échelle avec une taille plus petite que 250 *msec*. Pour atteindre cet objectif, le PMP (perceptual matching pursuit) est utilisé pour calculer une représentation invariante dans le temps, et parcimoneuse des signaux sonores ainsi que des seuils de masquage 2-D. Ensuite, le code d'authentification est inséré comme étant un tatouage caché à l'intérieur des coefficients parcimoneux. Pour localiser un segment falsifié du signal sonore hôte, les codes d'authentification de tous les segments intacts sont reconnus correctement tandis que lorsque le segment est falsifié, le code d'authentification ne peut pas être reconnu, indiquant une tentative d'attaque. Pour garantir la qualité du tatouage sonore, les données de tatouage sont façonnées par seuils de masquage. Nous montrons expérimentalement l'efficacité de notre approche en localisant les segments malicieusement attaqués des signaux sonores et nous testons aussi pour les signaux compressés en utilisant soit 32-64 kbps MP3 ou USAC (unified speech and audio coding).

3.2 Audio tampering localization using masking-aware MSS watermarking in the sparse domain

3.2.1 Abstract

One of the main applications of audio watermarking is the authentication verification and tampering detection from the audio signals. Localization of the smallest changes in audio and speech signals is essential for authentication of such signals. Current methods of audio authentication lack the ability to localize the short-size tampered segments in audio-speech signals (smaller than 250 ms) in the presence of malicious attacks such as removing, replacing and compression techniques. In this chapter, a blind audio tampering localization method is presented in the perceptual sparse domain using a modified spread spectrum (MSS) watermarking approach. The proposed method has the ability to localize a short segment of the audio signal tampered by replacing, removing or re-scaling with a size smaller than 250 ms. To achieve this goal, perceptual matching pursuit is used to compute a sparse and time-shift invariant representation of audio signals as well as 2-D masking thresholds. Then authentication code (which includes an Identity Number, ID) is inserted as a hidden watermark inside the sparse coefficients. To localize a tampered segment of the audio signal at the decoder, the IDs associated to intact segments are detected correctly, while the ID associated to a tampered segment is misdetected or not detected. To guarantee the high quality of watermarked audio, the watermark data is shaped by masking thresholds found by perceptual matching pursuit. We experimentally show the efficiency of our approach in localizing the tampered segments of the audio signals and in determining whether the signal has been compressed using either 32-64 kbps MP3 or USAC (unified speech and audio coding) compression.

3.3 Introduction

Audio signals might experience modifications (e.g., replacement, time shifting) intentionally by attackers or unintentionally by signal processing transforms. The high- quality audio-speech modifications can be performed using speech analysis-synthesis methods [Bordel *et al.*, 2016], voice conversion [Percybrooks and Moore, 2015] and audio morphing [Caetano and Rodet, 2013]. This high quality modifications might result in erroneous authentication of those signals [Hua *et al.*, 2016] (see Fig.3.1). Audio watermarking (AW) has been offered for audio authentication by the insertion of hidden, transparent and irremovable watermarks inside the signal [Cox *et al.*, 2007]. For audio authentication, current methods classify the whole signal as tampered or untampered [Gulbis *et al.*, 2008]. Some other approaches are able to localize the position of tampering based on fragile watermarking [Chen and Liu, 2007]. In fragile watermarking, the watermark itself is not robust against attacks. Thus by performing mild signal processing transforms on the whole signal such as time shifting, re-sampling, re-quantization, the whole watermark is removed and no tamper localization is performed afterwards. However, for authentication applications such as forensic or voice over IP, audio watermarking should be robust against mild modifications. Furthermore, it should be able to localize the position of tampered segments such as segments added, replaced, removed or compressed.

In this paper, a tamper localization method is presented based on a semi-fragile audio watermarking. The watermark is robust against mild signal processing modifications such as re-quantization, re-sampling and 20 dB additive white Gaussian noise while it is able to localize the short time malicious tamperings (in 250 ms frames) such as frame replacing and removing. The proposed method is able to determine if a segment has been compressed by 32-64 kbps MP3 and 24 kbps USAC [Neuendorf *et al.*, 2009] transforms.

In the proposed method, the kernel based sparse representation [Smith and Lewicki, 2005] (for self-synchronization and robust tamper localization) is combined with a modified spread spectrum watermarking method (MSS) and perceptual matching pursuit [Najaf-Zadeh *et al.*, 2008], [Pichevar *et al.*, 2010a]. A watermark stream is inserted into audio frames. This watermark stream includes a synchronization code and an ID (which is an incremental frame number), so that in the case of tampering attack in one segment of the signal, this watermark is misdetected at the decoder. To do so, we use perceptual matching pursuit (PMP) [Najaf-Zadeh *et al.*, 2008] to obtain a kernel based representation of the host signal and the watermark is inserted into the sparse coefficients. In this way, the re-synchronization of the tampered audio with the input untampered audio is performed in the spikegrarm domain.

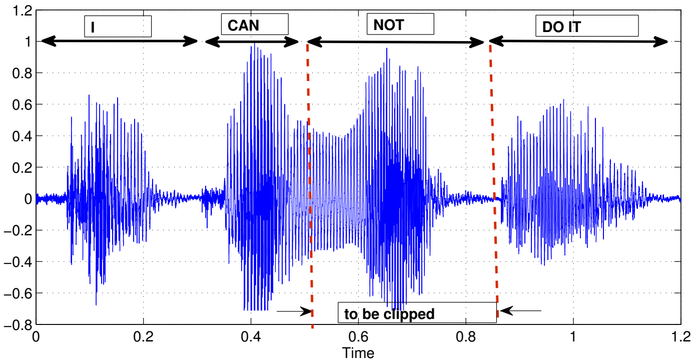

Figure 3.1 The concept of audio tampering. In this example, the utterance "NOT" is clipped from the speech signal by an attacker. Using the proposed tamper localization method, the location of the tampering can be identified.

For watermark insertion and extraction, the *MSS* method is proposed which guarantees a payload between 20-25 bps and also a high quality embedding (with average mean opinion score above 4.7). To design an efficient blind decoder we investigate the use of a cascade of projections computing block and a correlation decoder.

Finally, we show that our approach is able to discover the watermark associated to each frame and localize the de-synchronized tampered segments of the signal. Also, it is shown that the presented watermarking method preserves the transparency of the input signal and the watermark is not removed under common signal modifications such as re-quantization, down-sampling and 20 dB additive white Gaussian noise.

The paper is organized as follows. The kernel based sparse representation is introduced in section 3.4. The proposed perceptual sparse domain audio watermarking is represented in section 3.5. In section 3.6, it is explained how to control the distortion of the watermarking method. The multi-bit embedding using the concept of friend gammatones is described in section 3.7. Experimental results are presented in section 3.8. The section 3.9 is the conclusion.

3.4 Kernel based spikegram representation

In sparse representation, the signal $x[n], n = 1 : N$ (or $\boldsymbol{x}$ in vector format) is decomposed over a dictionary $\Phi = \{g_{c_j}[n - l_j]; \quad n = 1 : N, j = 1 : M\}$ where $g_{c_j}[n - l_j]$ indicates a gammatone at channel (center frequency) c_j shifted l_j samples along the time axis. The goal of sparse representation is to render a sparse vector $\boldsymbol{\alpha} = \{\alpha_j; \quad j = 1 : M\}$ which includes only a few non-zero coefficients with the smallest error of reconstruction for the

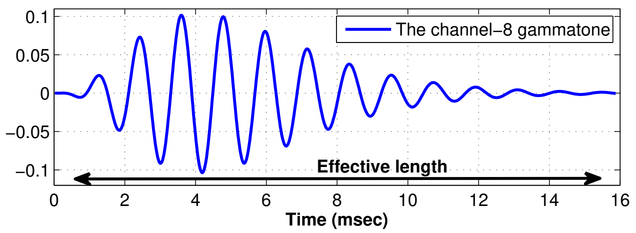

Figure 3.2 The channel-8 gammatone with the center frequency of 840 Hz and the effective length of 13.9 ms. Gammatones with odd channel numbers between 1-19 are selected for watermark insertion. The sampling frequency is 44.1 kHz.

host signal $\boldsymbol{x}$ [Smith and Lewicki, 2005], [Pichevar *et al.*, 2010a]. Hence,

$$x[n] = \sum_{j=1}^{M} \alpha_j g_{c_j}[n - l_j], \quad n = 1, 2, .., N \tag{3.1}$$

where α_j is a sparse coefficient. The dictionary Φ is represented by a 2D time-channel plane that comprises N_c channels of a gammatone filter bank along the channel axis repeated each q (time quantization) samples along the time axis (hence, $M = N_c \times N/q$). Thus $g_{c_j}[n - l_j]$ is one base of the dictionary which is located at a point corresponding to channel $c_j \in \{1, .., N_c\}$, and time sample $\text{ł}_j \in \{1, q, .., N\}$ inside the 2D time-channel plane. The spikegram is the 2D plot of the coefficients multiplied by their associated gammatones at different instances and channels (center frequencies). The number of non-zero coefficients per signal's length is defined as the sparsity of the representation.

Perceptual matching pursuit (PMP) is a recent approach which solves the problem in (3.1) for audio and speech. PMP uses a gammatone dictionary (equation (3.2)) in combination with masking [Najaf-Zadeh *et al.*, 2008]. PMP is a greedy method and is an improvement over matching pursuit [Mallat and Zhang, 1993] and generates masking thresholds for all gammatones in the dictionary. It selects only audible gammatones for which the sensation level is above an updated masking threshold and neglects the rest. The efficiency of PMP for signal representation is confirmed in [Najaf-Zadeh *et al.*, 2008]. The gammatone dictionary is bio-inspired and adapted to the natural sounds [Patterson *et al.*, 1988], [Smith and Lewicki, 2005] and is shown to be efficient for sparse representation [Pichevar *et al.*, 2010a]. A gammatone filter equation [Slaney, 1998b] has a gamma part and a tone part as below

$$g[n] = an^{m-1}e^{-2\pi ln}cos[2\pi(f_c/f_s)n + \theta] \tag{3.2}$$

where, n is the time index, m and l are used to tune the gamma part of the equation. Also, f_s is the sampling frequency, θ is the phase, f_c is the center frequency of the gammatone.

Table 3.1 Effective lengths, center frequencies and roll-off regions for gammatones used in this work.

Channel number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Effective length (ms)	55.2	39.1	31.0	25.8	22.0	18.7	16.1	14.0	12.1	10.7	9.3	8.1	7.2	6.3	5.4	4.7	4.1	3.5	2.9	2.4	2.0	1.7	1.4	1.0	0.8
Center frequency (Hz)	50	150	250	350	450	570	700	840	1k	1.2k	1.4k	1.6k	1.9k	2.2k	2.5k	2.9k	3.4k	4k	4.8k	5.8k	7k	8.5k	10.5k	13.5k	18.8k
Roll-off region (ms)	5.54	3.32	1.13	0.68	0.56	0.43	0.36	0.31	0.25	0.2	0.20	0.18	0.13	0.11	0.09	0.07	0.06	0.04	0.03	0.03	0.03	0.02	0.02	0.00	0.00

The term a is the normalization factor to set the energy of each gammatone to one.
In this paper, a 25-channel gammatone filter bank is used (each channel corresponds to one center frequency) and covers the frequency range of 20 Hz-20 kHz. Also, the effective length of a gammatone at channel c_j is symbolized with $leff_j$ and is defined as the time duration of the gammatone where the gammatone's envelope is greater than one percent of its maximum value. Moreover, the time duration between the peak of the auto-correlation function of a gammatone at channel c_j and the nearest zero to the peak is called the *roll-off* region R_j. In Table 3.1, center frequencies, effective lengths and roll-off lengths for the 25 gammatones, used in this work, are given. In Fig.3.2, the channel-8 gammatone is plotted for a sampling frequency of 44.1 kHz. The sparse kernel-based representation is shown to be time-shift-invariant and compact and high resolution [Smith and Lewicki, 2005]. These properties are attractive for watermarking [Wang Yong and Jiwu, 2010].

3.5 Perceptual sparse domain audio watermarking

3.5.1 Proposed modified SS-watermarking 1-bit embedder

As a preprocessing, PMP is run on the gammatone dictionary as described in [Najaf-Zadeh *et al.*, 2008] to find sparse coefficients α_j and their associated masking thresholds m_j at iterations $j \in \{1, 2..M\}$. The watermark embedder is shown in Fig.3.3. First, the watermark bit stream is generated as a cascade of synchronization code and frame ID. Then a pseudo noise (PN) sequence $p_j \epsilon \{-1, 1\}$ is generated using a key and the pseudo random noise generator (PRNG) [Klein, 2013b]. The auto-correlation function of PN is similar to Dirac delta function hence suitable for correlation detector [Cox *et al.*, 2007]. For watermark embedding, we propose a modified spread spectrum (MSS) in the sparse

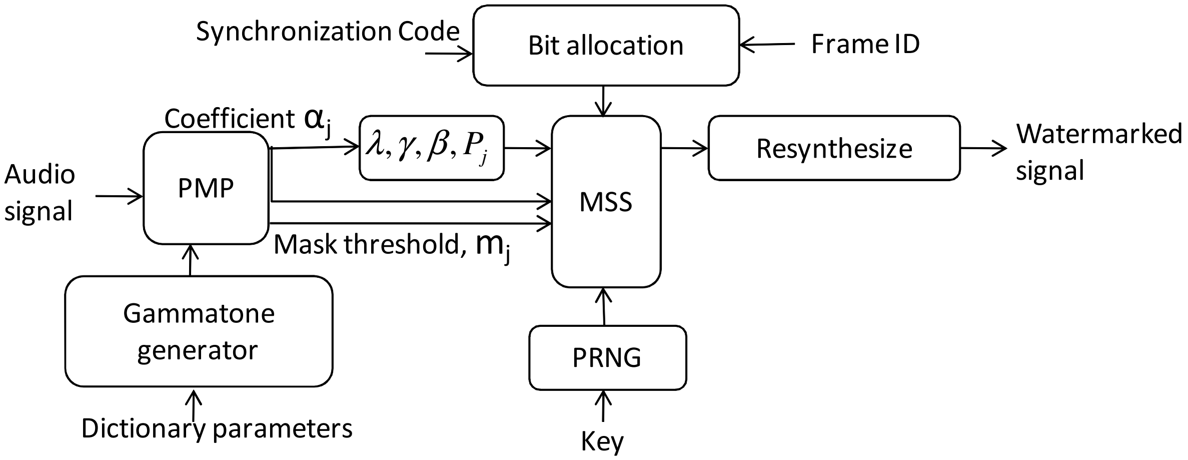

Figure 3.3 The proposed encoder. The audio signal is represented by spikegram from PMP coefficients. At each iteration, a sparse coefficient α_j and a masking threshold m_j is found. The watermark bit b is shaped via the masking threshold m_j and the robustness-quality trade-off factor β. The result is centred via γ and λ robustness tuning parameters and is added via MSS encoder to the sparse PMP coefficients. Finally, the watermarked signal is re-synthesized from the modified sparse coefficients. The PRNG [Klein, 2013b] is the pseudo noise generator which generates PN sequence p_j needed in MSS method.

domain. In MSS, each watermark bit $b\epsilon\{-1, 1\}$ is inserted into L coefficients α_j (randomly selected from M coefficients) to bear L watermarked coefficients w_j (for single bit watermarking $L = M$). Thus for one bit embedding, the embedding equations are as follows.

$$w_j = \alpha_j + (\beta b m_j - \gamma\lambda)p_j, \quad j = 1, .., L \tag{3.3a}$$

$$\lambda = \frac{1}{L}\sum_{j=1}^{L} P_j p_j \tag{3.3b}$$

$$P_j = \sum_{n=1}^{N} x[n] g_{c_j}[n - l_j] \tag{3.3c}$$

$$x_w[n] = \sum_{j=1}^{L} w_i g_{c_j}[n - l_j] \tag{3.3d}$$

where β is an additional quality control parameter to modify the masking threshold values m_j. The parameter P_j denotes the projection of the host signal $x[n]$ on the gammatone $g_{c_j}[n-l_j]$. The parameter λ is associated with the improved spread spectrum (ISS) algorithm [Malvar and Florencio, 2003] and here is called the projection annihilator parameter. Having the term λ at the encoder is essential to reduce the bit error rate of the correlation decoder [Malvar and Florencio, 2003]. The term γ is the proposed distortion-robustness control

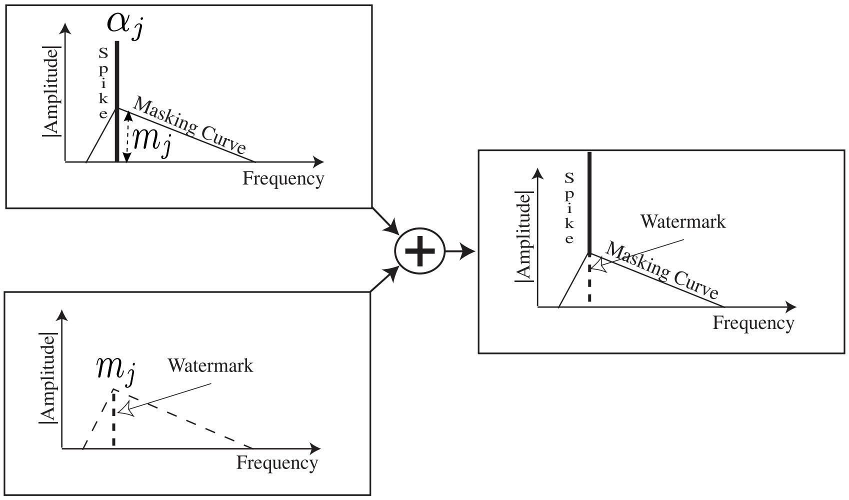

Figure 3.4 At iteration j of PMP, first a gammatone kernel is selected from the dictioanry of gammatone kernels and is called the original gammatone. Also, a sparse coefficient α_j and a masking threshold m_j associated to that original gammatone are computed (upper left plot). Then a gammatone with the same time-channel location as the original gammatone kernel with the amplitude of m_j is generated and called watermark gammatone (lower left plot). Finally, the watermark gammatone is embedded into the original gammatone using (right plot) (3.3). This procedure is continued at next iterations.

parameter. Note that the watermark embedder of spread spectrum (SS) [Malvar and Florencio, 2003] and improved spread spectrum (ISS) watermarking methods are governed by the same equation (3.3a), except that for SS and ISS watermarking, the innovative term γ is always equal to zero and one respectively [Malvar and Florencio, 2003]. However, in the proposed method, γ is set to zero when λ has the same sign as the inserted bit b, otherwise is set to one. The combination of (3.3a) with (3.3d), gives the watermarked signal $x_w[n]$. Thus the watermarked signal is $x_w[n] = x[n] + w[n]$ where

$$w[n] = \sum_{j=1}^{L} (\beta b m_j - \gamma\lambda) p_j g_{c_j}[n - l_j] \tag{3.4}$$

Hence, at each point of the time-channel plane associated to $x_w[n]$, two similar gammatones with different amplitudes are added together. The first one is the signal gammatone with amplitude α_j and the second one is the watermark gammatone with amplitude $(\beta b m_j - \gamma\lambda) p_j$ (Fig.3.4).

The masking term m_j attenuates the amplitude of watermark gammatone to be inaudible in the presence of the original signal gammatone (Fig.3.4).

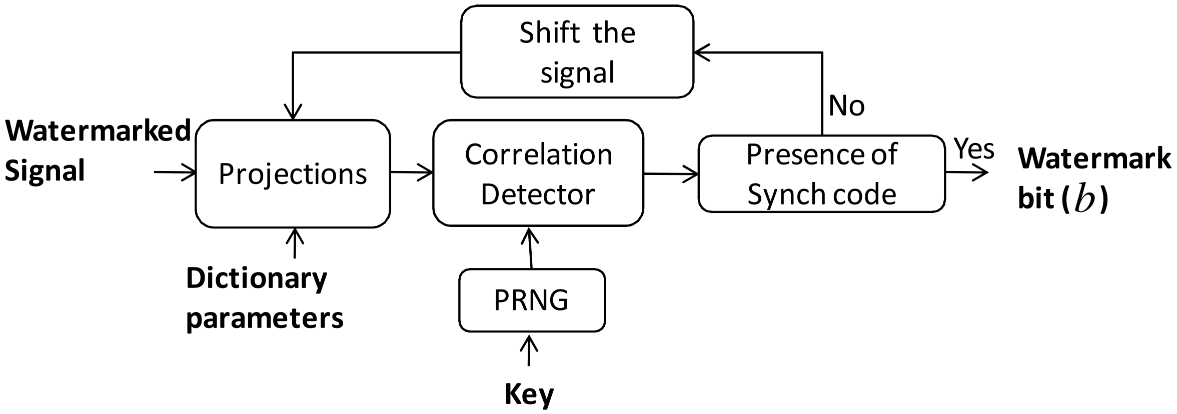

Figure 3.5 The proposed blind watermark decoder. The projection of the watermarked signal over each watermark gammatone is computed. Each watermark bit is found via correlation between the projections and the PN sequence associated to that watermark bit. If the input signal is shifted, then the decoding is done on the shifted versions of the signals until the acquisition of the synchronization code. PRNG: pseudo random noise generator.

3.5.2 Proposed 1-bit blind decoder

The decoder (Fig.3.5) receives the watermarked signal, shares the key, the synchronization code and dictionary parameters (g, N_c and q) with the encoder. Thus the positions of gammatones in the spikegram are identified using the dictionary parameters. The inserted watermark bit b is decoded from the watermarked signal's projections overs the watermark gammatones (with odd channel numbers between 1-19) in the spikegram. Firstly, we compute the projection of the watermarked signal on a watermark gammatone $g_{c_j}[n-l_j]$ as follows.

$$P_{wj} = \Big\langle x_w[n], g_{c_j}[n-l_j] \Big\rangle = \sum_{n=1}^{N} x_w[n] g_{c_j}[n-l_j] \tag{3.5}$$

Then, by combining (3.5) and (3.4), we have

$$\begin{aligned} P_{wj} &= P_j + \Big(\beta b m_j - \gamma\lambda\Big) p_j \\ &\quad + \sum_{i=1, i\neq j}^{L} p_i \Big(\beta b m_i - \gamma\lambda\Big) \Big\langle g_{c_i}[n-l_i], g_{c_j}[n-l_j] \Big\rangle \end{aligned} \tag{3.6}$$

The right term of the right side of equation (3.6), is the interference that the decoder receives from other watermark insertions. To get rid of this term, the watermark carrying gammatones, $g_{c_i}[n-l_i]$ and $g_{c_j}[n-l_j]$ should be uncorrelated. For this goal, firstly two watermark gammatones, should be at least one channel apart in the spectral domain and at least one gammatone effective length apart in the time domain. With these conditions complied, the correlation between two gammatones will be lower than 3 % (the maximum correlation between two gammatones equals one) and can be ignored. In this case, (3.6) becomes

$$P_{wj} = P_j + \Big(\beta b m_j - \gamma\lambda\Big)p_j \tag{3.7}$$

After finding the total L watermark projections using (3.7), the decoding is performed by computing the correlation between L projections associated to watermark bit b [Malvar and Florencio, 2003], and the binary pseudo noise p, as below

$$r = \frac{1}{L}\sum_{j=1}^{L} P_{wj}p_j \tag{3.8}$$

Combining (3.3a) with (3.6) and (3.8), we have

$$r = \frac{b}{L}\sum_{j=1}^{L} \beta m_j + \lambda(1-\gamma) \tag{3.9}$$

The parameter r is called the watermark strength factor (when $b = 1$) and should have a high value with the same sign as the watermark bit b. By applying maximum likelihood (ML) estimation, the watermark bit is decoded as $\acute{b} = sign(r)$ [Malvar and Florencio, 2003]. The right term in the right side of equation (3.9), is an interference term which we call signal contribution and is controlled by the proposed term γ at the encoder.

In this work, SS and ISS watermarking have the same decoding equations as in (3.9) [Malvar and Florencio, 2003], except that in SS, always the signal contribution is equal to λ and in ISS, it is zero. However in MSS, $\gamma = \frac{1}{2}\Big(1 - sign(b)sign(\lambda)\Big)$. Thus, the signal contribution changes between λ and zero based on the value of γ. The efficiency of the proposed method relies on the fact that in (3.9), the signal contribution weakens the process of ML estimation of b from r, only when it has not the same sign of b (see (3.9)). Otherwise, it improves the ML estimation. Thus, the role of the innovative term γ is to nullify the signal contribution (γ=1) when it has negative effect on the decoder, otherwise it keeps it (γ=0). In comparison to ISS method, by keeping the signal contribution ($\gamma = 0$), we do not add the noise term λ to the encoder in (3.3a) and this results in the improvement of the quality of MSS compared to ISS watermarking. Also, by supposing that in half times watermark bits and signal coefficients are positive or negative, then all the time with probability of 50%, γ is zero, hence improving the robustness of the decoder.

In Fig.3.6, experiments are done on several audio signals and the average watermark strength factor (r when $b = 1$) obtained from (3.9) is plotted versus different embedding channels for the two cases of ISS ($\gamma = 1$ all the time) and MSS (γ is selected adaptively)

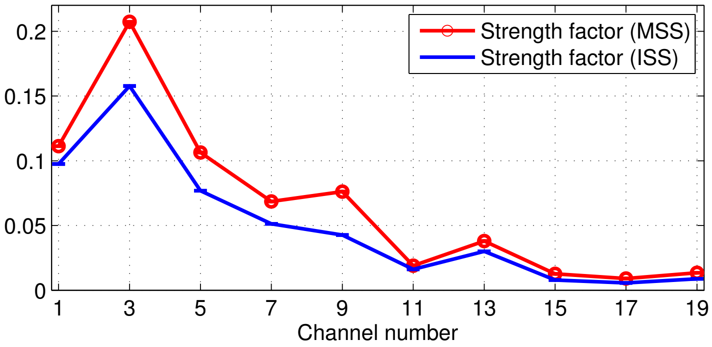

Figure 3.6 Comparison between the average robustness factor (r in (3.9) when $b = 1$) of ISS and MSS versus the embedding channel. As is seen, the robustness factor of MSS is always greater than the robustness factor of ISS for all cases. The experiments are performed on 100 audio signals including different music genres and English speech signals . Each signal is 3-minute long and is sampled at 44.1 kHz. All signals are normalized to have unit variance and zero mean.

where the parameter β is set to one and the embedding bit is $b = 1$. As is seen, all the time, the strength factor γ for the proposed MSS method is greater than the one for the ISS method. Moreover, greater improvements of strength factor occurs for lower channels (gammatones with lower center frequencies).

After all, for authentication of the found watermark bits, the decoder should verify the presence of the synchronization code for each frame of the watermarked signal. otherwise the decoder shifts the watermarked signal and performs the decoding and do this task several times until the acquisition of the synchronization code.

3.6 Distortion of the proposed method

In this section, we mention how to have high quality for the proposed method. This is done by computing the distortion of the watermarking method and controlling it to be masked under the masking thresholds of the input signal in the spikegram. Thus, we explain how to set the distortion controlling parameter β in the embedding equation (3.3a).

The distortion signal equals the difference between the watermarked and the original signal and is computed as

$$d[n] = x_w[n] - x[n] = \sum_{j=1}^{M} \Big(\beta b m_j - \gamma\lambda\Big) p_j g_{c_j}[n - l_j] \qquad (3.10)$$

The distortion signal $d[n]$ in (3.10) should be inaudible in the presence of the original signal $x[n]$. The variable $p_j \in \{-1, 1\}$ can be ignored since it only modifies the sign of the distortion. As in the proposed method the term γ takes only two values, either zero or one, thus we compute the distortion signal for these two cases. For the case of $\gamma = 0$ (meaning that λ and b have the same signs), (3.10) is changed to

$$d[n] = \sum_{j=1}^{M} \beta b m_j g_{c_j}[n - l_j] \tag{3.11}$$

Thus, to have an inaudible distortion, for each gammatone $g_{c_j}[n - l_j]$ in $d[n]$, its amplitude should be lower than its associated masking threshold m_j. Hence, if $\beta \leq 1$, all gammatones in the right side of equation (3.11) are masked under the gammatones associated to the signal representation (see Fig.3.4). Hence, in order to have inaudible distortion for the case of $\gamma = 0$, we freeze the term β to one.

Furthermore, for the case of $\gamma = 1$, as mentioned in section 3.5.1, the term λ and the watermark bit $b \in \{-1, 1\}$ have different signs. Hence, the term λ can be considered as $\lambda = -bC$ where C is a positive constant $C > 0$. Thus, in this case (3.10) can be written as

$$\begin{aligned} d[n] = \sum_{j=1}^{M} (\beta b m_j + bC) g_{c_j}[n - l_j] = \\ b \sum_{j=1}^{M} (\beta m_j + C) g_{c_j}[n - l_j] \end{aligned} \tag{3.12}$$

The watermark gammatones in the right side of equation (3.12) should be masked under the masking thresholds of the PMP representation. Hence, the term $(\beta m_j + C)$ should be at most equal to m_j. For having the greatest strength for the watermark gammatone, we set the amplitude of watermark gammatones to its associated masking threshold m_j. Thus for the case of $\gamma = 1$, the strength term β is obtained using (3.13) as below.

$$\beta = \frac{m_j - C}{m_j} \tag{3.13}$$

Hence, for the proposed method, the distortion is controlled by using the auditory based masking obtained from PMP (see Fig.3.4).

3.7 Multi-bit embedding and payload

3.7.1 Payload

Each watermark bit is embedded into L coefficients (L is called the repetition factor). The gammatones associated to these coefficients are called watermark gammatones. Thus, by having M_w watermark gammatones, the payload will be M_w/L. To achieve robustness against low-pass filtering, insertion is done into the low frequency content of the signal. Hence, watermark is not embedded into the 5 channels with the greatest center frequencies. Furthermore, to reduce correlation between watermark gammatones, every odd channel between 1-19 is selected. Therefore, a larger number of watermark gammatones means a greater payload.

3.7.2 Multi-bit watermarking using friend gammatones

In this section, we mention how to increase the number of watermark gammatones in order to have a greater payload using the principle of *friend* gammatones. So far, the watermark gammatones had the strict constraint of being uncorrelated. In this section, we show that even if the watermark gammatones are positively correlated, they can improve the efficiency of the watermarking decoder. In this chapter, watermark gammatones with positive correlations and the same channel number and located in the same time frame are called friend gammatones (Fig.3.7)). Also, for a given channel in the spikegram, and starting from sample one, a gammatone located at time samples equal to the multiples of effective length plus roll-off length is called a principle gammatone (Fig.3.7). Also, gammatones in the spikegram which have the same channel number as a principle gammatone and have positive cross correlations with it are *friend gammatones, or friend group* (see Fig.3.7).

How to find friends of a given gammatone?

For a given principle gammatone $g_{c_j}[n-l_j]$ in Fig.3.7, the auto-correlation of the gammatone, its peak and its zeros are computed. Gammatones in the roll-off region of the principle gammatone $g_{c_j}[n-l_j]$, i.e between l_j and $l_j + R_j \times F_s$ (F_s is the sampling frequency) are considered as its friend gammatones since their correlations with $g_{c_j}[n-l_j]$ are positive and have the same channel number (see Fig.3.8). Note that, with this definition, the friend

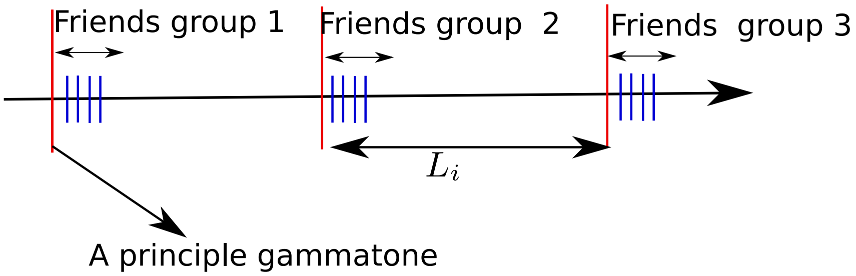

Figure 3.7 The position of principle gammatones and their friend groups for channel i in the spikegram of Fig.3.9. The friend gammatones of different principle gammatones are uncorrelated. L_i equals the effective length plus the roll-off length for channel i gammatone.

gammatones of a principle gammatone are also friends of each others.
For the principle gammatone $g_{c_j}[n-l_j]$, and the sampling frequency of F_s, there are $R_j \times F_s$ friend gammatones located between the two time samples l_j and $l_j + R_j \times F_s$. For a given channel, two principle gammatones with all their associated friends should be apart with as many as $(leff_j + R_j) \times F_s$ samples. In this case, friend kernels of two different principle gammatones have zero correlations.

Here, we show that for a given channel, insertion of one watermark bit b with the same pseudo noise value into friend gammatones increases the payload of the method. Hence, for the sake of simplification, we suppose that the same watermark bits have been inserted into the friend gammatones of a given gammatone. Thus for the detection of the watermark bit b, the rightmost side of (3.6), should not be obtrusive. By considering $c_{ij} = \Big\langle g_{c_i}[n - l_i], g_{c_j}[n - l_j]\Big\rangle$ and $c_{jj} = 1$, then (3.6) becomes

$$P_{wj} = P_j + \sum_{i=1}^{L} p_i\Big(\beta b m_i - \gamma\lambda\Big)c_{ij} \tag{3.14}$$

Thus, the decoding equation (3.9) becomes

$$r = \frac{b}{L}\sum_{j=1}^{L} \beta m_j c_{ij} p_i p_j + \lambda(1 - \gamma \sum_{i=1}^{L} c_{ij} p_i p_j) \tag{3.15}$$

As the same watermark bits are inserted into friend kernels, they can be assigned with the same PN values. Hence in (3.15), $p_i = p_j$. Thus we have

$$r = \frac{b}{L}\sum_{j=1}^{L} \beta m_j c_{ij} + \lambda(1 - \gamma \sum_{i=1}^{L} c_{ij}) \tag{3.16}$$

Note that, friend gammatones have positive correlation, i.e $c_{ij} > 0$, thus $\sum_{i=1}^{L} c_{ij}$ is greater than one. In the proposed method, the term γ equals to either zero or one. Here we show that for both cases of $\gamma = 0$ and $\gamma = 1$, the strength factor r in (3.16) is greater compared to when friend gammatones are not used in (3.9). First, when $\gamma = 0$, the left term of the right side of (3.16) is greater compared to (3.9) since $(\sum_{i=1}^{L} c_{ij} > 1)$ while the right terms in the right sides of both equations are the same. Thus in this case, the strength factor in (3.16) is higher compared to one in (3.9). Also, when $\gamma = 1$ this means that b and λ do not have the same signs. Then as $(1 - \gamma \sum_{i=1}^{L} c_{ij}) < 0$, therefore the two terms in the right side of (3.16) will have the same signs. In this case, the watermark strength is increased compared to the one obtained from (3.9) in which all the time, the rightmost side of equation (3.9) is zero.

3.7.3 Multibit embedding using friend gammatones

Hence the procedure for multi-bit watermarking is as follows (see Fig.3.9).

1. Signal representation: Represent the signal with PMP using 25-channel gammatones located at each time sample ($q = 1$), generate the spikegram, find the masking thresholds for any gammatone in the spikegram.
2. Finding friend gammatones: For odd channel numbers j between 1-19, the gammatones located at the time samples equal to the multiples of the sum of gammatone's effective length and roll-off region length ($leff_j + R_j$) are considered as principle gammatones. We find all associated friend gammatones of these principle gammatones. Consider the same pseudo random value p_j for each set of friend gammatones (see Fig. 3.9). All principle gammatones and their friends form a dictionary of watermark gammatones.
3. Watermark insertion: Insert one watermark bit into the gammatones located at each time frame (all of them are friends). This results in the watermark insertion into $R_j \times F_s$ gammatones per one time frame of the channel j.
4. Signal reconstruction: Reconstruct the watermarked signal from the modified coefficients.

3.7.4 The new improved payload using friend gammatones

For the mentioned multi-bit watermarking, the same decoding equation (3.16) is used. See Table 3.1 for the center frequency, roll-off region and effective length of gammatones

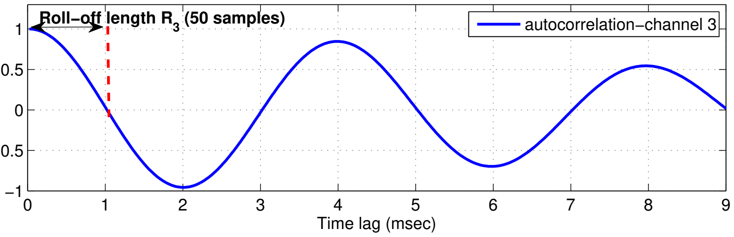

Figure 3.8 An example of roll-off region for channel-3 gammatone. For the channel number 3, 50 gammatones located at the beginning of each time frame in Fig.3.9 are selected as friend kernels for the watermark insertion.

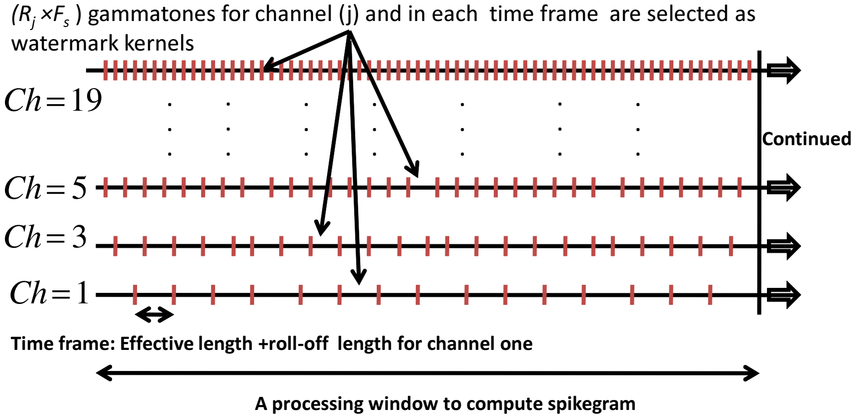
$(R_j \times F_s)$ gammatones for channel (j) and in each time frame are selected as watermark kernels
$Ch = 19$
$Ch = 5$
$Ch = 3$
$Ch = 1$
Continued
Time frame: Effective length +roll-off length for channel one
A processing window to compute spikegram

Figure 3.9 Multi-bit MSS watermarking. Each watermark bit is inserted into friend kernels located at an odd channel number between 1-19 and also in a time frame as long as the watermark effective length plus the roll-of region length. At the starting point of each time frame (between two red bars), $R_j \times F_s$ friend gammatones (which have positive correlation with one anohter) are chosen for watermark insertions (where F_s is the sampling rate and R_j is roll-off for a given channel in second). The watermark insertions are conducted using (3.3). Gammatones with greater channel numbers (center frequencies) have shorter effective lengths and roll-off region. This results in less watermark insertions into high channel numbers.

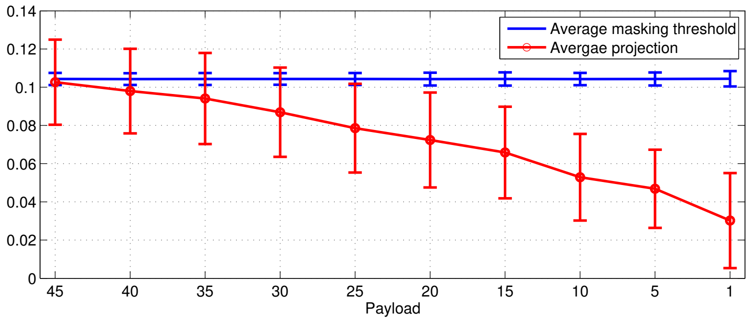

Figure 3.10 The average masking threshold (blue) and the average projection annihilator factor λ (red) for the proposed watermarking system versus payload (M_w/L). The average results are reported for 100 signals including audio and speech signals, 3 minutes each, sampled at 44.1 kHz. As is seen, the greatest achievable payload is between 20-25 where the average masking threshold is all the time greater than the average projection annihilator factor. The vertical bars indicate the 95% confidence intervals.

used in this work. By using the idea of friend gammatones, the number of watermark gammatones will be $M_w = [R_1/L_1 + R_3/L_3 + .. + R_{19}/L_{19}] \times F_s$ in which L_i indicated the effective length plus the roll-off region for channel i. Based on Table 3.1, the watermark insertion is done on $M_w = 12950$ gammatoness.

3.7.5 The maximum achievable payload of the proposed method for the average audio-speech signals

Based on (3.9) and (3.13), it is necessary to have a positive β. Otherwise, the sign of the strength factor might change and the watermark bit might be misdetected. To have a positive β, based on (3.13), for each watermark gammatone $g_{c_j}[n - l_j]$, the masking threshold m_j should be greater than the term λ.

In Fig.3.10, the average masking term m_j and the term λ are plotted for 100 audio signals including audio and English speech signals, 3 minutes each (5 hours in total). As is seen, with a payload between 20 bps and 25 bps, the projections remain lower than the masking threshold for more than 95% of the times. Thus the acceptable payload for the proposed method is between 20 bps-25 bps (which is associated to a repetition factor between 518 and 647).

3.8 Experimental results

3.8.1 Experimental setup

For the quality test, 6 types of audio signals have been selected to generate a dataset for our implementations including *speech (male female in French), Castanet, Percussion, Harmonics and song.* Objective and subjective difference grade tests are done on the pieces of 4.5-7.5 seconds of the mentioned signals. For the robustness test, 100 audio signals (of different music genres and English speech, in total 5 hours), 3 minutes each are used. All signals are sampled at 44.1 kHz and have 16 bits wave format. The *PMP* representation [Najaf-Zadeh *et al.*, 2008] is used with a gammatone filter bank to represent the audio signals in the dataset by coefficients with a maximum density of 0.5 (density equals the number of non-zero coefficients divided by the number of signal samples). A 25 channel gammatone filter bank is repeated at each sample along the time axis to build the 2-D dictionary matrix Φ. Note that, in all our experiments, the PMP automatically stops with a density between 15% and 30%. For robustness against high frequency degradation attacks, the watermark is not inserted into coefficients associated to the greatest 5 frequency channels. A 13-bit Barker sequence [Borwein and Mossinghoff, 2008] including -1 and 1 is the synchronization code. The watermark signature includes a 13-bit synchronization code and a 10-bits ID (the frame number) and is embedded in every second of the signal. This means an insertion of a total of 23 bits per second.

For the subjective difference grade (SDG) listening test, we followed the protocol mentioned in [ITU, 1996]. Fifteen subjects (varying from experts to people with no experience in audio processing, including male and female speakers, aged between 20-50 including, a mix of English speaking, French speaking and Persian speaking) participated in an 5-scale (minimum 1, maximum 5) MOS test by listening to signals via Bayerdynamic DT250 headsets. For the objective difference grade (ODG), an open source method was used [Kabal, 2002]. ODG is also a 5-scale test with minimum -5 for lowest quality and maximum 0 for the highest quality. ODG experiments are done on the excerpts of 5 seconds (in total 2 minutes of the audio signals, in Fig.3.11).

The SDG and ODG quality test results are shown in Fig.3.11. Overall, the average SDG and ODG results are equal to 4.7 and -0.3 respectively. A sample original signal and the difference between the original and the watermarked is shown in Fig. 3.12. The audio files including original and watermarked can be found at the following website: “http://www.gel.usherbrooke.ca/necotis/necotis-old/erfani.html”

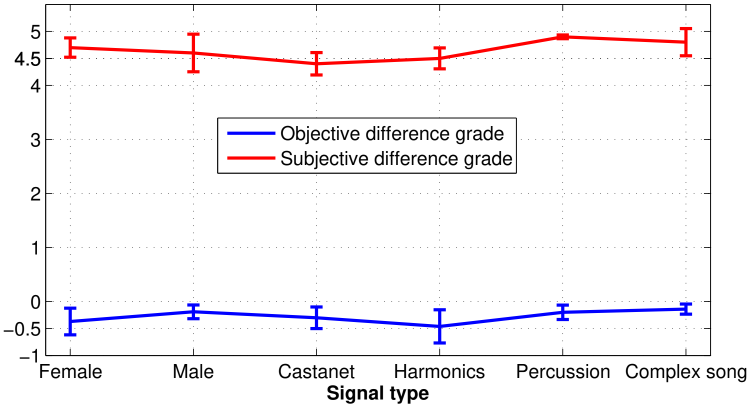

Figure 3.11 The average objective difference grade (blue) and subjective difference grade (red) results for the proposed method. The bar lines indicates 95% confidence interval. An average ODG higher than −0.5 or a *SDG* lower than 4.5 indicates that the quality change is imperceptible.

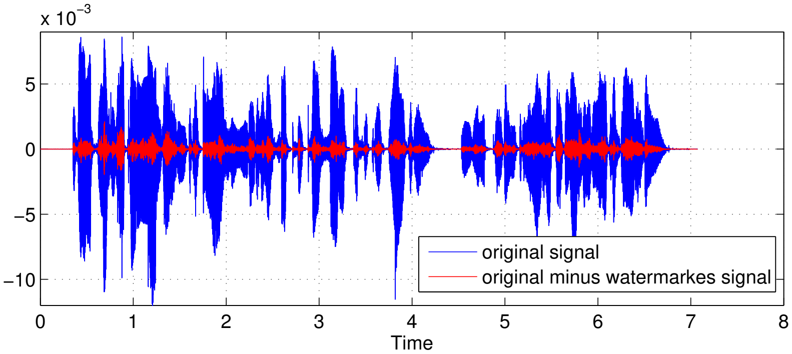

Figure 3.12 A sample original signal (blue), and the difference between the original and the watermarked one using the proposed method (red). The difference signal includes watermark gammatones which are masked under the masking thresholds of the original signal.

3.8.2 Bit allocation and Resynchronization

The watermark signature includes the synchronization code and the 1-second frame ID (in total 23 bits per second). This means that each watermark bit is inserted into $L = 12950/23 = 563$ watermark gammatones. The watermark gammatones are selected as in Fig.3.9. To have similar strength for all watermark bits, each bit is inserted into 563 randomly chosen gammatones from all watermark gammatones (in one second frame) using the Fisher-Yates shuffle permutation generator [Knuth, 1998]. The key that should be shared between the encoder and the decoder includes the initial states of the linear feedback shift register and the initial state for generating the permutation generator.

At the decoder, projections of thewatermarked signal on all gammatones in the 2D plane (dictionary) with odd channel numbers between 1-19, are computed. The watermarked gammatones are pinpointed as in Fig.3.9. The permutation and PN sequence generators are generated. All synchronization Barker codes (one for each second) should be detected (with less than 10% error rate). If one synchronization code is not detected, the tampering is alarmed for the 1-second frame associated to that synchronization code.

3.8.3 Experiments on attacks against proposed tampering localization method

The proposed tamper localization method is robust against ordinary transforms, and have erroneous detection of watermark bits under severe attacks. We insert the watermark signature (which includes a sequential ID number) into 1-second frames and evaluate the robustness of the method for each frame under the state of the art transforms on audio watermarking methods [Nikolaidis and Pitas, 2004] and strong tampering [Hua *et al.*, 2016].

— No attack: the same intact signal as at the encoder is received at the decoder.

— Downsampling: the watermarked signals are down-sampled from 44.1 kHz to 22.05 kHz.

— Gaussian noise attack: the watermarked signal is contaminated with additive white Gaussian noise so that the signal to noise ratio reaches 20 dB.

— MP3 attack: the raw wave files of the watermarked signals are compressed using 32 kbps and 64 kbps MP3 compression and then returned back to the raw wave format.

— 24 kbps Unified speech and audio coding (USAC) [Neuendorf *et al.*, 2009] which is a novel standard for low bit-rate speech and audio coding. USAC include two modes, the linear predictive (LPD) mode for speech signals and Fourier domain (FD) mode for audio signals.

- Low-pass filtering attack (LPF): watermarked signals are low pass filtered using the Butterworth filter with the cut off frequency of 11025 Hz.
- Re-quantization attack: the number of quantization bits is reduced from 16 bits to 8 bits and then again is returned to 16 bits.
- Cropping: segments of 500 samples of the watermarked signals are removed from the watermarked signal at five positions and subsequently replaced by segments of the watermarked signal contaminated with white Gaussian noise with a signal to noise ratio of 15 dB.
- Removing frames: 250 ms at the beginning of the frame number 4 (a quarter of the frame) is removed.
- Replacement attack: we replace the first quarter of frame number 4 (250 ms) of the watermarked signal with the first quarter of the associated frame in the original signal.
- Adding silent: 250 ms of the beginning of the frame number 4 (a quarter of the frame) is silenced (is set to zero).
- Time scaling: the watermarked signals are time scaled with the scale ratio of of 90 percent.
- Time shifting: the whole watermarked signal is time shifted as many as 100 samples.
- Large cropping: 10000 samples at 5 positions of the watermarked signal are cropped.
- Large Gaussian noise: the watermarked signals is contaminated with additive white Gaussian noise so that the signal to noise ratio reaches 10 dB.

The bit error rate (BER) is used as the mean of robustness-fragility evaluation. The BER is defined as the number of erroneously detected bits per all embedded bits as follows.

$$BER = \frac{\Sigma_{i=1}^{N/L}(b_i \oplus \acute{b_i})}{\Sigma_{i=1}^{N/L}(b_i b_i)} \tag{3.17}$$

where b_i and $\acute{b_i}$ are the i^{th} transmitted bit and its associated received bit respectively. When the sent bit at the encoder and its associated detected bit at the decoder have not the same signs, then the detected bit is called an erroneously detected bit. When the BER of watermark detection of the 13-bit synchronization code for one frame is more than 25%, it is classified as unauthenticated (tampered). As is seen from Table 3.2, the algorithm is robust against common signal modifications that can happen during the mobile and Internet transmissions. Furthermore, it is able to localize the place of maliciously-attacked frames. To evaluate the efficiency of the decoder, we made all attacks on the frame number

Table 3.2 Localization of tampered frames inside 100 audio signals including music and English speech 3 minutes each (5 hours). PMP is run to find sparse coefficients where the number of non-zero coefficients equals 50% of the signal's length. At iteration j of PMP, either $\beta = 1$ when $\gamma = 0$ or $\beta = \frac{m_j - C}{m_j}$ when $\gamma = 1$. All attacks are performed on frame number 4. In the case of (removing, replacing, adding) attacks, the process is done on a quarter of frame (250 ms). The results are shown for common signal modifications and severe attacks, NA: Not Applicable, NF: Not Found

Attack Name	**Condition**	**BER (%)**	**Frame No.**
No Attack	NA	1.5	NF
Down-sampling	to 22.05 kHz	3.6	NF
Gaussian noise	additive, 20 dB	5.1	NF
MP3	64 kbps	13.5	4
MP3	32 kbps	35	4
USAC	LPD,FD	37	4
LPF	Butterworh cutoff 11025 Hz	7.5	NF
Re-quantization	16 bits to 8 bits	2.3	NF
Cropping	500 samples 5 positions	6.7	NF
Removing a frame	250 ms long	51	4
Replace a frame	250 ms long	45	4
Adding silent frame	250 ms long	47	4
Time scaling	.90 scaling	45	4, 5
Time shifting	100 bits shifts	49	4
Big cropping	10000 samples 5 positions	43	4
Gaussian Noise	additive, 10 dB	39	4

4 (4^{th} second).

Table 3.2 shows the average results of our method to localize the tampered segments of audio signals for different attacks and conditions. Based on the results of Table 3.2, the average BER is very high where audio data is counterfeited via replacement, insertion, or deletion. Note that the proposed method classifies 24 kbps USAC, 64 kbps MP3 and 32 kbps MP3 attacked frames as tampered frames. This means that by using this method, we can determine whether the audio signal has been compressed. For the case of time scaling attack, the adjacent frame (here frame number 5) is wrongly considered as unauthenticated. This is because time scaling of one frame also destroys the authentication word of the neighbour frame. From Table 3.2 our method has robustness against signal modifications such as 64 kbps MP3, time shifting. Also, our method is able to extract the watermark "SYNCH" code from each 1-second frame of the watermarked audio (and then labeled the frame as untampered) for mild transforms such as re-sampling and re-quantization (see Table 3.2).

3.8.4 Comparison with state of the art works in audio authentication

Table 3.3 shows a comparison between our method and the state of the art methods in audio authentication. Here, the efficiency of the proposed method is only evaluated in terms of localizing the most malicious tampering on audio signals. Hence, the comparison is performed according to the ability of methods to localize the malicious changes in signal including replacing, adding, removing, time shifting (and scaling) frames of watermarked signal (corresponding to columns 1-4 of Table 3.3 respectively). Column 5 shows the number of frames mistakenly unauthenticated when one frame is attacked and column number 6 indicates the ability of the decoder to re-synchronize (the ability to find the synchronization code for re-synchronizing other intact frames). The average results of our experiments over 6 audio signals are compared to the reported results of four different states of the art methods for tamper localization.

In the first method [Unoki and Miyauchi, 2012] in Table 3.3, the audio tampering detection is based on a fragile watermarking where the watermark insertion is based on cochlear delay. Two different IIR filters which models two different cochlear delays are designed for zero-bit and one-bit embedding. At the decoder, the chirp-Z transform is used to detect the group delays corresponding to one and zero embedding.

In the second method [Steinebach and Dittmann, 2003], the basic features of audio signal for each processing frame are extracted including zero crossing rate, RMS and spectral features in the Fourier domain. The mentioned features are combined and hashed to create a signature of the original signal and this signature is embedded to the signal.

In the third method [Yuan and Huss, 2004], a fragile watermarking system is proposed based on the GSM encoder for the real time speech authentication. The audio watermarking is of zero-bit type and is based on GSM vocoder.

In the fourth method [Hua *et al.*, 2016], the audio authentication is done by exploring the absolute error-map of ENF (electronic network frequency) signals. They introduce the absolute-error-map (AEM) between the ENF signals obtained from the testing audio recording and the database. The AEM serves as an ensemble of the raw data associated with the ENF matching process.

In the fifth method, [Chen and Liu, 2007], the watermark is embedded in each time frame

Table 3.3 Comparison between the average results of the proposed method and those of recent methods in audio authentication. Ability of methods to locate: 1. the replaced, 2. the added, 3. the removed parts of the signal. 4. the length of the shortest tampered segment localized (in second). 5. the maximum number of falsely unauthenticated frames, 6. ability to determine the 32-64 kbps MP3 or 24 kbps USAC [Neuendorf *et al.*, 2009] compressed segments (NM: Not Mentioned).

Method	**1**	**2**	**3**	**4**	**5**	**6**
Unoki [Unoki and Miyauchi, 2012]	Yes	No	No	>1	NM	No
Stein [Steinebach and Dittmann, 2003]	No	No	No	many	Many	No
Yuan [Yuan and Huss, 2004]	Yes	Yes	Yes	> 1	1	No
Hua [Hua *et al.*, 2016]	Yes	Yes	Yes	> 1	Many	No
Chen [Chen and Liu, 2007]	Yes	Yes	Yes	> 1	2	No
Our Method	Yes	Yes	Yes	0.25	1	Yes

of the speech signal based on the line spectrum frequency (LSF) feature in the current frame, the pitch extracted from the succeeding frame, the watermark embedded in the preceding frame and the group index which is determined by the location of the current frame.

Due to the use of a kernel based representation and the insertion of the synchronization code and ID, our method has the ability to find the time-shifted and time-scaled frames of the audio signal. Also, our method is able to detect tampering for each 250 ms tampered frame which is the smallest length compared to other methods. In this case, when one frame is maliciously tampered, at most one additional frame is mis-classified as unauthorized. Thus the proposed method has the minimum number of mis-classified frames (only one frame) when only one frame is tampered compared to other methods. Moreover, through the insertion of an incremental ID number with the authentication mark, the ID of the tampered frame can be found.

3.9 Conclusion

In this paper, a blind, perceptual sparse-domain audio authentication method was presented using a proposed modified spread spectrum (MSS) watermarking. Using a perceptual kernel based representation method (PMP), our method inserts a watermark stream inside audio frames. We showed that the watermark is irremovable under signal processing modifications such as low-pass filtering. We also showed that compared to state of the art, not only our method does efficiently localize the shortest maliciously attacked segments of the signal (e.g., removed, replaced, and added, time-shifted segments), but it also has the ability to determine if the signal has been compressed by 32-64 kbps MP3 or 24 kbps USAC transforms. Our listening test shows the high quality of the watermarked signals. Our results confirm the suitability of the proposed method for authentication applications such as audio forensic.

Acknowledgment

This work was funded by CRC, NSERC and the Université de Sherbrooke. Many thanks to Hossein Najaf-Zadeh for the PMP code, to Malcolm Slaney for the auditory modelling toolbox, to Frédérick Lavoie and Roch Lefebvre for managing the listening test, to Philippe Gournay for providing the USAC code and to the participants of the listening test.

CHAPTER 4

COPYRIGHT PROTECTION IN SPARSE DOMAIN

4.1 Avant-propos

Auteurs et affiliation:

— Yousof Erfani: étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

— Ramin Pichevar: professeur associé, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

— Jean Rouat: professeur titulaire, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

Date de soumission: 6 Mars 2016
Revue: IEEE/ACM Transactions on Forensics.

Titre français: Tatouage sonores en utilisant le spikegramme et une approche à deux dictionnaires.

Résumé français:

Cet article présente une nouvelle technique de tatouage sonore en utilisant une représentation basé sur les noyaux gammatones. Une représentation parcimonieuse perceptive (spike gramme) est combiné avec un dictionnaire de gammatones pour construire une représentation robuste des sons. Par rapport aux méthodes traditionnelles du tatouage de phase où la phase des coefficients de Fourier sont modifiés, dans le présent document, les bits de tatouage est inséré en modifiant la phase des noyaux gammatones. En outre, les bits de tatouage sont automatiquement intégrés uniquement dans les noyaux avec des amplitudes élevées où toutes les coefficients non significatives ont déjà été retirés. Deux méthodes de tatouage sont proposées, l'une est basée sur l'insertions dans le signe de gammatones (une dictionnaire méthode). Un autre est basé sur l'insertions dans le signe et la phase de noyaux de gammatones (deux dictionnaire méthode). La robustesse de la méthode proposée est illustrée contre 32 kbps MP3 avec un taux d'insertion de 56,5 bps alors que l'état de l'art de charge utile pour le tatouage sonore, robuste contre 32 kbps

MP3, est plus bas que $50,3$ bps. En outre, nous avons montré que la méthode proposée est robuste contre le 24 kbps USAC (Unified speech and audio coding, modes prédictifs linéaires et Fourier) avec une charge utile moyenne de $5-15$ bps. En outre, il est démontré que la méthode proposée est robuste contre une variété des transformes tout en préservant la qualité.

4.2 Audio watermarking using spikegram and a two-dictionary approach

4.2.1 Abstract

This paper introduces a new audio watermarking technique based on a perceptual kernel representation of audio signals (spikegram). Spikegram is a recent method to represent audio signals. It is combined with a dictionary of gammatones to construct a robust representation of sounds. In traditional phase embedding methods, the phase of coefficients of a given signal in a specific domain (such as Fourier domain) is modified. In this paper, the watermark bit stream is inserted by modifying the phase and sign of gammatones. Moreover, the watermark is adaptively embedded only into kernels with high amplitudes where all masked gammatones have been already removed. Two embedding methods are proposed, one includes the watermark embedding into the sign of gammatones (one dictionary method) and the other one is based on watermark embedding into both sign and phase of gammatone kernels (two-dictionary method). The efficiency of the proposed spikegram watermarking is shown via several experimental results. First, robustness of the proposed method is shown against 32 kbps MP3 with an embedding rate of 56.5 bps. Second, we showed that the proposed method is robust against unified speech and audio codec (24 kbps USAC, Linear predictive and Fourier domain modes) with an average payload of 5-15 bps. Third, it is robust against simulated small real room attacks with a payload of roughly 1 bps. Lastly, it is shown that the proposed method is robust against a variety of signal processing transforms while preserving quality.

4.3 Introduction

An analysis by the Institute for Policy Innovation concludes that every year global music piracy is making 12.5 billion of economic losses, 71060 U.S. jobs lost, a loss of 2.7 billion in workers' earnings and a loss of 422 million in tax revenues, 291 million in personal income tax and 131 million in lost corporate income and production taxes. Most of the music piracy is because of rapid growth and easiness of current technologies for copying, sharing, manipulating and distributing musical data [Siwek, 2007].

As one promising solution, audio watermarking has been proposed for post-delivery protection of audio data. Digital watermarking works by embedding a hidden, inaudible watermark stream into the host audio signal. Generally, when the embedded data is easily removed by manipulation, the watermarking is said to be fragile which is suitable for authentication applications, whereas for copyright applications, the watermark needs to be robust against manipulations [Cox *et al.*, 2007]. Watermarking has also many other applications such as copy control, broadcast monitoring and data annotation [Steinebach and Dittmann, 2003; Boho and Wallendael, 2013; Majumder1 *et al.*, 2013]. For audio watermarking, several approaches have been recently proposed in the literature. These approaches include audio watermarking using phase embedding techniques [Arnold *et al.*, 2014], cochlear delay [M. Unoki, 2015], spatial masking and ambisonics [Nishimura, 2012], echo hiding [G. Hua and Thing, 2015a], [G. Hua and Thing, 2015b; Y. Xiang, 2015], patchwork algorithm [Xiang *et al.*, 2014b], wavelet transform [Pun and Yuan, 2013], singular value decomposition [Lei *et al.*, 2013] and FFT amplitude modification [D. Megas, 2010]. State of the art methods introduce phase changes in the signal representation (i.e., from the phase of the Fourier representation) [Arnold *et al.*, 2014], [Ngo and Unoki, 2015], while we adopt a more original strategy by using two dictionary of kernels and by shifting the sinusoidal term of the gammatones [Strahl and Mertins, 2009], [Slaney, 1998a].

There are two types of watermarking systems: zero-bit and multi-bit [Nikolaidis and Pitas, 2004]. For the former the goal is to verify the presence of the watermark stream while for the latter the goal is to decode watermark bits out of watermarked signals. In this paper, the watermarking is of multi-bit type and could be used for data annotation.

Multiple dictionaries for sparse representation applications has already drawn the attention of researchers in signal processing [Valiollahzadeh *et al.*, 2009], [Son and Choo, 2014], [Adler and Emiya, 2012], [Fevotte *et al.*, 2006]. For example, in [Valiollahzadeh *et al.*, 2009], authors propose a two-dictionary method for image inpainting where one decomposed image serves as the cartoon and the other as the texture image. Also, a watermark detection algorithm was proposed by Son et al. [Son and Choo, 2014] for image watermarking where two dictionaries are learned for horizontally and vertically clustered dots in the half tone cells of images. In [Fevotte *et al.*, 2006], authors propose an audio denoising algorithm using a sparse audio signal regression with a union of two dictionaries of modified discrete cosine transform (MDCT) bases. They use long window MDCT bases to model the tonal parts and short window MDCT bases to model the transient parts of the audio signals. In all mentioned methods, the goal is to have an efficient representation of the signal. However for audio watermarking, one goal is to manipulate the signal representation in a way to find adaptively the spectro-temporal content of the signal for efficient transmission of watermark bits.

In this paper, for the first time, we propose an embedding and decoding method for audio watermarking which jointly uses multiple dictionaries (including gammatones and their phase-shifted versions) and a spikegram of the audio signal. It has been shown in [Smith and Lewicki, 2005] that spikegram is time-shift invariant where the signal is decomposed over a dictionary of gammatones. To do so, we use the Perceptual Matching Pursuit (PMP) [Pichevar *et al.*, 2010a]. PMP is a bio-inspired approach that generates a sparse representation and takes into account the auditory masking at the output of a gammatone filter bank (the gammatone dictionary is obtained by duplicating the gammatone filter bank at different time samples).

The proposed method is blind, as the original signal is not required for decoding. Also, the only information needed to be shared between the encoder and the decoder include the key, which is used as the initial state for a Pseudo Noise (PN) sequence, the type and parameters for dictionary generation. To evaluate the performance of the proposed method, extensive experimental results are done with a variety of attacks.

Robustness against lossy perceptual codecs is a major requirement for a robust audio watermarking, thus we decided to evaluate the robustness of the method against 32 kps MP3 (although not used that often anymore, it is still a powerful attack which can be used as an evaluation tool). The proposed method is robust against 32 kbps MP3 compression with the average payload of 56.5 bps while the state of the art robust payload against this attack is lower than 50.3 bps [Khaldi and Boudraa, 2013]. In this paper, for the first time, we evaluate the robustness of the proposed method against USAC (Unified Speech and Audio Coding) [Quackenbush, 2013; Yamamoto *et al.*, 2013; Neuendorf *et al.*, 2009]. USAC is a strong contemporary codec (high quality, low bit rate), with dual options both for audio and speech. USAC applies technologies such as spectral band replication, CELP codec and LPC. Experiments show that the proposed method is robust against USAC for the two modes of linear predictive domain (executed only for speech signals) and frequency domain (executed only for audio signals), with an average payload of 5-15 bps. The proposed method is also robust against simulated small real room attacks [Lehman, 2016], [E. Lehmann and Nordholm, 2007] for the payload of roughly 1 bps. Lastly, the robustness against signal processing transforms such as re-sampling, re-quantization, low-pass filtering is evaluated and we observed that the quality of signals can be preserved.

In this paper, the sampled version of any time domain signal is considered as a column vector with a bold face notation.

4.4 Spikegram kernel based representation

4.4.1 Definitions

With a sparse representation, a signal $x[n], n = 1 : N$ (or $\boldsymbol{x}$ in vector format) is decomposed over a dictionary $\Phi = \{g_i[n]; n = 1 : N, i = 1 : M\}$ to render a sparse vector $\boldsymbol{\alpha} = \{\alpha_i; i = 1 : M\}$ which includes only a few non-zero coefficients, having the smallest reconstruction error for the host signal $\boldsymbol{x}$ [Smith and Lewicki, 2005], [Pichevar *et al.*, 2010a]. Hence,

$$x[n] \approx \sum_{i=1}^{M} \alpha_i g_i[n], \quad n = 1, 2, .., N \tag{4.1}$$

where α_i is a sparse coefficient. A 2D time-channel plane is generated by duplicating a bank of N_c gammatone filters (having respectively different center frequencies) on each time sample of the signal. Also, all the gammatone kernels in the mentioned 2D plane form the columns of the dictionary Φ (Hence, $M = N_c \times N$). Thus $g_i[n]$ is one base of the dictionary which is located at a point corresponding to channel $c_i \in \{1, .., N_c\}$, and time sample $\tau_i \in \{1, 2, .., N\}$ inside the 2D time-channel plane (Fig.4.1). The spikegram is the 2D plot of the coefficients at different instants and channels (center frequencies). The number of non-zero coefficients in α_i per signal's length N is defined as the density of the representation (note that sparsity = 1-density).

To compute the sparse representation in (4.1), many solutions have been presented in the literature including Iterative Thresholding [Blumensath and Davies, 2008], Orthogonal Matching Pursuit (OMP) [Joel A. Tropp, 2007], Alternating Direction Method (ADM) [S. Boyd, 2011], Perceptual Matching Pursuit (PMP) [Pichevar *et al.*, 2010a]. Here, we use the perceptual matching pursuit, because it is not computationally expensive, is a high resolution representation for audio signals, and generates auditory masking thresholds and removes the inaudible content under the masks [Pichevar *et al.*, 2010a]. Perceptual Matching Pursuit (PMP) is a recent approach which solves the problem in (4.1) for audio and speech using a gammatone dictionary [Pichevar *et al.*, 2010a] , [Najaf-Zadeh *et al.*,

Table 4.1 The effective lengths and center frequencies for gammatone kernels used in this work.

Channel number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Effective length (ms)	55.2	39.1	31.0	25.8	22.0	18.7	16.1	14.0	12.1	10.7	9.3	8.1	7.2	6.3	5.4	4.7	4.1	3.5	2.9	2.4	2.0	1.7	1.4	1.0	0.8
Center frequency (Hz)	50	150	250	350	450	570	700	840	1k	1.2k	1.4k	1.6k	1.9k	2.2k	2.5k	2.9k	3.4k	4k	4.8k	5.8k	7k	8.5k	10.5k	13.5k	18.8k
											0.20	0.18	0.13	0.11	0.09	0.07	0.06	0.04	0.03	0.03	0.03	0.02	0.02	0.00	0.00

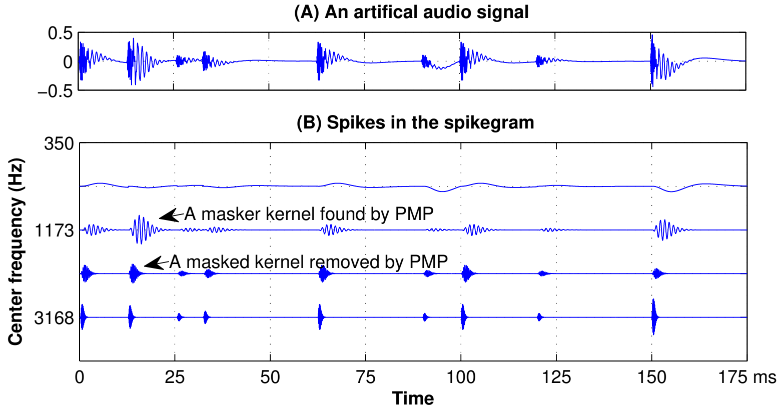

Figure 4.1 A 2D plane of gammatone kernels of a spikegram generated from PMP [Pichevar *et al.*, 2010a], [Najaf-Zadeh *et al.*, 2008] coefficients. The 2D plane is generated by repeating $N_c = 4$ gammatones at different channels (center frequencies) and at each time samples. A gammatone with non-zero coefficient is called a spike.

2008]. PMP is a greedy method and an improvement over Matching Pursuit [Mallat and Zhang, 1993]. PMP finds only audible kernels for which the sensation level is above an iteratively updated masking threshold and neglects the rest. A kernel is considered as a masked kernel if it is under the masking of (or close enough in time or channel to) another masker kernel with larger amplitude. The efficiency of PMP for signal representation is confirmed in [Pichevar *et al.*, 2010a] and [Najaf-Zadeh *et al.*, 2008]. The gammatone filter bank (used to generate the gammatone dictionary) is adapted to the natural sounds [Smith and Lewicki, 2005] and is shown to be efficient for sparse representation [Pichevar *et al.*, 2010a]. A gammatone kernel equation [Strahl and Mertins, 2009] has a gamma part and a tone part as below

$$g[n] = an^{m-1}e^{-2\pi ln}cos[2\pi(f_c/f_s)n + \theta], n = 1, .., \infty \qquad (4.2)$$

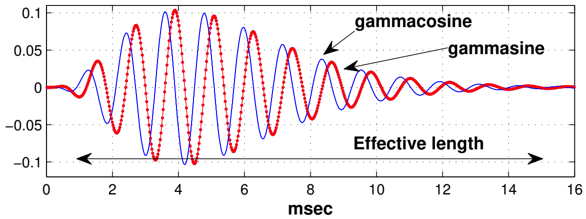

Figure 4.2 A sample gammacosine (blue) and gammasine(red) (for channel-8) with a center frequency of 840 Hz and an effective length of 13.9 msec. Gammasines and gammacosines are chosen in the watermark embedding proceess based on their correlation with the host signal and the input watermark bit. The sampling frequency is 44.1 kHz.

in which, n is the time index, m and l are used for tuning the gamma part of the equation. f_s is the sampling frequency, θ is the phase, f_c is the center frequency of the gammatone. The term a is the normalization factor to set the energy of each gamatone to one. Also, the effective length of a gammatone is defined as the duration where the envelope is greater than one percent of the maximum value of the gammatone. In this paper, a 25-channel gammatone filter bank is used (Table 4.1). Their bandwidths and center frequencies are fixed and chosen to correspond to 25 critical bands of hearing. They are implemented at the encoder and the decoder using (4.2). Also, a gammatone is called a gammacosine when $\theta = 0$ or a gammasine when $\theta = \pi/2$. In Table 4.1, center frequencies and effective lengths for some gammatones, versus their channel numbers are given. In Fig. 4.2, channel 8 gammasine and gammacosine are plotted.

4.4.2 Good characteristics of spikegram for audio watermarking

— Time shift invariance:
The spikegram representation in (4.1) is time shift invariant (see [Smith and Lewicki, 2005] for the proof). Therefore, it is suitable for robust watermarking against time shifting de-synchronization attack.

— Low host interference when using spikegram:
In (4.1), many gammatones have either zero coefficients or masked, thanks to PMP. Therefore, compared to traditional transforms such as STFT and Wavelet transforms, spikegram is expected to yield less host interference at the decoder (see Fig.13 and Fig.14 for experiments regarding the dependence of the error rate and the quality with the sparsity of the spikegrams).

— Efficient embedding into robust coefficients:
The watermark bits are inserted only into large amplitude coefficients obtained by PMP, where all inaudible gammatones have been a priori removed from the representation.

4.5 Two-Dictionary Approach

The watermark bit stream is symbolized by $\boldsymbol{b}$ which is an $M_2 \times 1$ vector $(M_2 < M)$. The goal is to embed the watermark bit stream into the host signal. $\boldsymbol{K}$, a $P \times 1$ vector $(P < M_2)$, is the key which is shared between the encoder and the decoder of the watermarking system. Also, the sparse representation of the host signal $\boldsymbol{x}$ on the gammacosine dictionary (i.e., α_i) is assumed to be known.

The proposed method relies on the fact that the change in signal quality should not be perceived when changing the phase of specific gammatone kernels. Moreover, it is called a two dictionary approach, as a candidate kernel for watermark insertion, is adaptively selected from a gammacosine or gammasine dictionary. Note that compared to traditional phase modulation techniques which impose phase modulation on the block based representation of signal on tones, here we make phase modification on gammatones in the sparse kernel based representation of signal.

4.5.1 Multi-bit watermarking using the spikegram

For multi-bit watermarking, the host signal $x[n]$ ($\boldsymbol{x}$ in vector format) is first represented using the kernel based representation in (4.1). First, M_2 gammatones $g_k[n]$ from the representation in (4.1) are selected (the selection of watermark kernels is explained in section 4.5.5). These gammatones form the watermark dictionary D_1 and carry the watermark bit stream $b_k, k = 1, 2, .., M_2$. Other $M_1 = M - M_2$ kernels form the signal dictionary D_2. The signal and watermark dictionaries are disjoint subsets of the gammatone dictionary used for sparse representation in (4.1), thus $D_1 \cap D_2 = \emptyset$. Each watermark bit b_k serves as the sign of a watermark kernel. Hence (4.1) becomes

$$y[n] = \sum_{i=1}^{M1} \alpha_i g_i[n] + \sum_{k=1}^{M2} b_k \left|\alpha_k\right| g_k[n] \tag{4.3}$$

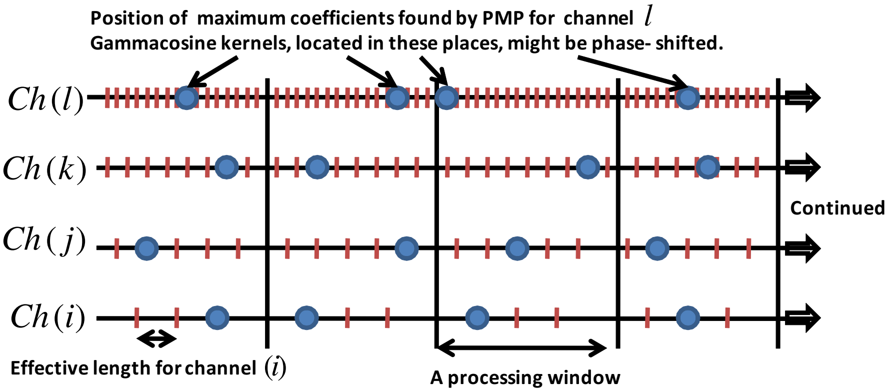
Position of maximum coefficients found by PMP for channel l
Gammacosine kernels, located in these places, might be phase- shifted.
$Ch(l)$
$Ch(k)$
$Ch(j)$
$Ch(i)$
Continued
Effective length for channel (i)
A processing window

Figure 4.3 Watermark insertion using the two-dictionary method. First, the spikegram of the host signal is found using PMP with a dictionary of 25-channel gammacosines, located at each time sample along the time axis. Then for each processing window and each channel and based on the embedding bit b, the gammacosine, or gammasine (located at a blue circle) with maximum strength factor (m_c or m_s) is chosen for the watermark insertion. In this work, gammatone channels $Ch's$ are selected in the range of 1-4 and 9-19 (odd channels only) for the watermark insertion. Also, to get the same embedding strength for different embedding channels, processing windows of different channels have the same length.

where $y[n]$ is the watermarked signal. In (4.3), if the watermark and signal dictionaries use the same gammatone kernels, the watermarking becomes a one dictionary method. In one dictionary method, the watermark bits are inserted as the sign of gammatone kernels. In two dictionary method, in addition to the manipulation of the sign of gammatone kernels, their phase also might be shifted as much as $\pi/2$, based on the strength of the decoder. Hence, for the two-dictionary approach, each watermark kernel is chosen adaptively from a union of two dictionaries, one dictionary of gammacosines and one dictionary of gammasines.

The p^{th} watermark kernel in the watermark dictionary is found adaptively and symbolized with $\boldsymbol{f}_p$ which is either a gammasine or a gammacosine.
Thus for the two dictionary method, the embedding equation in (4.3) becomes

$$y[n] = \sum_{i=1}^{M1} \alpha_i g_i[n] + \sum_{k=1}^{M2} b_k \left|\alpha_k\right| f_k[n] \tag{4.4}$$

and for the decoding of the p^{th} watermark bit, we compute the projections of the watermarked signal on the p^{th} watermark kernel as follows

$$\begin{aligned} < \boldsymbol{y}, \boldsymbol{f}_p >= \sum_{i=1, i \neq p}^{M1} \alpha_i < \boldsymbol{g}_i, \boldsymbol{f}_p > + \\ b_p \left|\alpha_p\right| + \sum_{k=1, k \neq p}^{M2} b_k \left|\alpha_k\right| < \boldsymbol{f}_k, \boldsymbol{f}_p > \end{aligned} \tag{4.5}$$

The number of samples used to compute the projection in (4.5) is equal to the gammatone effective length. The goal is to decode the watermark bit as the sign of the projection $< \boldsymbol{y}, \boldsymbol{f}_p >$. We later show how to find the best watermark kernels so that the first two terms in the right side of (4.5) have the same signs as the watermark bit b_p. The right term in the right side of (4.5) is the interference the decoder receives from other watermark bit insertions. To remove this interference term, the watermark gammatones should be uncorrelated. In fact, to design the watermark dictionary, we choose a subset of the full overcomplete dictionary in such a way that the watermark kernels are spectro-temporally far enough and hence uncorrelated. Thus the watermark bits will be decoded independently even if there are correlations between watermark and signal gammatones, that are shown in (4.7). Hence, in Fig. 4.3, for each channel and time sample, two neighbor watermark kernels should be separated with at least one effective length and at least one channel. Note that with this assumption, the correlation between watermark gammatones will be less than 0.02. As embedding of multiple watermark bits are performed independently, thus in next, only the single bit watermarking using the two dictionary method is explained in next sections.

4.5.2 The proposed one-bit embedder

Equation (4.1) is used to resynthesize the host signal $\boldsymbol{x}$ from sparse coefficients and gammacosines.

Now, we want to embed one bit $b \in \{-1, 1\}$ from the watermark bit stream $\boldsymbol{b}$ by changing the sign and/or the phase of a gammacosine kernel $\boldsymbol{g}_p$ (the p^{th} kernel found by PMP, still to be determined later in this section) with amplitude α_p (to be determined) located at a given channel and processing window (each processing window is a time frame including several effective lengths of a gammatone, Fig.4.3).
To find an efficient watermark kernel $\boldsymbol{f}_p$ which bears the greatest decoding performance for the watermark b, we write the 1-bit embedding equation as follows

$$y[n] = \sum_{i=1, i\neq p}^{M} \alpha_i g_i[n] + b\,|\alpha_p|\, f_p[n] \tag{4.6}$$

where the watermarked kernel $\boldsymbol{f}_p$ for a given channel number can be a gammacosine ($\boldsymbol{g}c$) or a gammasine ($\boldsymbol{g}s$) which are zero and $\pi/2$ phase-shifted versions of the original gammatone kernel $\boldsymbol{g}_p$, respectively. The correlation between the watermarked signal $\boldsymbol{y}$ and the watermarked kernel $\boldsymbol{f}_p$, is found as below

$$< \boldsymbol{y}, \boldsymbol{f}_p >= \sum_{i=1, i\neq p}^{M} \alpha_i < \boldsymbol{g}_i, \boldsymbol{f}_p > +b\,|\alpha_p| \tag{4.7}$$

Hence, to design a simple correlation-based decoder, the sign of the correlation in the left side of (4.7) is considered as decoded the watermark bit. In this case, for correct detection of the watermark bit b, the interference term should not change the desired sign at the right hand side of (4.7). Moreover, the gammatone dictionary is not orthogonal, hence the left term in the right side of (4.7) may cause erroneous detection of b. For a strong decoder, two terms on the right side of (4.7), should have the same sign with large values. We later show that by finding an appropriate gammacosine or gammasine in the spikegram, the right side of (4.7) can have the same sign as the watermark bit b. In this case, the module of correlation in (4.7) is called watermark strength factor m_p for the bit b and a greater strength factor means a stronger watermark bit against attacks. In this case, (4.7) becomes

$$< \boldsymbol{y}, \boldsymbol{f}_p >= bm_p \tag{4.8}$$

For a large value strength factor (and with the same sign of the watermark bit), we search the peak value of the projections when a gammatone candidate (gammacosine or gammasine) is projected to each column of the dictionary. Thus, for a given channel, a processing window and watermark bit b, we do the following procedure to find the phase, position and the amplitude of the watermarked kernel $\boldsymbol{f}_p$ (Fig. 4.4). For a given channel, we consider the watermark gammatone candidate $\boldsymbol{f}_p$ (the p^{th} gammatone kernel in the signal representation of (4.1)) to be a gammacosine $\boldsymbol{g}c$ or a gammasine $\boldsymbol{g}s$. Then, do the following steps for both gammasine and gammacosine candidates:

— Shift the watermark gammatone candidate $\boldsymbol{f}_p$ alongside all processing windows, at time shifts equal to multiples of the gammatones' effective length. For each shift compute the correlation of the watermarked signal with the sliding watermark candidate kernel. Then, find the absolute maximum of the correlation (watermark strength factor) $\left|< \boldsymbol{y}, \boldsymbol{f}_p >\right|$ using (4.7) (Fig.4.3). The result is a strength factor,

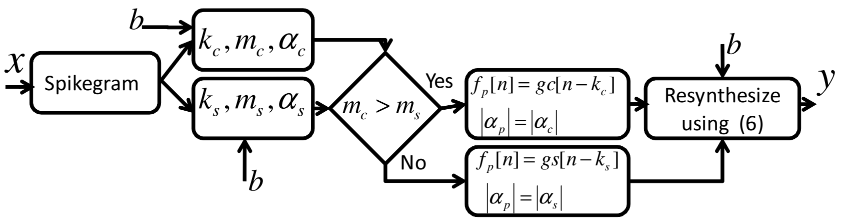

Figure 4.4 The proposed embedder for a given channel and processing window. The gammasine or gammacosine with maximum strength factor is chosen as the watermark kernel and its amplitude is set to its associated sparse coefficient in the spikegram. Finally (4.6) is used to resynthesize the watermarked signal $\boldsymbol{y}$ (in vector format). m_s and m_c are respectively the strength factors for gammasine candidate and gammacosine candidate.

symbolized as m_c for gammacosine, located at time sample k_c with amplitude α_c and also another strength factor, symbolized as m_s for a gammasine kernel located at k_s with the amplitude α_s. Thus $m_c = |< \boldsymbol{y}, gc[n - k_c] >|$, $m_s = |< \boldsymbol{y}, gs[n - k_s] >|$.

— Afterwards, the gammacosine or gammasine with greater strength factor is chosen as the final watermark gammatone $\boldsymbol{f}_p$ and its time shift (sample), amplitude and phase are registered. Gammatone or gammasine with greater strength factor is chosen as the final watermark gammatone $\boldsymbol{f}_p$ with the final watermark strength factor being $m_t = max(m_c, m_s)$. The respective k_c or k_s, amplitude α_c or α_s and phases are kept. Therefore, the algorithm finds the optimal watermark gamatone from two dictionaries including one dictionary of gammacosines and one dictionary of gammasines.

— After all, the watermarked signal is synthesized using (4.6), $\boldsymbol{f}_p$ with its amplitude set to $b\,|\alpha_p|$. This is equivalent to finding a time-channel point in the spikegram (Fig.4.3) where the optimal position for embedding is found. Then, $\boldsymbol{g}_p$ is replaced with the watermark gammatone $\boldsymbol{f}_p$.

4.5.3 The proposed one-bit decoder

At the decoder, the same search procedure, used in the embedder to find the watermarked kernel candidate, is applied. Therefore for a given channel and processing window, the decoding procedure is shown Fig. 4.5.

For a given channel, suppose the watermark gammatone candidate $\boldsymbol{f}_p$ to be a gammacosine $\boldsymbol{gc}$ or a gammasine $\boldsymbol{gs}$. Then do the following steps:

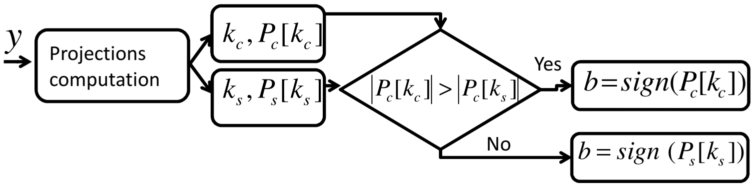

Figure 4.5 The proposed one-bit decoder. The projections with maximum absolute value for gammacosines and gammasines are found as $P_c[k_c]$ and $P_s[k_s]$ and at time sampes k_c and k_s respectively. The watermark bit $\hat{b}$ is considered as the sign of the projection with the largest absolute value.

- Shift the watermark kernels $\boldsymbol{g}c$ and $\boldsymbol{g}s$ alongside the processing window at time shifts (respectively k_c or k_s) equal to multiples of the gammatone's effective length. For each k (either k_c or k_s), compute the correlation of the watermarked signal with the sliding watermark kernel candidate.
- Then, find the absolute maximum of the correlation $P_c[k] = |< \boldsymbol{y}, gc[n-k] >|$ and $P_s[k] = |< \boldsymbol{y}, gs[n-k] >|$ (Fig.4.5). The result is one absolute maximum correlation $m_c = max(|P_c[k]|)$ for a gammacosine located at the time sample k_c with the amplitude α_c and also another absolute maximum correlation $m_s = max(|P_s[k]|)$ for a gammasine kernel located at k_s with the amplitude α_s.
- Finally, if $|m_c| > |m_s|$ then $b = sign(P_c[k_c])$ otherwise $b = sign(P_s[k_s])$.

In Fig.4.6, the watermark strength factor is plotted versus the embedding channel ranges for the cases of one dictionary (when using only gammacosine kernels) and two dictionaries (when both gammasine and gammacosine kernels are used). As expected, the strength factors for the two-dictionary method is all the time greater than the one for one dictionary method. The maximum improvement occurs for middle channels between 5 and 16. Also, towards greater embedding channel ranges, the strength factor becomes smaller. This is because, for the outside of this channel range, we have usually small coefficient values, which do not contribute too much to the strength of the watermarking systems.
For an insight about the robustness of the methods against 20 dB additive white Gaussian noise (AWGN), the average maximum peaks of the noise signal is plotted as a horizontal line. As is seen, the embedding channel ranges which are robust against 20 dB white Gaussian noise are wider for two dictionary method compared to one dictionary method (for one dictionary method it is below channel number 16 while for the two-dictionary method, it spans 1-24 channel range.)

As illustrated in Fig.4.3, for channel j, each processing window includes several effective lengths of the gammatone, l_j. Thus, for each channel j, the algorithm searches for a

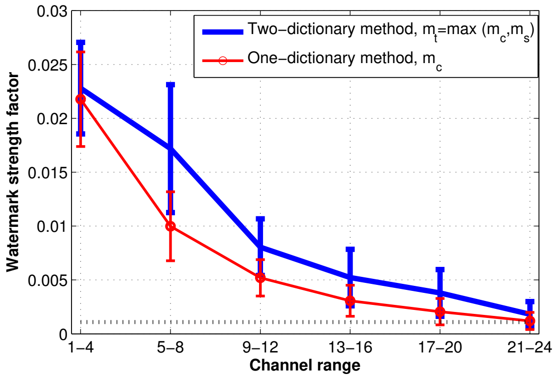

Figure 4.6 The average watermark strength factor for one-dictionary (only gammacosine used) and two-dictionary (gammacosine and gammasine used) methods versus embedding channel range. Simulations are done on 5 hours (100 signals, 3 minutes each) of different music genres and English voices. In each embedding /decoding experiment, watermark is inserted in a specific channel range with a 45 msec processing windows. The maximum peak of the noise with 20 dB AWGN is also plotted as a horizontal dashed line in black. The 95% confidence intervals are plotted as vertical lines at the center of the average results for three consequensive channels. Input signals are normalized to unit l^2 norm.

watermark gammatone candidate among $\lceil \frac{L_P}{l_j} \rceil$ gammatones (Fig. 4.3, vertical red lines in each processing window).

Thanks to the content-based aspect of the approach, phase shifts and sign changes occur adaptively. Therefore, depending on the signal, change in the sign and phase of the gammatone is not necessary when generating the watermark. In that situation, some watermark gammatones are similar in shape and phase to signal gammatones. This is one of the good features of the proposed method which contributes to the quality of the watermarked signals.

4.5.4 Robustness of the proposed method against additive white Gaussian noise

We consider the robustness of the proposed method against additive white Gaussian noise (AWGN) as the basic criterion for robust watermarking. For robust watermarking evaluation, we suppose that an AWGN $z[n]$ is added to the watermarked signal $\boldsymbol{y}$ of (4.6).

Therefore, (4.8) becomes

$$r =< \boldsymbol{y}, \boldsymbol{f}_p >= m_p b + < \boldsymbol{z}, \boldsymbol{f}_p > \tag{4.9}$$

in which, r is the correlation computed for the decoding of bit b, $\boldsymbol{z}$ is the AWGN, with mean zero and variance σ_n^2. As the gammatone kernel $\boldsymbol{f}_p$ has zero mean and unit variance, mean and variance of $< \boldsymbol{z}, \boldsymbol{f}_p >$ are respectively 0 and σ_n^2. It is assumed that for each channel number c, the strength factor m_p are samples from a white Gaussian random process with a specific mean λ and variance σ^2 (Fig.4.6). Assuming that the decorrelation between the watermark strength factor for channel k and noise z_n, the mean $m_r b$ and variance for the correlation term r in (4.9) are, $m_r = m_p b$ and $\sigma_r^2 = \sigma_n^2 + \sigma^2$. Therefore, by considering the decoded watermark bit $\hat{b} = sign(< \boldsymbol{y}, \boldsymbol{f}_p >)$, the error probability of the decoder for channel c is as below

$$\begin{aligned} p_k = Pr\{\hat{b} < 0 \mid b = 1\} = \tfrac{1}{2}\text{erfc}(\tfrac{m_r}{\sigma_r\sqrt{2}}) \\ = \tfrac{1}{2}\text{erfc}(\tfrac{\lambda_k}{\sqrt{2(\sigma_n^2+\sigma^2)}}) \end{aligned} \tag{4.10}$$

Where $erfc(.)$ is the complementary error function. In (4.10), the probability of error is low when we have larger mean and smaller variance for each channel's strength factor. In Fig.4.7, the estimated error probability of the decoder in (4.10) is plotted versus the embedding channel number and different levels of signal to noise ratio (SNR). As is seen, for 20 dB SNR, the error probability of the first 20 channels stay below .04. Moreover, an additive noise with lower SNR has a more destroying effect on channels with higher center frequencies.

4.5.5 Designing efficient dictionaries for high quality robust watermarking

To design the watermark dictionary, we consider three conditions. First, embedding into low frequency channels (1-4) and high frequency channels (greater than 9) is preferred. Based on our empirical results, we do not add watermarks in channels 5 to 8, because of the energy greater sensitivity of the ear in this channel range.

Second, embedding into lower channels bears more robustness against AWGN attack (Fig.4.6). Lastly, watermark gammatone kernels should be uncorrelated. They should be separated at least by one channel (along the channel axis) and one effective length (along

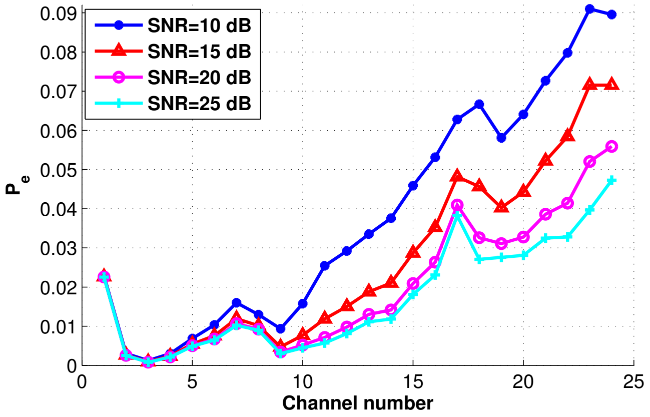

Figure 4.7 The error probability of the decoder under additive white Gaussian noise with different signal to noise ratios. The mean and variance of the watermark strength factor are estimated from 100 signals including music and English voices, 3 minutes each.

the time axis). The final implementation uses channels 2, 4, 9, 11, 13, 15, 17 and 19 for watermark insertion.

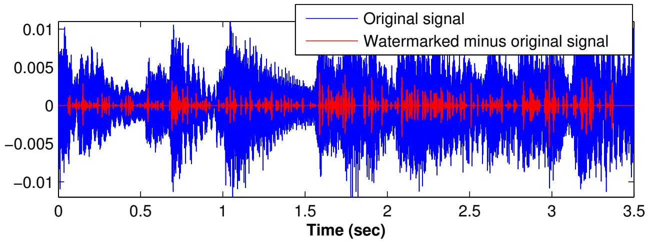

Figure 4.8 The time domain waveforms for the original signal (blue) and the difference between the watermarked and the original signal (red). The original signal includes a solo harpsichord instrument sampled at 44.1 kHz. Each gammatone (a spike) in the difference signal (red) indicates the insertion of one watermark bit.

4.6 Experimental Setup

Table 4.2 lists the simulation conditions.

Test signals

For the quality test, ABC/HR tests [ITU, 1997] are done on pieces of 10 seconds of 6 types of audio signals: Pop, Jazz, Rock, Blues, Speech (in French) and Classic. Titles of the musical pieces are listed in Table 4.3. For the robustness test, simulations are done on 100 audio signals, 3 minutes each (5 hours in total) including different music genres and English voices. Each signal is sampled at 44.1 kHz and has 16-bit wave format. A sample original signal and the original signal minus the watermarked one are plotted in Fig.4.8.

Sparse signal representation using PMP and gammatone dictionary

In all experiments, the sparsity of PMP representation is 0.5. A bank of 25 gammacosine kernels distributed from 20 Hz to 20 kHz is implemented to generate spikegrams according to the conditions given in Table 4.2.

Resynchonization and Watermark stream generation

The first 13 bits of the embedded bit stream per each second is devoted to the synchronization Barker sequence +1+1+1+1+1-1-1+1+1-1+1-1+1 [Borwein and Mossinghoff, 2008]. This allows synchronization between the decoder and encoder. The other bits in each second are devoted to the watermark bit stream $\boldsymbol{b}$. For robustness against cryptographic attacks [Voloshynovskiy *et al.*, 2001], at each frame, the watermark bit stream $\boldsymbol{b}$ is multiplied by a Pseudo Noise (PN) sequence [Klein, 2013b] $\boldsymbol{p}$ in which $p_i \in \{-1, 1\}$. Thus each embedded bit in the watermark stream will be $b_i p_i$.

For synchronization between the encoder and the decoder, we define a 1000 ms rectangular sliding window, multiply it to the watermarked signal at the decoder and decode the watermark bits from all 22 successive processing windows in the sliding window. Note that each 1 second sliding window includes 22 processing windows. Then, if the Barker code is decoded with more than 75 % accuracy, then the rest of decoding is performed. If the synchronization Barker code is not acquired, the sliding window is shifted and the same mentioned procedure is continued. Note that the signal is not shifted for synchronization, thanks to the time shift invariance property of spikegram representation [Smith and Lewicki, 2005]. Moreover, using the mentioned synchronization approach, all 45 ms processing windows inside each 1000 ms sliding window are also synchronized. A corruption in one processing window, might result in at most 45ms$\times$44100 $=$ 1984 corrupted samples. In this case, to resynchronize the decoder with the embedder, there might be a need to search for the barker code with 1984 shifting of the sliding window. As the decoding is done in real time, the resynchronization procedure is not computationally expensive. For resynchronization of critically time rescaled watermarked signals, a search approach to find the best time rescaling ratio, could be applied as in [Arnold *et al.*, 2014].

Table 4.2 The default computer simulation conditions

Quality test	Audio signals of Table 4.3, 10 seconds each
Quality measure	Subjective difference grade [ITU, 1997]
Robustness test	100 audio signals, 3 minutes each
	50 speech signals [VOA, 2015], 50 music signals [Bensound, 2015]
Signal characteristics	sampled at 44.1 kHz, quantized at 16 bits
Processing window	45 msec
2D spikegram	25-channel gammatone filter bank [Strahl and Mertins, 2009]
	repeated each time sample
Sparse representation	PMP on 10- second frames, with, 50% sparsity
Synchronization code	13-bit Barker sequence [Borwein and Mossinghoff, 2008]
Robustness measure	Bit Error Rate (BER)

Table 4.3 Excerpts of 10 seconds from these audio files are chosen for the ABC/HR [ITU, 1997] listening test, and the ODG results after watermarked ($\gamma = .01$)

Audio Type	Title (author or group Name)	ODG
POP	Power of love (Celine Dion)	-.41
Classical	Symphony No. 5 (Beethoven)	-.85
Jazz	We've got (A tribe Called Quest)	-.37
Blues	Bent Rules (Kiosk)	-.23
Rock	Enter Sad man (Metallica)	-.69
Speech	French Voice (A Female)	-.1

Finally, for the extraction of the watermark bits, the decoder uses the key ($\boldsymbol{K}$) to generate a PN sequence $\boldsymbol{p}$. Then each p_i multiplies with its bit stream $b_i p_i$ to find the watermark bit b_i (hint: $b_i p_i p_i = b_i$). If the watermarked signal is shifted one sample along the time axis, then the required time for the resynchronization equals the decoding time of one input frame (1 second) minus the preprocessing time.

4.7 Experimental Evaluation

As a preprocessing task both for the encoder and the decoder, a linear feedback shift register (LFSR) should be designed to generate a PN sequence. The key $\boldsymbol{K}$ includes $\log_2(M_2)$ bits as the initial state of the LFSR. It also comprises two decimal digits associated to the spikegram generation, including, number of channels N_c and time shifts q (in this work, $N_c = 25$, $q = 1$) meaning two "7-bit" ASCII codes. Thus 14 bits are devoted to the spikegram parameters. In total, the key includes $14+\log_2(M_2)$ bits. The spikegram

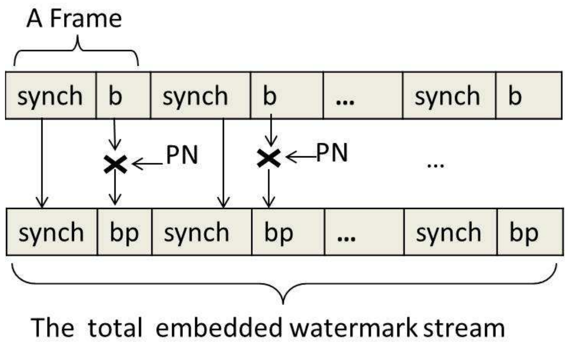

Figure 4.9 Generation of the embedded watermark stream from the watermark bits and synchronization code. A 13-bit synchronization Barker code is inserted into each second of the signal. The watermark bits in each frame are multiplied by a PN sequence.

parameters and the initial state of the LFSR are part of the key which increases the robustness of the proposed method against cryptographic attacks.

4.7.1 Quality

The embedding channels 2, 4, 9, 11, 13, 15, 17, 19 are selected for watermark insertion. In the embedding channels from 9 to 19, each watermark bit is inserted through three watermark kernels with the highest strength factors. However, for the first two channels (2 and 4), one watermark bit is inserted in each processing window. Therefore, the total number of watermark insertions in each processing window L_P (in second) is $2 + 6 \times 3$ bits, and we have $1/L_P$ processing windows per second. Hence, the total number of embedded bits per second is $M_2 = 20/L_P$, while the number of distinct embedded watermark bits per second is $8/L_P$.

Moreover, the total distortion depends on the quality of PMP representation. As the PMP coefficients for the silent parts are zero, the proposed method also does not insert watermark into the silent parts of the signal. Thus the watermarking payload is calculated for the non-silent parts of the signal. As the PMP coefficients change from signal to signal, the robustness and the quality of the algorithm is also content dependent.

To assess the quality of the watermark signals, ABC/HR listening tests were conducted based on ITU-R BS.1116 [ITU, 1997] on segments of 10 seconds of audio signals given in Table 4.3. Experiments are conducted for different embedding percentage γ (γ is the percentage of embeddings per one second frames and equals $M_2/44100$. Note that, γ is different from payload). For the quality measurement, fifteen random subjects (varying from experts to people with no experience in audio processing, aged between 20-50 including male and female) participated in 5-scale ABC/HR tests by listening to signals using bayerdynamic DT250 headsets in a soundproof listening room. In Fig.4.10, the average

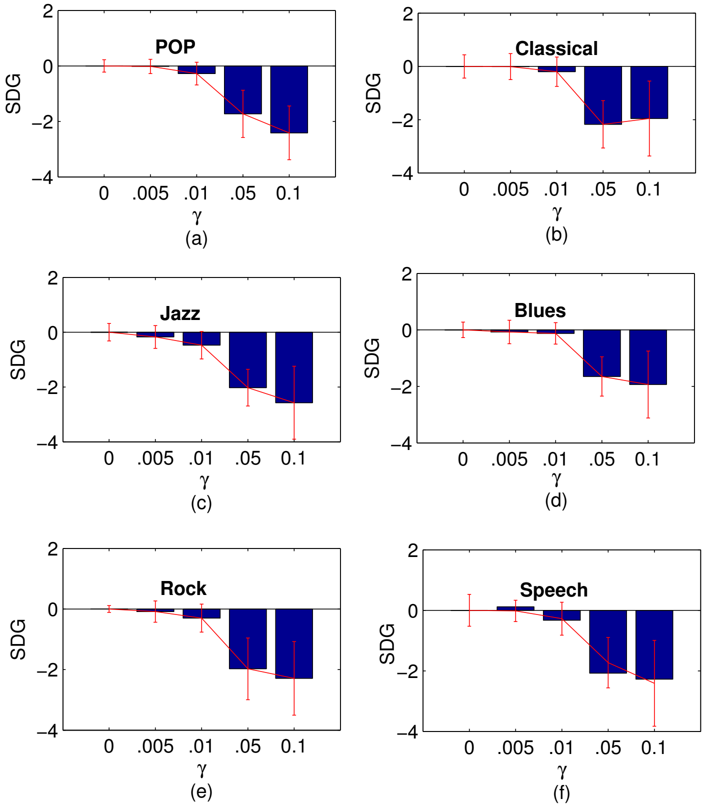

Figure 4.10 Subjective difference grade (SDG) as a function of embedding percentage factor γ for 4 minutes of each audio clip, described in Table 4.3, including POP (a), Classical (b), Jazz (c), Blues (d), Rock (e), Speech (f). The bottom ends of the bars indicate SDG means and the vertical red line segments represent the 95% confidence intervals surrounding them. $\gamma = 0$ indicates the original signals and $\gamma = .01$ is also used to generate watermarked signals for the robustness test.

subjective difference grade (SDG) [ITU, 1997] for several types of test signals in respect with the embedding percentage factor γ is plotted. The tips of the bar charts and the vertical red line segments on them indicate the mean SDG values and their associated 95 % confidence intervals respectively. The SDG indicates the difference between the average quality grade of the watermarked signal (given by listeners) minus the quality grade of the original signal (which is zero). The SDG is a quality difference grade between zero and -5 and a SDG strictly smaller than -1 means low quality.

As is seen from Fig.4.10, by increasing the embedding percentage factor γ, the quality of watermark signals, except for the classical audio, degrades and the confidence interval widens. For the classical audio, by increasing the embedding percentage γ from .05 to .1, the average SDG, rated by listeners, improves. One reason for this is because the selected classical signal in our test includes no silent and has a noise like spectrum. Thus adding a small amount of watermark noise (by increasing γ) might not be exactly perceived by the listeners.

In all results of Fig.4.10, when γ is not higher than .01, the SDG is greater than -0.5 and confidence intervals are smaller than 0.5 (vertical red lines) and cross the line SDG $= 0$. Thus for the robustness test, to ensure high quality for the watermark signals, γ is set to .01.

A sample of 10 seconds of each original signal type and its associated watermarked signal can be downloaded at the link: http://alum.sharif.ir/~yousof_erfani/

The signals in Table 4.3 are watermarked with $\gamma = .01$, their objective difference grade (ODG) results are computed using the open source PEAQ [Kabal, 2002] test and reported in Table 4.3.

4.7.2 Payload and Robustness

The payload (bit rate) of the method is defined as the number of watermark bits embedded inside each second of the host signal while these bits are decoded accurately at the decoder. For the case of $\gamma = .01$, the number of watermark kernels is $M_2 = .01 * 44100 = 441$. Hence, the processing window length equals $L_P = 20/441 = 45msec$ and the maximum attainable payload for the proposed method is $8/45msec = 177$ bps. Fig.4.11 shows how many embedded watermark bits are perfectly recoverable under different types of attacks.

Table 4.4 Parameters used for the attack simulations

Attack	Condition	set up values
No Attack	-	-
Resampling	Frequency (KHz)	22, 16
Requantization	bits	8, 12
Low-pass filtering	2^{th} Butterworth, Cut-off	0.5, 0.7× $22.05Hz$
Additive white noise	SNR (dB)	20, 25, 30
MP3 compression	Bit rate (kbps)	64, 32
Random cropping	Cropping per total length	0.125%, 0.250%
Amplitude Scaling	Scale ratio	0.5, 2
Pitch scaling	Scale ratio	0.95, 1.05
Time scaling	Scale ratio	0.95, 1.05

We use the same number of watermark kernels both for higher and lower bit rates for the results reported on Fig.4.11. Hence, in the case of lower bit rates, we use larger repetitive coding factor. Therefore, the watermark decoder for lower bit rates is stronger compared to the case of higher bit rate embeddings. The high quality of the average watermarked signals are confirmed for a payload of 177 bps ($\gamma = .01$) in Fig.4.10 and therefore for other bit rates in Fig.4.11.

The Bit Error Rate (BER) of the decoder is defined as the number of erroneously detected bits at the decoder per all embedded bits. In Fig. 4.11, to test the robustness of the proposed method, the BER was computed for a variety of attacks including: noise addition, MP3 compression, re-sampling, low-pass filtering and re-quantization. The parameter setting for each attack is given in Table 4.4.

The audio editing tools used in the experiment are CoolEdit 2.1 [CoolEdit, 2013] (for re-sampling and re-quantization) and Wavepad [WavePad, 2013] Sound Editor for MP3 compression. Other attacks of Table 4.4 are written in MATLAB [Matlab, 2014]. In addition, for all attacks, frame synchronization is performed using the resynchronization approach mentioned in section IV.3. In Fig.4.11, the most powerful attacks are the MP3 32 kbps (with average robust payload of 56.5 bps) and the MP3 64 kbps (with average robust payload of 77 bps). The proposed method has robustness against low-pass filtering, with a robust payload greater than 89 bps. Also, the robust payload for all other attacks is around 95 bps.

Moreover, random cropping is done by setting to zero, 0.125 or 0.250 percent of the signal's samples, at random places, and in every 1-second frame. By random cropping, we have 55 or 100 corrupted samples per second. Hence, random cropping changes the spikegram coefficients obtained by PMP at the decoder. However, the decoder searches for high peaks in the spikegram which are robust to mild modifications (i.e., very low value coefficients are very prone to cropping). As we increase the percentage of cropped samples, we expect more degradation on high value coefficients and hence more BER.

As an interesting observation, when a 10 dB additive white Gaussian noise was added to the signal, we observed that more than 90 percent of the error occurs because of the misdetection of the location and type of correct gammatone (gammacosine versus gammasine) at the decoder. This is because under attacks, many peak amplitudes in the projection search space might be very close to the true peak that might be misdetected. Under moderate attacks, it is less probable that the sign of the high amplitude peaks be changed.

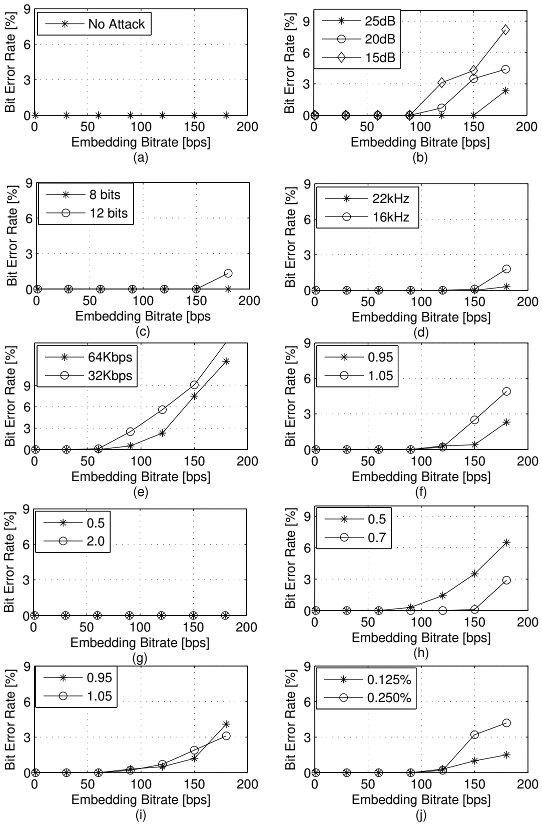

Figure 4.11 Robustness test results. 5 hours of different music genres and English speech (100 signals, 3 minutes each) with conditions presented in Table 4.2 are watermarked. The average BER versus the payload (bps) is plotted when the watermarked signals are exposed to the attacks (with conditions listed in Table 4.4) including (a) No attack (b) AWGN (c) Re-quanitization (d) Re-sampling (e) MP3 compression (f) Pitch scaling (g) Amplitude scaling (h) LPF(i) Time scaling (j) Random cropping. $\gamma = .01$, sparsity of the signal coefficient vector is forced to 0.5. The robust payload for the no-attack, and amplitude scaling conditions is 177 bps.

In Fig.4.12, the average bit error rate of the decoder is plotted versus the attack strength level for important attacks including time rescaling (with rescaling factors between 1.05 and 1.13), cropping (with corruption between 0.64% and 0.32%) and LPF (low pass filtering, 2^{th} order Butterworth with cut-off frequency between 4 kHz and 22 kHz). The payload in these experiments equals 100 bps. The experimental conditions are the same as in Table 4.4. As is seen, even for low pass filtering (cut-off frequency greater than 6 kHz), the bit error rate remains smaller than 10 %. For cropping attack, (0.250 percent of samples are randomly put to zero), the bit error rate is small. The strongest attack is time rescaling. With a factor of 1.10, we still have more than 10 percent of bit error rate.

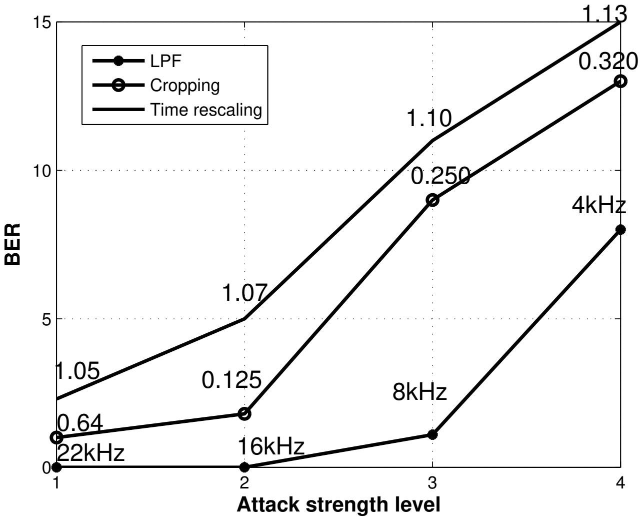

Figure 4.12 The average BER of the decoder under different attack strengths.

Note that our approach does not embed watermark into the 5 most high frequency channels leading to a robust audio watermarking method against low pass filtering.

Moreover, there is a trade-off between the quality of the signal in (4.1) and the BER of the decoder. In Fig.4.13 and Fig. 4.14, respectively, the average (ODG) of the watermarked signals and BER of the decoder are plotted versus the number of gammatone channels and the density of the coefficients (when the payload is 100 bps, and there is no attacks). Results are obtained for 5 hours of different music genres and English speech (100 signals, 3 minutes each). Increasing the number of gammatone channels and density means using more coefficients in the sparse representation. Hence, sparsity imposes a trade-off between quality and bit error rate of the decoder. Using more coefficients for the sparse representation results in more average ODG in Fig. 4.13 for the watermarked signals and at the same time, it results in more average BER in Fig.4.14.

When density increases (sparsity is reduced), the BER and ODG becomes closer. This is because, for a greater density, we use more PMP iterations. New gammatones found with the last iterations have smaller coefficients and therefore have less impact on the quality and the BER of the decoder.

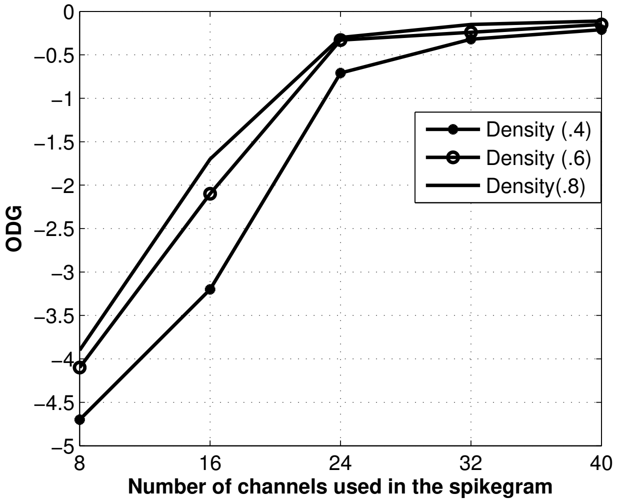

Figure 4.13 Average objective difference grade versus the number of gammatone channels in the spikegram and density (density=1-sparsity).

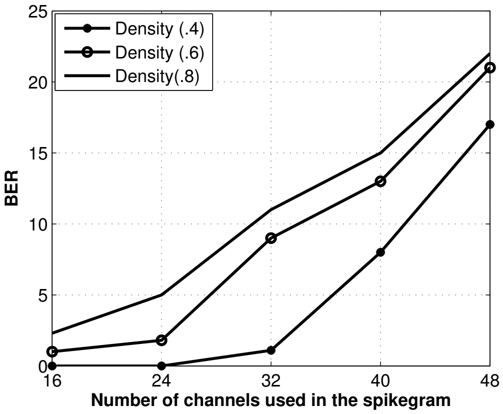

Figure 4.14 Average BER of the decoder versus the number of gammatone channels in the spikegram and density.

4.7.3 Robustness of the proposed method against a new generation codec, USAC

In this section, the robustness of the proposed method is evaluated against a new generation codec called unified speech and audio coding (USAC) [Neuendorf *et al.*, 2009]. USAC applies linear prediction in time domain (LPD) along with residual coding for speech signal segments and frequency domain (FD) algorithms for music segments. Also it is able to switch between the two modes dynamically in a signal-responsive manner. In Fig.4.15, the bit error rate results of the proposed decoder under the USAC attack are plotted. As is seen, for channels 2,4 and 8, the bit error rate is smaller than .03. As the processing window length for the USAC experiments is 200 msec, hence 5 bits per channel is embedded in each second. Thus the robust payload against USAC is between 5 bps-15 bps.

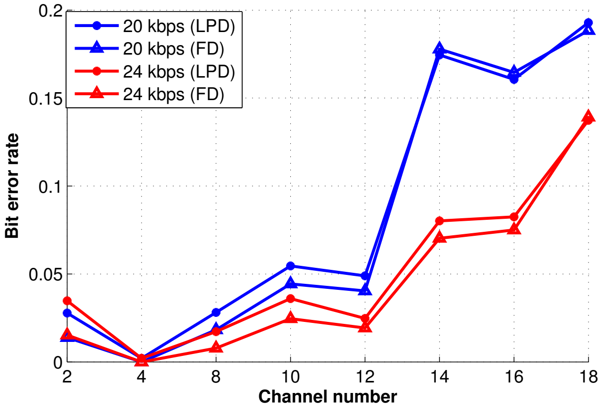

Figure 4.15 The BER of the proposed decoder under the unified speech and audio coding (USAC) [Neuendorf *et al.*, 2009] for different bitrates (24 kps and 20kps) and different modes (linear prediction (LPD) and Fourier domain (FD)). The horizontal axis indicates the embedding channel. As is seen, only for embedding channels 2, 4 and 8, the BER is smaller than .02. The processing window length is 200 msec. Also, the BER for LPD mode is slightly larger than the BER for FD mode. The experiments are done on 100 audio signals, including different music genres and English voices, 3 minutes each.

4.7.4 Real-time watermark decoding using the proposed method

The computational complexity of the proposed scheme was analysed on a personal computer with an Intel CPU at a frequency of 2.5 GHz and DDR memory of 512 MB using a MATLAB 7 compiler. The decoding procedure includes computing projections and finding a maximum value between several projections. Our experiments show that the required time for the decoding of one second of the watermarked signal is 780 msec. Also the preprocessing time that includes creating the gammacosine and gammasine kernels, the pseudo noise, is around 2.3 second. This indicates that, after the initial preprocessing stage, the proposed method can be used for real-time decoding of the watermark bits.

4.7.5 Comparison to recent audio watermarking techniques

Table 4.5 compares the proposed method for robust audio watermarking and several recent methods in terms of robustness against 32 kbps MP3 attacks. As is seen, the proposed method has a greater robust payload against 32 kbps MP3 compression compared to the mentioned recent methods. In the proposed method, PMP removes the coefficients associated with inaudible content of the signal which are under the masking thresholds and the watermark bits are inserted into high value coefficients. Therefore, this helps having more robustness against MP3 attack in which the perceptual masking is also used. Note that, the conditions of attacks in the caption of Table 4.5 are comparable to the conditions described in these references. Also, to the author's knowledge, this is the first report on the robustness of an audio watermarking system against next generation codec USAC. A bit error rate smaller than 5% is achieved with an averaged payload comprised between 5 to 15 bps.

4.7.6 Discussion: prior works based on non-sparse decomposition and perceptual representation of signals

There are several methods which might have similarities to the proposed approach. In [Coumou and Sharma, 2008], a speech watermarking method is proposed that uses pitch modifications and quantization index modulation (QIM) for watermark embedding and is robust against de-synchronizaion attacks. Although [Coumou and Sharma, 2008] is robust against low bit rates speech codecs such as AMR codec, no payload results are given for audio signals. In [Khaldi and Boudraa, 2013], after empirical mode decomposition of the audio signals, the watermarking embedding is done on the extrema of last IMF (intrinsic mode function) using QIM. Table 4.5 confirms that our approach outperforms this method in terms of robustness against 32 kbps Mp3 compression. In [Wu *et al.*, 2005], the watermark is inserted into the wavelet coefficients using QIM. Also, in [Kirovski and Hagai, 2003], the spread spectrum (SS) is applied on MDCT coefficients along with psychoacoustic masking for single-bit watermarking. Long duration audio frames are used along with cepstral filtering at the decoder. There are several differences between our approach and the above-mentioned transform domain methods. First, we evaluate the efficiency of a new transform, called spikegram, for robust watermarking. We introduce a new framework for audio watermarking called two-dictionary approach. The encoder and the decoder search in a correlation space to find the maximum projection (minimum signal interference). Second, the proposed approach is a phase embedding method on gammatone kernels with uses of masking. Gammatone kernels are the building blocks to represent the audio signal. Watermark bits are inserted into the kernels that are most efficient for decoding. Third, the proposed method takes care of efficient embedding into non masked

Table 4.5 Comparison to recent methods. The average results for 5 hours of different music genres and English voices have been compared to the average reported results. During the attacks, the watermarked signals are modified as follows: for the random cropping (Crop), the number of cropping per total length equals 0.125%. For re-sampling, the signals are re-sampled at 22.05 kHz. For re-quantization, the signals are re-quantized at 8 bits. For AWGN, 20dB additive white Gaussian noise is added to the signal. For pitch and amplitude scaling, the pitch and the amplitude of the signals are scaled with the .95 and 0.5-2 scaling ratios, respectively. For LPF, signals are low pass filtered with cut-off frequency equals to 11.025 kHz. NM means "not mentioned".

Method	Payload(bps)	MP3(kbps),BER	Crop	AWGN	Resample	Requantization, 8 bits	Pitch Scaling	Amplitude Scaling	LPF
Bhat [Bhat *et al.*, 2010]	45.9	32, .00	.00	.00	.00	.00	NM	NM	.00
Khaldi [Khaldi and Boudraa, 2013]	50.3	32, 1.00	.00	.00	1.00	.00	NM	NM	.00
Yeo [Yeo and Kim, 2003]	10	96, ≈.20	NM	NM	.00	.00	NM	.00	NM
Shaoquan [Wu *et al.*, 2005]	172	96, ≈.07	.00	<3.00	.00	.00	NM	NM	NM
Zhang [Zhang *et al.*, 2012]	43.07	64, .22	NM	8.64	.63	1.39	NM	.04	.46
Nishimura [Nishimura, 2012]	100	64, .00	NM	≈1.0	.00	.00	.00	.00	.00
Our Method	≈56.5	32, .00	.00	.00	.00	.00	.00	.00	.00

coefficients which make it robust against attacks such as universal speech and audio codec (24 kbps USAC) [Neuendorf *et al.*, 2009] and 32 kbps MP3 compression. It finds the sparse high amplitude coefficients, removes inaudible gammatones which are located under other gammatones' perceptual masks. Then, in each processing window, the proposed method adaptively finds the greater coefficient. Also, thanks to the PMP sparse representation, many coefficients are removed from the representation which fall under masks. Hence, again the signal interference is reduced at the decoder.

4.7.7 Robustness against analogue hole experiments

Although it is not the main goal of the proposed watermarking system to be robust to the analogue hole, its robustness is evaluated in a preliminary experiment. In Fig. 4.16, the bit error rate of the proposed method against a simulated real room are given using the image source method for modeling the room impulse response (RIR) [All, 1979], [E. Lehmann and Nordholm, 2007]. We embed one bit of watermark in each second of the host signal (1 bps payload). We use an open source MATLAB code [Lehman, 2016], [E. Lehmann and Nordholm, 2007] to simulate the room impulse responses. A cascade of RIR of a $4m \times 4m \times 4m$ room with a 20 dB additive white Gaussian noise is considered as the simulated room impulse response. Also, only one microphone and loud speaker are modeled. The experiments are done for three distances d between the loudspeaker and the microphone including $d = 1, 2$ and 3 meters (d denotes the distance between the microphone and the speaker). For watermark embedding, all the bits in each 1-second frames are generated using a pseudo random number generator. A spread spectrum (SS) correlation decoder is used. Hence, the 1-second sliding window is shifted sample by sample until the correlation of the SS decoder is above 0.75. Then, the watermark bit is decoded as the sign of the SS correlation. Results are reported in Fig.4.16. From Fig.4.16, the decoder can be robust against the analogue hole, when $d = 1$ meter, with a BER lower than 5 %. While for $d = 2$ or 3 meters, the BER increases sharply. The experiments are done on the 5 signals presented in Table 4.3.

4.7.8 Robustness against time rescaling attack

In the context of 1-bit watermark decoding, we first compute the correlation between the watermarked signal with a sliding gammasine or gammacosine candidate for the given channel j and different time samples $k = 1, 2, .., N_P$ where N_P is the number of time samples in the processing window. Then, the decoded watermark bit is the sign of the peak correlation, i.e, $g_{opt} = argmax_{g_{j,k}}(|< y[\alpha n], g_{j,k} >|)$, $\hat{b} = sign(< y[\alpha n], g_{opt} >)$, where

$$< y[\alpha n], g_{j,k}[n] >=< y[n], g_{j,k}[\frac{n}{\alpha}] >, \quad k = 1, 2, .., N_P \tag{4.11}$$

When α is close to one, the position of peaks in (4.11) do not change compared to the no-attack situation. This results in robustness against mild time rescaling for single bit watermarking. In multibit watermarking, the watermark gammatone is inserted in odd channel numbers. Thus, when α is close to one, odd channel numbers have weak correlations. This means that time rescaling, with small rescaling factor, do not affect the decoder of the multi-bit watermarking. Figure 4.12 reports the bit error rate for α between 1.05 and 1.13.

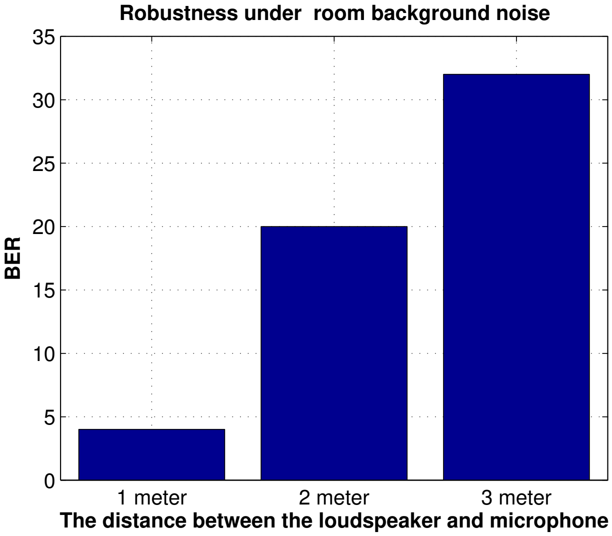

Figure 4.16 The bit error rate of the proposed method against a simulated analogue hole in combination with a 20 dB additive noise.

4.8 Conclusion

A new technique based on a spikegram representation of the acoustical signal and on the use of two dictionaries was proposed. Gammatone kernels along with perceptual matching pursuit are used for spikegram representation. To achieve the highest robustness, the encoder selects the best kernels that will provide the maximum strength factors at the decoder and embeds the watermark bits into the phase of the found kernels. Results show better performance of the proposed method against 32 kbps MP3 compression with a robust payload of 56.5 bps compared to several recent techniques. Furthermore, for the first time, we report robustness result against USAC (unified speech and audio coding) which uses a new standard for speech and audio coding. It is observed that the bit error rate is still smaller than 5% for a payload comprised between 5 and 15 bps. The approach is versatile for a large range of applications thanks to the adaptive nature of the algorithm (adaptive perceptive masking and adaptive selection of the kernels) and to the combination with well established algorithms coming from the watermarking community. It has fair performance when compared with the state of the art. The research in this area is still in its infancy (spikegrams for watermarking) and there is plenty of room for improvements in future works. Moreover, we showed that the approach can be used for real-time watermark decoding thanks to the use of a projection-correlation based decoder. In addition, two-dictionary method could be investigated for image watermarking.

Acknowledgment

This work was initially funded by CRC and then by NSERC and Université de Sherbrooke. Many thanks to Hossein Najaf-Zadeh for the PMP code, to Malcolm Slaney for the auditory modeling toolbox, to Frédérick Lavoie and Roch Lefebvre for managing the listening test, to Philippe Gournay for providing the USAC code and to the participants of the listening test. Thanks to the reviewers for their constructive comments that improved the paper.

CHAPTER 5

PERCEPTUAL ATTACKS IN SPARSE DOMAIN

5.1 Introduction

In robust audio watermarking (AW), a watermark bit stream is transformed into a hidden noise. This hidden noise is then inserted into the signal and should remain intact even in the presence of moderate to strong attacks such as re-sampling, re-quantization, MP3 and de-synchronization. Therefore, proposing strong attacks that challenge the irremovability of watermarks may introduce essential preconditions for robust AW design. In this chapter, we establish three perceptual attacks that can be used by the watermarking community to assess the performance of a robust watermarking algorithm. We define the perceptual attack as an attack which aims at removing the watermark by corrupting inaudible regions in the spectro-temporal representation of the signal. For instance, MP3 compression is a perceptual attack. However, spikegram using PMP [Pichevar *et al.*, 2011], [Najaf-Zadeh *et al.*, 2008], while preserving the quality, might highly manipulate the inaudible content of the signal, even more than MP3. Hence, they deserve to be analysed as possible attacks. In this chapter, first we present the spikegram representation that we use to design perceptual attacks. Then the spread spectrum (SS) [Malvar and Florencio, 2003] watermarking is explained as one of the basic watermarking systems. Afterwards, three perceptual attacks based on the spikegram are presented to attack the spread spectrum watermarking.
These three perceptual attacks are PMP attack, inaudible noise adding and the sparse replacement attack. For measuring the efficiency of these attacks, they are implemented against the spread spectrum AW. It is shown that under the sparse replacement attack, the spread spectrum decoder is degraded, with a greater factor than when attacking with 32 kbps MP3 and 24 kbps USAC (unified speech and audio coding [Neuendorf *et al.*, 2009]). Hence, the proposed sparse replacement attack can be considered as a strong attack on AW systems.

5.2 Using spikegram to attack spread Spectrum watermarking

Throughout this chapter, we evaluate the efficiency of the spikegram as a perceptual sparse attack compared to 32 kbps MP3 compression which is a well known state of the art perceptual attack, and 24 kbps unified speech and audio coding (USAC). The spikegram is found for audio signals using PMP along with gammatone dictionary [Smith and Lewicki, 2005].

5.3 Spread spectrum watermarking

To show the applicability of the proposed attacks, we consider the watermarking method to be additive spread spectrum (SS) with a normalized correlation detector [Malvar and Florencio, 2003]. The embedding equation in SS watermarking is as below,

$$x[n] = s[n] + \alpha b p[n], n = 1, .., L \tag{5.1}$$

where $s[n]$ and $x[n]$ are the original and the watermarked signals respectively. $b \in \{1, -1\}$ is the inserted watermark bit, α is the watermark strength factor and $p[n] \in \{-1, 1\}$ is the values of the pseudo random sequence (PN) [Klein, 2013a] obtained using a linear feedback shift register [Malvar and Florencio, 2003], [Klein, 2013a].

Equation (5.1) indicates that in SS watermarking, watermark is an additive noise and comprises a pseudo noise (PN) sequence with two-level amplitudes α and $-\alpha$. Also the correlation decoder is used. In other words, the decoded watermark bit $\tilde{b}$ is equal to the sign of the normalized correlation between the watermarked signal and the PN sequence as below

$$\tilde{b} = \frac{1}{\alpha L} sign(\sum_{i=1}^{L} x[n]p[n]) = sign(b + \frac{1}{\alpha L}\sum_{i=1}^{L} s[n]p[n]) \tag{5.2}$$

Thus, the normalized correlation equals $b \in \{1, -1\}$ plus an interference term related to the host signal $s[n]$.

The goal of sparse attacks based on spikegram is to modify the audio signal, without changing its quality, in order to increase the error rate of the watermark decoder. For SS watermarking, the goal is to reduce the normalized correlation from one to zero (when $b = 1$). For stronger attacks, the normalized correlation of the SS decoder is reduced more.

5.4 The proposed perceptual attacks

5.4.1 The spikegram attack using the PMP

First, through the spikegram representation of the watermarked signal, the sparse PMP coefficients and their associated masking thresholds are obtained. Those coefficients which are smaller than 0.1 percent of the windowed signal norm and fall below the masking thresholds are ignored and the attacked signal is reconstructed from remaining coefficients. As mentioned in chapter 2, in the spikegram found by PMP representation, a masking threshold is computed for each gammatone. Each gammatone kernel with sensation level below its associated masking theshold is removed from the spikegram. Hence, the spikegram representation using PMP modifies the frequency content of the signal. Usually in watermarking systems, watermark bits are inserted into low frequency content of the signal. By spikegram representation, the watermarking bits which are hidden in the low frequency spikes might change. Hence, spikegram representation can interfere with the decoding of watermarking bits while preserving the audio quality.

5.4.2 The inaudible noise attack using the spikegram

In inaudible noise attack, the goal is to shape an inaudible noise to be added to the signal. As this additional noise modifies the spectro-temporal content of the signal, the watermark is expected to be modified under this attack.
The steps of this attack are as follows,

1. A spikegram representation of the watermarked signal is obtained using PMP representation and masking thresholds of the kernels are determined. Hence the masking obtained for time-channel i, j is $m_{i,j}$.
2. Using a linear feedback shift register [Klein, 2013a], a pseudo random sequence (PN) is created including only -1 and +1, with the length equal to the number of gammatone kernels. Hence for the time-channel i, j, the corresponding PN value is $p_{i,j}$.
3. Each coefficient (point) in the spikegram is modified using the PN sequence and the masking thresholds obtained by PMP as below,

$$\tilde{c}_{i,j} = c_{i,j} + p_{i,j} m_{i,j} \tag{5.3}$$

where $c_{i,j}$ and $\tilde{c}_{i,j}$ are the original and modified coefficients associated to the time sample i and the channel number j.

4. A new spikegram is reconstructed using the modified coefficients $\tilde{c}_{i,j}$ and the attacked watermarked signal is re-synthesized from the modified spikegram.

The efficiency of the inaudible noise shaping attack comes from (5.3). The right term in (5.3) is the masking threshold associated to the time-channel point i, j in the spikegram with the amplitude one or negative one. When re-synthesizing the signal from the modified spikegram coefficients, we will have two coefficients for each time-channel sample i, j, one is $c_{i,j}$ for the original gammatone and another is $p_{i,j}m_{i,j}$ for the masked gammatone. Hence, we expect the quality of the re-synthesized watermarked signal using the modified spikegram to be similar to the quality of the original watermarked signal.

5.4.3 The sparse replacement attack using the spikegram

Audio signals include frames in the time domain that either their perception is similar to average ear or their time domain waveforms are similar. For example, for the music signals similar notes are played several times. These notes might have very close time domain waveforms or evoke the same perception for the average ear. As another example, in a speech signal (spoken by one person) in many time frames, perceptually similar vowels and fricatives are available. This brings upon the idea of replacement attack where different watermark bits are placed in similar contents of the audio signal. The watermarking is inserted usually as an additional noise into the signal, hence this noise might change slightly the perceptually similar frames of the audio signal. In this case, by exchanging the approximate perceptually similar contents of the signal together, watermark bits associated to these contents will be mis-detected at the decoder while the quality does not change too much [Kirovski and Petitcolas, 2007].

Here, we propose a replacement attack using the spikegram representation of audio signals. This attack is based on the replacement of perceptually similar features (time spikes) of the signal obtained using the spikegram.

5.4.4 A new replacement signal manipulation for audio signals

In this section, we define a new perceptual signal manipulation based on replacement of signal contents. Then we describe the sparse replacement attack based on that. The steps of the replacement signal manipulation are as below,

1. First, the spikegram coefficients and their associated masking thresholds are obtained by representing the signal on gammatone filter bank using PMP. Then the coefficients under their associated masking thresholds are suppressed and a 2-D time-channel spikegram represents the remaining coefficients.
2. Second, each coefficient in the spikegram is multiplied to its associated gammatone kernel. The resulted wave is called a spike (See Figure 5.1).
3. Then, at each time sample, we add together all the spikes across the vertical axis (channel axis) and name the resulted wave, a time spike.
4. Afterwards, we make a dataset of all time spikes of the input signal. As for each time sample there is one time spike, the number of time spikes equals as many as the input signal's samples. We start from the beginning sample of the signal, find its associated time spike, consider it as the current time spike. We find the K time spikes in the dataset of time spikes which have a short time distance to the current time spike and are perceptually similar to it (the current time spike and these K time spikes should have roughly similar energies and shapes). Then, we replace the current time spike with the average of the K found time spikes. We continue this procedure for the next sample of the input signal.
5. Finally, the resynthesized attacked watermarked signal is calculated by adding up all time spikes in the modified spikegram.

As can be seen in Figure 5.1, the time spikes TS_1, TS_5 and TS_7 have similar shapes and amplitude. Also the time spikes TS_3 and TS_8 are similar. As the watermarked signal is the summation of all these time spikes, if we replace TS_1 with TS_5 or TS_7 or a combination of the two and TS_3 with TS_8, the quality change of the audio signal might not be perceptible. However, if there are different watermark bits in different time spikes, the watermark bits will be mis-detected because of the de-synchronization.

If we define the l^2 norm of the vector $\boldsymbol{x} = \{x_1, x_2, ..., x_M\}$ as, $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^{M} x_i^2}$, then the similarity $d_{i,j}$ between time spike TS_i and TS_j is defined, between two time spikes as below

$$d_{i,j} = \frac{< TS_i, TS_j >}{\Big(max(\|TS_i\|, \|TS_j\|)\Big)^2} \tag{5.4}$$

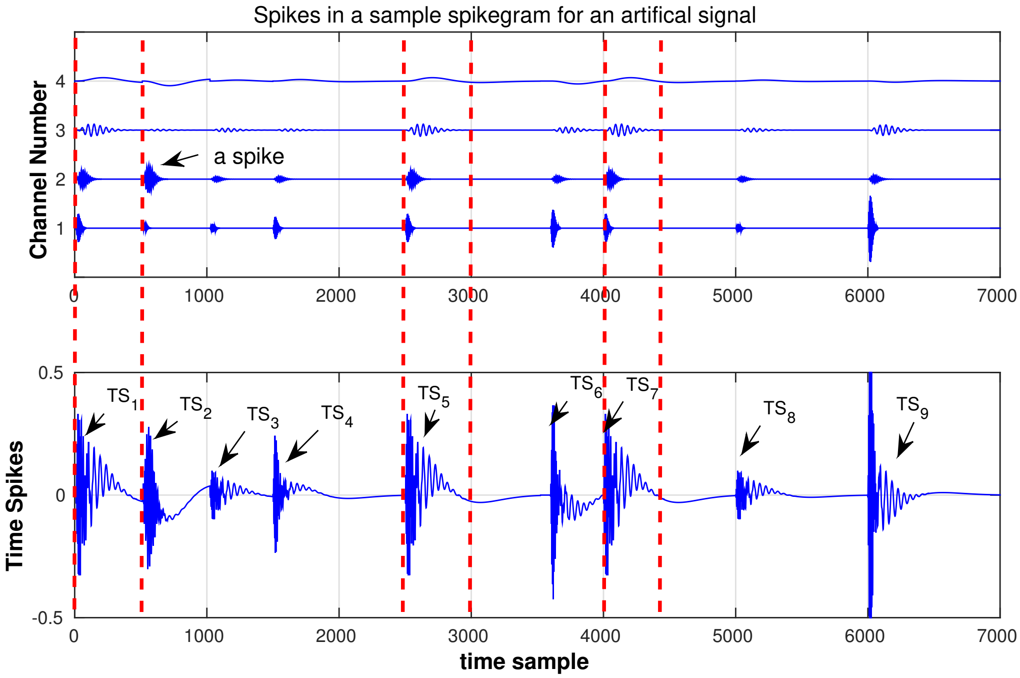

Figure 5.1 In sparse replacement attack, an audio signal is represented by spikegram using the perceptual matching pursuit. Then masked coefficients and gammatones are removed from the representation. Each gammatone in a specific time-channel point is multiplied to its associated coefficient obtained by PMP, and it is called a spike (in the time-channel plane, the upper figure). A time spike (time spikes are shown in the lower part of Figure 5.1) is computed at each time sample as the summation of all spikes along the channel axis on that time sample. When time-spikes have roughly similar shapes, we merge them and this is the case for TS_1, TS_5 and TS_7.

where $< TS_i, TS_j >$ indicates the dot product between the two time spikes. The normalization term in the denominator of (5.4) helps having the same similarity for different pairs of time spikes, with similar waveforms, but with different energies (amplitudes).

Note that the similarity cost function d_{ij} is not computed for the time spikes with zero norms. If we consider $< TS_i, TS_j >= \|TS_i\|\|TS_j\|cos(\theta_{ij})$, where $\theta_{i,j}$ is the angle between the two time spike vectors TS_i and TSj, then (5.4) can be rewritten as either $d_{ij} = \frac{\|TS_i\|}{\|TS_j\|}cos(\theta_{ij})$ or $d_{ij} = \frac{\|TS_j\|}{\|TS_i\|}cos(\theta_{ij})$. Hence, the similarity cost function $d_{i,j}$ is a value between -1 and 1, since all the time in (5.4), the module of the denominator is higher than the module of the numerator. For two similar time spikes, the angle θ_{ij} is close to zero and their norms are roughly equal, hence the similarity function d_{ij} is approximately one.

In Figure 5.2, two time spikes from the SS watermarked POP signal "The power of love" are plotted. The two time spikes include different watermark bits. Here, the similarity cost function d_{ij} is equal to 0.97.

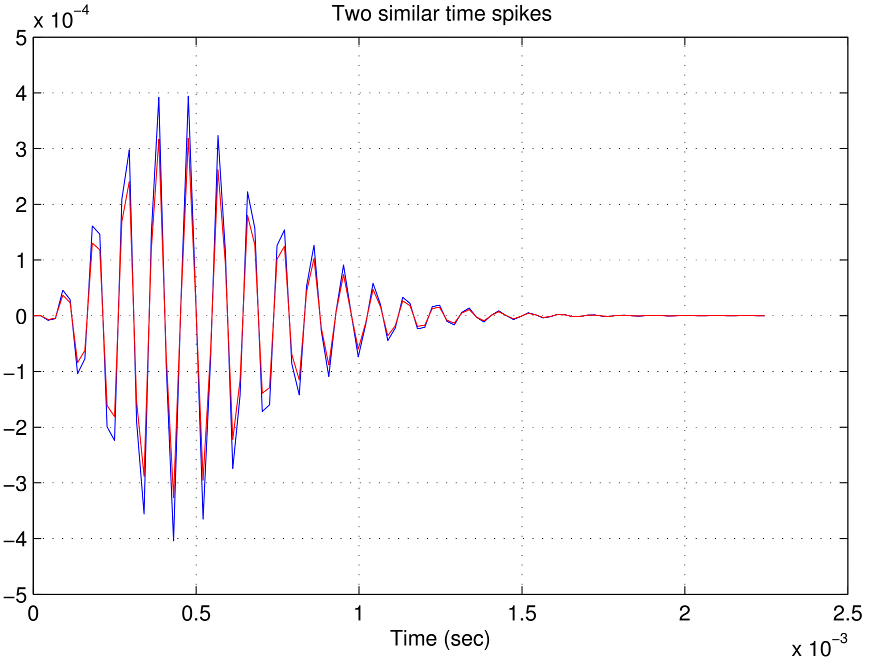

Figure 5.2 The comparison between two roughly similar time spikes from the audio signal “The power of love”, sampled at 44.1 kHz. The similarity cost function is equal to .97.

In Figure 5.3, for each time spike in the time axis, the number of similar time spikes in one second of the watermarked signal is plotted. The test signal used in this experiment is the watermarked POP audio file “The power of love”, sampled at 44.1 kHz. As is seen, for each time spike, there is on average 5.64 similar time spikes per second. Therefore, in sparse replacement attack for each second of this audio signal, each time spike can be replaced with 5.6 similar time spikes (or their averages).

5.5 The dataset and description of experiments

The dataset includes 5 raw audio signals presented in Table 5.1, each quantized at 16 bit and sampled at 44.1 kHz (Classic, Jazz, Vocal, POP, Blues), with a duration of 5 minutes.

The processing window duration for the SS watermarking is 22.7 ms, meaning that the parameters L is $22.7 * 44100 \approx 1000$ samples. The SS amplitude α equals 0.1 percent of the windowed signal norm. To generate the 2-D time-channel plane for the spikegram, 25 channel gammatone filters are sampled at each one time sample. Also, PMP is run on the $4-5$ second segments of the audio signals in Table 5.1, for the number of iterations as

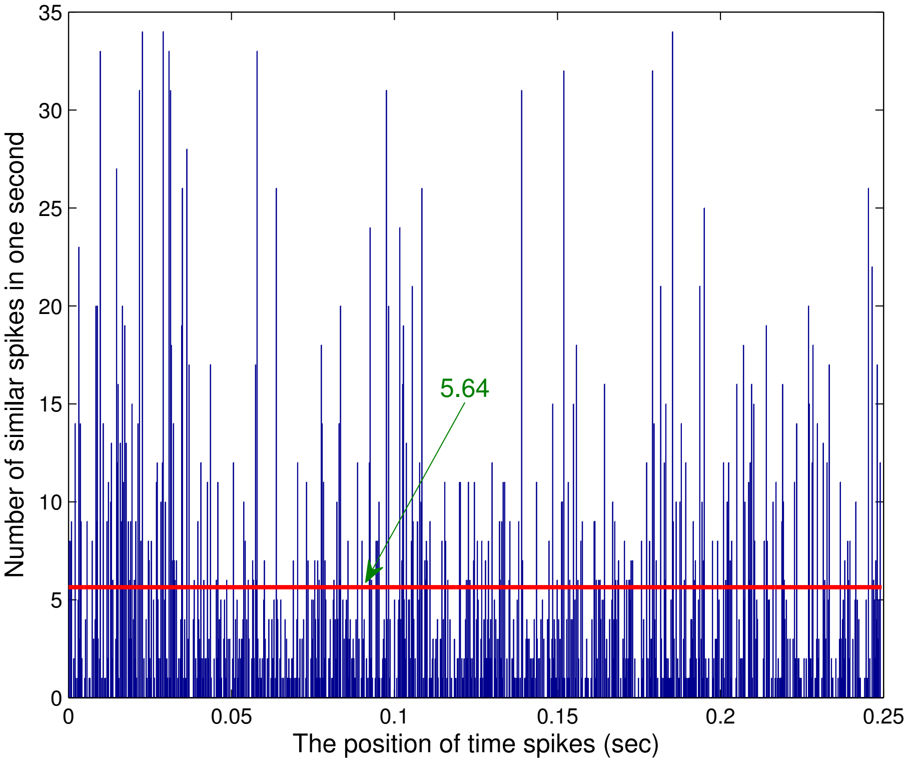

Figure 5.3 For each time spike (on the horizontal axis) inside the first 0.25 second of the watermarked audio signal “The power of love”, the number of similar time spikes is shown on the vertical axis. For a given time spike, only similar time spikes located at the following one second time frame are measured. To be considered as perceptually similar, two time spikes should have a similarity cost function d_{ij} between 0.95 and .99. As is seen, for each time spike, there are on average $5-6$ similar time spikes per second. The red line indicates the average number of similar time spikes for spikes in the first 0.25 second of the signal.

many as 20% of the segment length.

We consider two time spikes as similar if their similarity cost function is between .95 and .99 (these values are found empirically, so that the attack is effective and the quality of the signals in the dataset is not affected too much). Moreover, we replace the current time spike with similar time spikes located at most, 0.5 second prior or after the current time spike.

Table 5.1 The audio files chosen for the perceptual sparse attack test

Audio Type	Title (Author or Group Name)
POP	Power of love (Celine Dion)
Classical	Symphony No. 5 (Beethoven)
Jazz	We've got (A tribe Called Quest)
Blues	Bent Rules (Kiosk)
Speech	French Voice (A Female)

5.6 Results

The three mentioned attacks are performed on the audio signals in Table 5.1. In all attacks, a 25 channel gammatone filter bank is repeated at each time sample to generate a 2D dictionary for the PMP representation. For the PMP noise adding attack, the number of taps for the linear feed back shift register equals the $log_2 N$, where N is the number of signal's samples. The MATLAB software was used for implementing the attacks. For the SS correlation decoder, the correlation PN sequence and the watermarked signal is computed where we consider the exact synchronization between the watermarked signal and the PN sequence is assumed. The original signals and the attacked ones can be found at the link below,
http://www.gel.usherbrooke.ca/necotis/necotis-old/yerfani.html
In Figure 5.4, the average results are shown for the three proposed perceptual attacks in the spikegram domain compared to the 32 kbps MP3 attack and 24 kbps USAC coding (FD mode and LPD modes) on the same test signals mentioned in Table 5.1). As is seen in Figure 5.4, the normalized decoder correlation for the SS decoder is around one. This means that the signal interference at the decoder of SS in (5.2) is very small since the length of the windowed signal (L) for the SS watermarking is sufficiently high. Figure 5.4 shows that the PMP attack efficiency is roughly similar to the efficiency of the MP3 attack as in our experiments the average normalized correlation decoder for both attacks are reduced around 0.17-0.22. For the PMP noise adding attack, the average correlation at the decoder even reduces around 0.13 which shows the efficiency of perceptual noise adding attack is in the range of MP3 attack.

Moreover, the efficiency of 24 kbps USAC coding as an attack in both FD and LPD modes, is roughly between the MP3 attack and the replacement attack. Also, USAC in LPD mode is stronger in attacking the spread spectrum decoder than in its FD mode.
It is clear from Figure 5.4, that the sparse replacement attack reduces the average correlation of the SS decoder to around 0.83. This means that sparse replacement attack is a very strong attack compared to other attacks mentioned in this chapter. One reason is that in

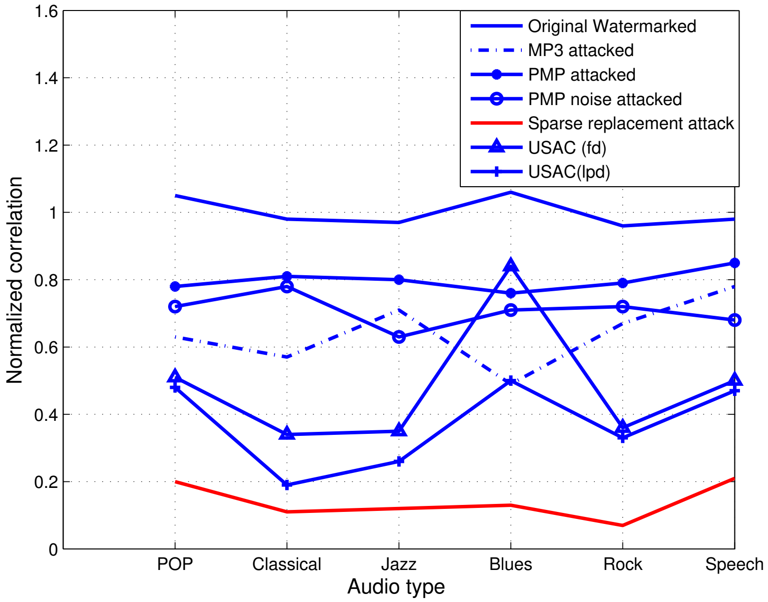

Figure 5.4 The comparison between the average correlation amount of the SS watermarking decoder for different perceptual attack situations and different audio types. Experiments are done on 5- minute long audio signals chosen from Table 5.1. The MP3 compression has the 32 kbps bit rate. The parameter L for SS watermarking equals 1000 (for a 23 msec processing window). The parameter α is equal to 0.1 percent of the windowed signal norm. Having smaller normalized correlation values for the SS decoder, means the attack is stronger. As is seen, the 32 kbps MP3 compression, PMP and inaudible noise attacks have roughly the same strength. As a perceptual attack, 24 kbps USAC in LPD mode is stronger than in FD mode (both modes were run for all signals), while in both modes USAC is on average stronger than MP3 attack. Moreover, the efficiency of sparse replacement attack is much greater than other attacks.

the replacement attack, we have already considered the effect of the PMP attack. Moreover, by replacing the time spikes, we are benefiting from the strong de-synchronization attack.

5.6.1 Robustness of the two-dictionary method against the perceptual attacks mentioned in this chapter

Here, we measure the robustness (in terms of BER) of the two-dictionary method (TDA) mentioned in chapter 4 against the three proposed perceptual attacks. In all attacks, the same experimental setup is used. The same gammatone dictionary is used, the sampling frequency is 44.1 kHz, and the input signals are given in Table 4.3. For more on experimental

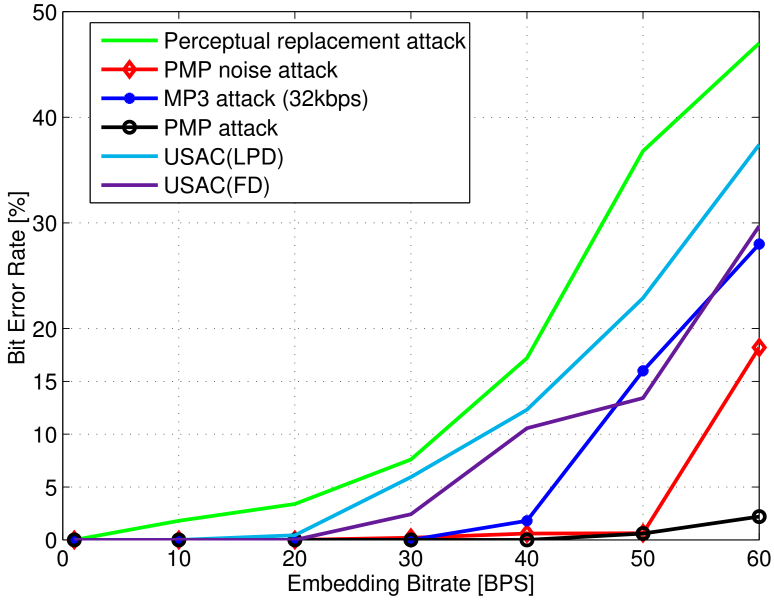

Figure 5.5 The average bit error rate of the TDA decoder under the following attacks: sparse replacement attack (green line), PMP noise attack (red line), 32 kbps MP3 attack (blue line), the PMP attack (black line), 24 kbps USAC FD mode (violet line) and 24 kbps USAC LPD mode (light blue line). The signal used for performing attacks are given in Table 4.3. For the TDA method, 25 gammatone kernels are repeated each time sampel to generate the spikegram with the sampling rate of 44100 Hz and the number of iterations in PMP equals 20 % of the number of signal's samples.

setup, see the caption of Fig. 5.5. The results of perceptual attacks on the two-dictionary method is given in Figure 5.5. In Figure 5.5, for a better comparison, the robustness of two-dictionary method is also shown against 32 kbps MP3 attack and 24 kbps USAC (FD and LPD modes). As is seen in Figure 5.5, still the sparse replacement attack is the strongest compared to all other mentioned attacks. Also, PMP causes a very low bit error rate at the decoder of TDA. This is reasonable, as in the encoder of TDA, the signals are already re-synthesized by PMP, thus they are robust against PMP attack. Also, the decoder of TDA has roughly the same bit error rate against 32 kbps MP3 attack and PMP noise attack. The efficiency of 24 kbps USAC for attacking the two-dictionary method (results are shown in Figure 5.5) is lower than when attacking spread spectrum watermarking (results are shown in Figure 5.4). One reason is that, in SS watermarking, the watermark is a spread spectrum, wide band noise which is spread on the whole spectrum of the signal. Hence, the spectrum of the signal in SS watermarking can be extensively modified by the 24 kbps USAC encoder. In the proposed TDA method, the watermark bits are inserted only in the phase (sign) of high value coefficients and low frequency kernels which are more robust against transforms and attacks.

For the TDA experiments in this chapter, the simulation conditions are given in Table 4.2.

5.7 Summary

In this chapter, three perceptual attacks in the sparse domain were presented. These attacks were tested for a simple spread spectrum watermarking system. However without loss of generality they can be performed on all other watermarking methods. We showed that these attacks can outperform the 32 kbps MP3 attack. Based on the experiments of this chapter, perceptual sparse replacement manipulation is a very strong attack compared to 32 kbps MP3 compression. Also, we showed that 24 kbps USAC, as an attack, is stronger than 32 kbps MP3 attack (the strength of 24 kbps USAC is greater in the LPD than in FD mode) but still weaker than the proposed sparse replacement attack. Also, we showed that the proposed TDA method has more than 50-60 bps robust payload (where the bit error rate of the decoder is smaller than 5%) under the perceptual attacks such as PMP attack, PMP noise attack. Also TDA has a robust payload in the range of 5-15 bps with 24 kbps USAC coding and lower than 5 bps under the sparse replacement attack.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, three applications of spikegram, for the copyright protection of audio signals are proposed and developed. For the spikegram, the bio-inspired gammatone dictionary is used and perceptual matching pursuit (PMP) is applied as the optimization algorithm. The PMP is chosen to generate sparse representation in the spikegram, since it generates the masking thresholds for the gammatone kernels used in this research. Moreover, the efficiency of PMP for signal representation is confirmed in the literature [Pichevar *et al.*, 2010a; Najaf-Zadeh *et al.*, 2008].

6.1 Conclusion

First, a novel, blind, perceptual sparse-domain audio authentication method was presented using a proposed modified spread spectrum (MSS) watermarking. Using the time-shift-invariant spikegram based on PMP, our method inserts a semi-fragile watermark stream inside audio frames. We showed that the watermark is robust against ordinary signal processing modifications such as low-pass filtering while it considers the maliciously attacked segments (such as removed segments) of the signal as tampered. The malicious attacks in our experiments include the time rescaling, the frame replacement and the de-synchronization attacks. We also showed that compared to state of the art, our method does efficiently localize the maliciously attacked frames of the signal (e.g., removed, replaced, added and time-shifted frames) with a segment size smaller than 250 msec. Our listening test confirms the high quality of the watermarked signals where the mean opinion test (MOS) results, for the proposed method, are above 4.5. Our results confirm the suitability of our method for authentication applications such as audio forensics. Moreover, the MISS method presented in chapter 3, can be used as a general embedding method for watermarking and stenography.

Second, a new technique namely two-dictionary method, was proposed for audio watermarking applications. The proposed method uses the spikegram model of audio signals using gammatone kernels. It finds the appropriate watermark kernels among gammatone filters in the spikegram based on the decoding strength of the input watermark bit and embeds the watermark bits into the phase of found gammatones. It is shown that the TDA is error free in the case of no-attack situation. Moreover, in comparison to Improved Spread Spectrum watermarking (ISS), TDA does not introduce additional distortion to the encoder. It is shown that the uncorrelatedness of watermark bases helps designing a very robust audio watermarking method. Our experiments show high robustness against 32 kbps MP3 compression with a robust payload of 56.5 bps compared to several recent techniques. The proposed method has robustness against the new generation codec USAC (unified audio and speech coding) with a payload of (5 bps-15 bps). Robustness against USAC makes the proposed method suitable for copyright protection applications on cellphone devices.

Finally, three perceptual attacks were presented using the spikegram including the spikegram attack, the inaudible noise attack and the sparse replacement attack. In the spikegram attack, the audio signal is represented and reconstructed using the spikegram. In the inaudible noise attack, a pseudo random noise is generated using a linear feedback shift register (LFSR), shaped using the masking thresholds of PMP and added to the signal. In the sparse replacement attack, the perceptually similar time spikes are found in the spikegram domain and replaced together. These attacks were tested on a simple spread spectrum watermarking system. However without loss of generality they can be performed on all other watermarking methods. We showed that the PMP and the inaudible noise attack can be as strong as 32 kbps MP3 attack. While the perceptual sparse replacement attack is stronger than 32 kbps MP3 compression.

Also, two recommendations are proposed to make watermarking systems robust against sparse replacement attack. The first recommendation is to shape the watermark noise in the perceptual sparse (PMP) domain prior to watermark insertion. Thus when perceptually inaudible content of the signal is removed under the proposed attacks, the watermarks are not affected. The second recommendation is to insert the same watermark bits, into the perceptually similar time spikes of the signal so as to gain robustness against replacement attack.

6.2 Future work

In this section, we give perspectives for the future work as below,

— There is a need for real time sparse representation. One suggestion is to increase the speed of perceptual matching pursuit. One approach can be the investigation of parallel computing architectures on the input signals using GPU.

— The audio tamper localization approach, in chapter 3, can be further developed to be used for image tampering detection. In this case, the MISS method can be used but still there is a need to discover the characteristics of the vision system (e.g characteristics similar to the spectro-temporal masking of the auditory system) to have a high quality watermark embedding.

— Reversible watermarking can not be done in real time using TDA and can be further explored as a future work. In reversible watermarking in addition to the watermark bit stream, the original signal is also obtained at the decoder.

— Designing audio watermarking with high robustness against sparse replacement attack can be explored as a future work. In this case, during the insertion of watermark bits, we should take care of similarities in the representation. Also the efficiency of sparse replacement attack can be further explored for other watermarking methods such as quantization index modulation [Vasic and Vasic, 2013] and patch work method [Xiang *et al.*, 2014a].

— We showed that, USAC as a next generation codec, can also be considered as a novel attack on audio watermarking systems. Making the proposed TDA method in chapter 4, even more robust against USAC, can be further explored as a future work. In this case, we should investigate how the USAC manipulates the spectro-temporal content of the signal and insert watermark bits into the spectro-temporal contents which are not affected by the USAC coding.

CHAPTER 7

CONCLUSIONS ET TRAVAUX FUTURS

Dans cette thèse, trois applications de spikegrammes, pour la protection des signaux sonores sont proposées et développées. Pour le spikegramme, le dictionnaire gammatone bio-inspiré est utilisé et le perceptual matching pursuit (PMP) est appliqué comme algorithme d'optimisation. PMP est utilisé pour obtenir la représentation parcimonieuse, car il génère des seuils de masquage pour les noyaux de gammatone et son efficacité pour la représentation du signal est confirmée dans la littérature.

7.1 Conclusions

Tout d'abord, un nouvelle méthode d'authentification sonore est présentée dans le domaine parcimonieux en utilisant une version modifiée du système de tatouage à spectre étendu appelé MSS (Modified Spread Spectrum). En utilisant le spikegramme basé sur PMP, notre méthode insère les bits de tatouage semi-fragile à l'intérieur des trames de signal. Nous avons montré que le tatouage est robuste contre les modifications ordinaires de traitement de signaux tels que filtrage passe-bas alors qu'il considère les segments malicieusement attaqués du signal comme falsifiés. Nous avons également montré que, par rapport à l'état de l'art, notre méthode permet de localiser efficacement les segments qui sont malicieusement attaqués (par exemple, supprimé, remplacé, ajouté, décalé). Notre test d'écoute montre la grande qualité des signaux tatouée. Nos résultats confirment la pertinence de notre méthode pour les applications d'authentification tels que VoIP. La méthode MSS présentée ici, peut être utilisée comme méthode générale pour le tatouage.

Deuxièmement, une nouvelle technique qui est appelée la méthode à deux dictionnaires (TDA, Two Dictionaries Method), a été proposée pour les applications de tatouage. La méthode proposée utilise le modèle de spikegramme des signaux sonores à l'aide filtres gammatones. Elle utilise deux dictionnaires différents qui sont sélectionnés en fonction du bit d'entrée et du contenu du signal. Elle trouve les filtres gammatones appropriés (appelés les gammatones de tatouage) sur la base de la connaissance du bit de tatouage d'entrée, et incorpore les bits de tatouage dans la phase des filtres gammatones de tatouage. Il est montré que la TDA est libre d'erreurs dans le cas d'aucune situation d'attaque. En

outre, TDA ne crée pas de distorsion supplémentaire au décodeur, parce que le signal original ne se comporte pas comme une interférence dans le décodeur. Il est démontré que la décorrélation des noyaux de tatouage permet la conception d'une méthode de tatouage sonore très robuste.
Nos expériences ont montré la meilleure performance de la méthode proposée contre 32 kbps compression MP3 avec une charge utile de 56.5 bps par rapport à plusieurs techniques récentes. Nos expériences montrent que la méthode proposée est robuste contre la compression MP3 à 32kbps avec une capacité de 56.5 bps qui est plus élevé par rapport à l'état de l'art. Aussi, La méthode proposée est robuste vis-à-vis du nouveau codec USAC (unified audio and speech coding) avec une charge utile de 5-15 bps.

Enfin, trois attaques perceptuelles ont été présentées en utilisant le spikegramme y compris l'attaque du spikegramme, l'attaque du bruit inaudible et l'attaque par remplacement parcimoneux. Ces attaques ont été testées pour une méthode de spectre étendu simple. Toutefois, sans perte de généralité, elles peuvent être effectuées sur toutes les autres méthodes de tatouage. Nous avons montré que PMP et l'attaque par bruit inaudible sont aussi forts que l'attaque 32 kbps MP3. Sur la base de ces expériences, l'attaque de remplacement parcimoneuse perceptive est une attaque très forte par rapport à la compression MP3 à 32kbps.
En outre, deux recommandations sont proposées pour rendre les systèmes de tatouage robustes contre les attaques de remplacement. La première recommandation est de façonner le tatouage dans le domaine de la perception parcimoneuse avant l'insertion. Ainsi, lorsque le contenu perceptuel inaudible du signal est éliminé sous les attaques proposées, le tatouage n'est pas affecté. La deuxième recommandation est l'insertion du même filigrane, dans les trames perceptuellement similaires du signal afin de gagner de la robustesse contre les attaques de remplacement.

7.2 Les travaux futurs

Nous présentons ci-dessous quelques perspectives pour le travail futur,

- Pour un traitement en temps réel qui utilise les spikegrammes,, une suggestion est d'augmentation la vitesse de PMP en utilisant l'idée de calcul parallèle avec GPU.
- L'approche par localisation des attaques telle que proposée dans cette thèse pourrait être appliquée aux images. Dans ce cas, la méthode MSS peut être utilisée. Toutefois pour pouvoir bien l'appliquer aux images, il est nécessaire de connaître les seuils

de masquage visuels (selon le même principe que les seuils de masquage dans le système auditif) afin de mieux insérer les bits de tatouage.

- Un travail futur serait d'amélioration de la robustesse de la méthode à deux dictionnaires contre les attaques de remplacement. Lors de l'insertion de bits de tatouage, nous devrons prendre soin de similitudes de représentation entre les différents segments du signal acoustique. En outre, l'efficacité de l'attaque de remplacement parcimonieuse peut être explorée plus pour d'autres méthodes de tatouage telle que QIM ('quantization index modulation') [Vasic and Vasic, 2013] et la méthode 'patch working' [Xiang *et al.*, 2014a].
- Nous avons montré que le 24 kbps USAC qui est un nouveau codec peut également être considéré comme une attaque. Un travail futur intéressant serait d'étudier la méthode TDA proposée pour le contexte d'attaques par CODEC USAC afin de la rendre plus robuste. Pour cela, nous devrions améliorer notre compréhension de l'impact de l'USAC sur les caractéristiques spectro-temporelles du signal pour pouvoir insérer les bits de tatouage dans le contenu spectro-temporel non affecté par le CODEC USAC.

LIST OF REFERENCES

(1979). Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, pp. 943–950.

Adler, A. and Emiya, V. (2012). Audio inpainting. *IEEE Transactions on Audio, Speech and Language Processing*, volume 20, number 3, pp. 922–932.

Arnold, M., Chen, X.-M., Baum, P., Gries, U. and Doerr, G. (2014). A phase-based audio watermarking system robust to acoustic path propagation. *IEEE Transactions on Information Forensics and Security*, volume 9, number 3, pp. 411–425.

Bensound (2015). Royality free music, http://www.bensound.com/royalty-free-music. *The page was consulted in Sept. 2015.*

Bhat, V., Sengupta, I. and Das, A. (2010). An adaptive audio watermarking based on the singular value decomposition in the wavelet domain. *Elsevier Journal on Digital Signal Processing*, volume 20, number 6, pp. 1547–1558.

Blumensath, T. and Davies, M. E. (2008). Iterative thresholding for sparse approximations. *The Journal of Fourier Analysis and Applications*, volume 14, pp. 629–654.

Boho, A. and Wallendael, G. V. (2013). End-to-end security for video distribution. *IEEE Signal Processing Magazine*, volume 30, number 2, pp. 97–107.

Bordel, G., Penagarikano, M., Rodriguez-Fuentes, L., Alvarez, A. and Varona, A. (2016). Probabilistic kernels for improved text-to-speech alignment in long audio tracks. *IEEE Signal Processing Letters*, volume 23, number 1, pp. 126–129.

Borwein, P. and Mossinghoff, M. J. (2008). Barker sequences and flat polynomials. *In book: Number Theory and Polynomials, LMS Lecture Notes, Cambridge University Press*, pp. 71–88.

Caetano, M. and Rodet, X. (2013). Musical instrument sound morphing guided by perceptually motivated features. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 21, number 8, pp. 1666–1675.

Candes, E. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, volume 51, number 12, pp. 4203 – 4215.

Chen, O. T. and Liu, C. H. (2007). Content-dependent watermarking in compressed audio with identifying manner and location of attacks. *IEEE Trans on Audio, Speech and Language Processing*, volume 15, number 5, pp. 1605–1616.

Chen, S.-T., yu Hsu, C. and Huang, H.-N. (2015). Wavelet-domain audio watermarking using optimal modification on low-frequency amplitude. *IET Signal Processing*, volume 9, number 2, pp. 166–176.

Chui, C. K. and Montefusco, L. (2014). Wavelets: Theory, algorithms, and applications. *Academic Press*, pp. 271–295.

CoolEdit (2013). Adobe edition, http://www.adobe.com. *The page was consulted in December 2014.*

Coumou, D. J. and Sharma, G. (2008). Insertion, deletion codes with feature-based embedding: A newv paradigm for watermark synchronizastion with applications to speech watermarking. *IEEE Trans. on Information Forensics and Security*, pp. 153–165.

Cox, I., Miller, M., Bloom, J., Fridrich, J. and Kalker, T. (2007). *Digital Watermarking and Steganography*, 2nd edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

D. Megas, J. Serra-Ruiz, M. F. (2010). Efficient self-synchronised blind audio watermarking system based on time domain and fft amplitude modification. *Signal Processing.*

Daudet, L. (2006). Sparse and structured decompositions of signals with the molecular matching pursuit. *IEEE Transaction on Audio, Speech and Language Processing*, volume 14, number 5, pp. 1808–1816.

Domingo, P. (2015). Digital music report. *Int. Federation of the Phonographic Industry.*

Donoho, D. L. (1995). Denoising by soft-thresholding. *IEEE Transactions on Information Theory*, volume 41, pp. 6134–627.

E. Lehmann, A. J. and Nordholm, S. (2007). Reverberation-time prediction method for room impulse responses simulated with the image-source model. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07)*, pp. 159–162.

Fallahpour, M. and Megías, D. (2010). High capacity audio watermarking using the high frequency band of the wavelet domain. *Springer: Multimedia Tools and Applications*, pp. 1–14.

Fevotte, C., Daudet, L., Godsill, S. and Torresani, B. (2006). Sparse regression with structured priors: Application to audio denoising. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 57–60.

G. Hua, J. G. and Thing, V. L. L. (2015a). Cepstral analysis for the application of echo-based audio watermark detection. *IEEE Trans. on Information Forensics and Security*, pp. 1850–1860.

G. Hua, J. G. and Thing, V. L. L. (2015b). Time-spread echo-based audio watermarking with optimized imperceptibility and robustness. *IEEE Trans. Audio, Speech, Language Process*, pp. 227–239.

Gulbis, M., Muller, E. and Steinebach, M. (2008). Content-based authentication watermarking with improved audio content feature extraction. In *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IHMSP).* pp. 620–623.

Ho, A. T. S. and Li, S. (2015). *Forensic Authentication of Digital Audio and Video Files.* Wiley-IEEE Press, pp. 704–.

Hromádka, T., DeWeese, M. R. and Zador, A. M. (2008). Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS Biol*, volume 6, number 1, p. e16.

Hua, G., Goh, J. and Thing, V. L. L. (2015). Time-spread echo-based audio watermarking with optimized imperceptibility and robustness. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, volume 23, number 2, pp. 227–239.

Hua, G., Zhang, Y., Goh, J. and Thing, V. L. L. (2016). Audio authentication by exploring the absolute-error-map of enf signals. *IEEE Transactions on Information Forensics and Security*, volume 11, number 5, pp. 1003–1016.

ITU (1996). Methods for subjective determination of transmission quality. *Recommendation P800, Publisher: International Telecommunication Union.*

ITU (1997). Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems. *Itu-r BS.1116-1, Publisher: International Telecommunication Union.*

Joel A. Tropp, A. C. G. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. on information theory*, volume 53, pp. 4655–4666,.

Kabal, P. (2002). An examination and interpretation of itu-r bs.1387: Perceptual evaluation of audio quality. *TSP Lab Technical Report, Dept. Electrical and Computer Engineering, McGill University.*

Kale, K. V., Mehrotra, S. C. and Manza, R. (2010). Computer vision and information technology. *I. K. International Pvt Ltd*, pp. 673–674.

Katzenbeisser, S. and Petitcolas, F. A. P. (2000). Information hiding techniques for steganography and digital watermarking,. *Eds. Norwell, MA, USA: Artech House.*

Khaldi, K. and Boudraa, A. O. (2013). Audio watermarking via emd. *IEEE Transactions on Audio, Speech and language Processing*, volume 21, number 3, pp. 675–680.

Kirovski, D. and Hagai, A. (2003). Audio watermark robustness to desynchronization via beat detection. *5th International Workshop on Information Hiding*, volume 2578, pp. 160–176.

Kirovski, D. and Petitcolas, F. A. (2007). Replacement attack on arbitrary watermarking systems. *IEEE Trans. on Audio, Speech, and Language Processing*, volume 15, number 6, pp. 1922–1931.

Klein, A. (2013a). *Stream Ciphers*. Springer, Berlin, Heidelberg.

Klein, A. (2013b). *Stream Ciphers*. Springer, Berlin, Heidelberg.

Knuth, D. E. (1998). Seminumerical algorithms. the art of computer programming 2. *Boston: Addison–Wesley*, p. 145–146.

Kuribayashi, M. (2014). Simplified map detector for binary fingerprinting code embedded by spread spectrum watermarking scheme. *IEEE Transactions on Information Forensics and Security*, volume 9, number 4, pp. 610–623.

Lehman, E. (2016). Fast isam, http://www.eric-lehmann.com/fast-ism-code/. *The software is used in July. 2016,.*

Lei, B., Soon, I. Y. and Tan, E. L. (2013). Robust svd-based audio watermarking scheme with differential evolution optimization. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 21, number 11, pp. 2368–2378.

M. Unoki, R. M. (2015). Robust, blindly-detectable, and semi-reversible technique of audio watermarking based on cochlear delay. *IEICE Trans. Inf. Syst.*, pp. 38–48.

Majumder1, S., Devi1, K. J. and Sarkar, S. K. (2013). Singular value decomposition and wavelet-based iris biometric watermarking. *IET: journal on Biometrics*, volume 2, number 1, pp. 21–27.

Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, volume 41, number 12, pp. 3397–3415.

Malvar, H. and Florencio, D. A. (2003). Improved spread spectrum: a new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, volume 51, number 4, pp. 898–905.

Matlab (2014). The language of technical computing, http://www.mathworks.com/. *The software is used in Dec. 2014, under the license of university of Sherbrooke.*

Najaf-Zadeh, H., Pichevar, R., Lahdili, H. and Thibault, L. (2008). Perceptual matching pursuit for audio coding. In *Audio Engineering Society Convention 124*.

Neuendorf, M., Gournay, P., Multrus, M., Lecomte, J., Bessette, B., Geige, R., Bayer, S., Fuchs, G., Hilpert, J., Rettelbach, N., salami, R., Schuller, G., Lefebvre, R. and Grill, B. (2009). Unified speech and audio coding scheme for high quality at low bitrates. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1 –4.

Ngo, N. M. and Unoki, M. (2015). Robust and reliable audio watermarking based on phase coding. *IEEE International Conference on in Acoustics, Speech and Signal Processing (ICASSP)*, pp. 345–349.

Nikolaidis, N. and Pitas, I. (2004). Benchmarking of watermarking algorithms. *in Book: Intelligent Watermarking Techniques, World Scientific Press*, pp. 315–347.

Nishimura, R. (2012). Audio watermarking using spatial masking and ambisonics. *IEEE Trans. Audio, Speech, Language Process*, pp. 2461–2469.

Parvaix, M., Krishnan, S. and Ioana, C. (2008). An audio watermarking method based on molecular matching pursuit. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*. pp. 1721 –1724.

Patterson, R., Nimmo-smith, I. and J. Holdsworth, P. R. (1988). An efficient auditory filter bank based on the gammatone function. *SVOS Final Report: The Auditory Filter Bank.*

Percybrooks, W. and Moore, E. (2015). A new HMM-based voice conversion methodology evaluated on monolingual and cross-lingual conversion tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, volume 23, number 12, pp. 2298–2310.

Pichevar, R., Najaf-Zadeh, H. and Thibault, L. (2011). New trends in biologically-inspired audio coding. *Book Chapter, INTECH: Recent Advances in Signal Processing*, pp. 37–59.

Pichevar, R., Najaf-Zadeh, H., Thibault, L. and Lahdili, H. (2010a). Auditory-inspired sparse representation of audio signals. *Elsevier Journal on Speech Communication*, volume 53, pp. 643–657.

Pichevar, R., Zadeh, H. N. and Mustiere, F. (2010b). Neural-based approach to perceptual sparse coding of audio signals. *International Joint Conference on Neural Networks*, pp. 1–8.

Pun, C. M. and Yuan, X. C. (2013). Robust segments detector for de-synchronization resilient audio watermarking. *IEEE Transactions on Audio, Speech and Language Processing*, volume 21, number 11, pp. 2412–2424.

Quackenbush, S. (2013). Mpeg unified speech and audio coding. *IEEE Multimedia*, volume 20, number 2, pp. 72–78.

Ratanasanya, S., Poomdaeng, S., Tachaphetpiboon, S. and Amornraksa, T. (2005). New psychoacoustic models for wavelet based audio watermarking. *IEEE International Symposium on Communications and Information Technology*, volume 1, pp. 602 – 605.

Razaviyayn, M., Tseng, H. W. and Luo, Z. Q. (2014). Dictionary learning for sparse representation: Complexity and algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 5247–5251.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G. and Olhausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, volume 20, number 10, pp. 2526–2563.

S. Boyd, N. Parikh, E. C. B. P. J. E. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, volume 3, pp. 1–122.

Sheikh, M. and Baraniuk, R. G. (2007). Blind error-free detection of transform-domain watermarks. In *IEEE International Conference on Image Processing (ICIP)*. volume 5. pp. 453–456.

Siahpoush, S., Erfani, Y., Rode, T., Lim, H. H., Rouat, J. and Plourde, E. (2015). Improving neural decoding in the central auditory system using bio-inspired spectro-temporal representations and a generalized bilinear model. In *37th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC)*. pp. 5146–5150.

Singh, A. K., Kumar, B., Dave, M. and Mohan, A. (2015). Robust and imperceptible spread-spectrum watermarking for telemedicine applications. *Springer*, pp. 1–7.

Siwek, S. E. (2007). The true cost of copyright idustry piracy to the us economy. *Published by Institiute for Policy Innovation (IPI).*

Slaney, M. (1998a). An efficient implementation of the patterson-holdsworth auditory filter bank. *Apple Computer Technical Report*, volume 35.

Slaney, M. (1998b). An efficient implementation of the patterson-holdsworth auditory filter bank. *Apple Computer Technical Report*, volume 35,1998.

Smith, E. and Lewicki, M. S. (2005). Efficient coding of time-relative structure using spikes. *Neural Computation*, volume 1, number 17, pp. 19–45.

Smith, E. and Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, volume 439, number 7079, pp. 978–982.

Son, C. H. and Choo, H. (2014). Watermark detection from clustered halftone dots via learned dictionary. *Elsevier Journal on Signal Processing*, volume 102, pp. 77–84.

Steinebach, M. and Dittmann, J. (2003). Watermarking-based digital audio data authentication. *Eurasip Journal on Applied Signal Processing*, pp. 1001–1015.

Strahl, S. and Mertins, A. (2009). Analysis and design of gammatone signal models. *The Journal of the Acoustical Society of America*, pp. 2379–2389.

Unoki, M. and Miyauchi, R. (2012). Detection of tampering in audio signals with inaudible watermarking technique. In *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.* pp. 118–121.

Valiollahzadeh, S. M., Nazar, M., Zadeh, M. B. and Jutten, C. (2009). A new approach in decomposition over multiple-overcomplete dictionaries with application to image inpainting. *IEEE conference on Machine Learning for Signal Processing*, pp. 1–6.

Vasic, B. and Vasic, B. (2013). Simplification resilient ldpc-coded sparse-qim watermarking for 3d-meshes. *IEEE Transactions on Multimedia*, volume 15, number 7, pp. 1532–1542.

VOA (2015). Voa english, http://learningenglish.voanews.com/. *The page was consulted in Sept. 2015.*

Voloshynovskiy, S., Pereira, S. and Pun, T. (2001). Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks. *IEEE Communications Magazine*, volume 39, number 8, pp. 118–126.

Wang, X., Wang, P., Zhang, P., Xu, S. and Yang, H. (2014). A blind audio watermarking algorithm by logarithmic quantization index modulation. *Multimedia Tools and Applications*, volume 71, number 3, pp. 1157–1177.

Wang Yong, W. S. and Jiwu, H. (2010). Audio watermarking scheme robust against desynchronization based on the dyadic wavelet transform. *EURASIP Journal on Advanced Signal Processing*, volume 2010, pp. 1110–8657.

WavePad (2013). Audio editing software, http://www.nch.com.au/wavepad/. *The page was consulted in December 2014.*

Wu, S., Huang, J., Huang, D. and Y.Q, S. (2005). Efficiently self-synchronized audio watermarking for assured audio data transmission. *IEEE Transactions on Broadcasting*, volume 51, number 1, pp. 69–76.

Wu, Z. (2015). Information hiding in speech signal for secure communication. Syngress, Oxford, pp. iii –.

Xiang, Y., Guo, S., Natgunanathan, I. and Nahavandi, S. (2014a). Patchwork-based audio watermarking method robust to de-synchronization attacks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, volume 22, number 9, pp. 1413–1423.

Xiang, Y., Natgunanathan, I., Guo, S., Zhou, W. and Nahavandi, S. (2014b). Patchwork-based audio watermarking method robust to de-synchronization attacks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, volume 22, number 9, pp. 1413–1423.

Xiang, Y., Natgunanathan, I., Rong, Y. and Guo, S. (2015). Spread spectrum-based high embedding capacity watermarking method for audio signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, volume 23, number 12, pp. 2228–2237.

Xu, Z., Ao, C. and Huang, B. (2016). Channel capacity analysis of the multiple orthogonal sequence spread spectrum watermarking in audio signals. *IEEE Signal Processing Letters*, volume 23, number 1, pp. 20–24.

Y. Xiang, I. Natgunanathan, D. P. W. Z. S. Y. (2015). A dual-channel time-spread echo method for audio watermarking. *IEEE Trans. on Information Forensics and Security,*, pp. 383–392.

Yamamoto, Y., Chinen, T. and Nishiguchi, M. (2013). A new bandwidth extension technology for mpeg unified speech and audio coding. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 523–527.

Yeo, I. and Kim, H. J. (2003). Modified patchwork algorithm: a novel audio watermarking scheme. *IEEE Transactions on Speech and Audio Processing*, volume 11, number 4, pp. 381 – 386.

Yiqing Lin, W. H. A. (2014). *Audio Watermark: A Comprehensive Foundation Using MATLAB*. Springer.

Yuan, S. and Huss, S. A. (2004). Audio watermarking algorithm for real-time audio integrity and authentication. *Proceedings of the 2004 Workshop on Multimedia and Security*, pp. 220–226.

Zhang, P., Xu, S. Z. and Yang, H. Z. (2012). Robust audio watermarking based on extended improved spread spectrum with perceptual masking. *International Journal of Fuzzy Systems*, volume 14, number 2, pp. 289–295.