SPARSE SIGNAL DECOMPOSITION TECHNIQUES FOR MULTIMEDIA FINGERPRINTING

by

XIAOLI LI

B.Eng. in Electrical Engineering, P.R. China, 1996
MASc. in Electrical and Computer Engineering Program with Specialization in Computer Networks, Canada, 2004

A dissertation
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Doctor of Philosophy
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2011

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

Xiaoli Li

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Xiaoli Li

Abstract

SPARSE SIGNAL DECOMPOSITION TECHNIQUES FOR MULTIMEDIA FINGERPRITING

Xiaoli Li

Doctor of Philogophy
Electrical and Computer Engineering
Ryerson University, 2011

This dissertation focuses on digital multimedia content protection from the copyright point of view. Several approaches aiming to resolve the challenge to some extent in the emerging area of multimedia protection were proposed and studied.

This study proposes an approach to secure the authorized media sharing in Peer-to-Peer (P2P) networks. The P2P networks was initially designed for bandwidth saving, but its file sharing property was later on put to use for pirate. This situation has not been improved effectively until now. The approach aims to embed an unique-mark (fingerprint) into each authorized copy in P2P networks so that it can be used to track the pirate initiator. This study also proposes another scheme for protecting the ownership of digital media files that have been circulated without copyright mark embedded. To protect this type of files, the ownership of each file needs to be stored associated with its meta-data (such as the ownership, title, and artist) and can be identified correctly later on. Since the size and the number of the media files to be stored are extremely large, the mini versions (fingerprints) of the files become necessary to be derived.

The common criteria of designing these two approaches are to ensure the fingerprint is compact, robust, discriminative, and ease of computation. To well balance the criteria, the sparse decomposition techniques play a very important role. The results of the tests under various distortions show the proposed fingerprinting schemes are very promising for real applications.

Acknowledgments

First, I would like to thank my supervisors, Prof. Sridhar Krishnan and Prof. Ngok-Wah Ma for their continuous and clear guidance, unconditional support from many aspects, high-standard requirement, and encouragement throughout the course of my research work. This work would not be possible without their patience and kindness. It is a great honor and pleasure to work under their supervision. I deeply appreciate your helps.

I would like to thank Prof. Ling Guan, Prof. Lian Zhao and Dr. Eddie Law for their insightful comments and suggestions in the beginning of my thesis work. I would like to acknowledge the Department of Electrical and Computer Engineering at Ryerson University, the Natural Science and Engineering Research Council of Canada for providing me financial support throughout my research work. I am grateful to my friends in SAR lab for providing such a warm and friendly environment. Thank you all for creating such a productive research atmosphere. Your friendship, help, and comments are very much appreciated. I would also like to express my gratitude to computer networks program and my friends: Arseny Taranenko, Amir Esmailpour, Maria Gracias, Veljko Knezevic, ..., thank you all for your trust, help, support, and friendship. I would like to thank the members of ELS at Ryerson University: Dr. Robert Roseberry, Christopher Brierley, ... Thank you all for helping me in English writing.

I would like to express my special thanks to my parents. Thank you for your love, patience, and encouragement. You are always the peaceful harbor whenever I feel frustrated and tired. To my sister and her family, thank you for your consistent support.

Contents

List of Figures

List of Tables

List of Appendices

List of Abbreviations

BER - Bit Error Rate
CBID - Content-Based audio Identification
CWT - Continuous Wavelet Transform
CWTs - Continuous Wavelet Transforms
DCT - Discrete Cosine Transform
DFT - Discrete Fourier Transform
DRM - Digital Rights Managements
DWT - Discrete Wavelet Transform
DWTs - Discrete Wavelet Transforms
FT - Fourier Transform
FFT - Fast Fourier Transform
HMM - Hidden Markov Models
HVS - Human Visual System
IMDCT - Inverse Modified Discrete Cosine Transform
KB - Kaiser-Bessel
KBD - Kaiser-Bessel Derived
LDA - Linear Discriminant Analysis
MDCT - Modified Discrete Cosine Transform
MMP - Molecular Matching Pursuit
MP - Matching Pursuit
P2P - Peer-to-Peer
PCA - Principal Component Analysis
SFM - Spectral Flatness Measure
SPSS - Statistical Package for the Social Sciences
SS - Spread Spectrum
STFT - Short Time Fourier Transform
SVD - Singular Value Decomposition
SVM - Support Vector Machine
TF - Time-Frequency
UGC - User-Generated-Content
WVD - Wigner-Ville Distribution

Chapter 1

Literature Survey and Motivations

1.1 Introduction

The recording industry has been fighting piracy for the last several decades. Especially the problem of the Internet piracy, since the digital multimedia content can be easily copied and transmitted without any loss of fidelity over the Internet. Therefore it is critical for content owners to take effectual steps to prevent their digital content from illegal duplication. Digital Rights Managements (DRM) were created to hold back the growing piracy. DRM is a collection of technologies from different disciplines, including signal processing, coding theory, information theory, the human visual/audio system, probability theory, detection and estimation theory. Algorithms derived from multiple disciplines have been incorporated and employed in watermarking and fingerprinting. This dissertation proposed fingerprinting approaches to support copyright protections.

The first proposed scheme is inspired by the inherent security problem within the P2P file sharing networks where the pirate utilizes the property of P2P networks to do free file sharing. Therefore, this approach is designed to trace the traitors who illegally distribute the authorized copy to non-authorized users in P2P networks. The whole design consists of the fingerprint generation, embedding, distribution, identification, ownership verification and robustness verification. The proposed scheme utilizes the concept of watermarking by embedding a unique watermark into each individual copy of the original media file before distribution. First, the wavelet technique obtains a low frequency representation of the

media file and the PCA technique finds the features of the representation. After that, a set of fingerprint matrices can be created based on a proposed algorithm. Finally, each matrix combines with the low frequency representative to become a unique fingerprinted matrix. The fingerprinted matrix is not only much smaller than the original media file in size but also contains the most important information. Without this information, the quality of the reconstructed media file will be very poor. In the distribution stage, the uniquely fingerprinted matrix will only be dispensed by the source host and leave the rest of the media file for P2P networks to handle. The proposed scheme is applied in video distribution. Our results indicate that the proposed fingerprint has shown strong robustness against common attacks such as Gaussian noise, median filter and lossy compression.

The second fingerprinting scheme aims to protect the legacy content and its original author's copyright. It is designed to find the most representable features (or signatures) of an ownership registered media in order to identify unauthorized sharing and block the pirate distribution of the media. We call this protection as legacy content protection. A scheme based on sparse signal analysis is proposed in this dissertation.

Before proposing this fingerprinting scheme, we first studied the signal classification using the Matching Pursuit (MP) technique [1]. In MP, the media is sparsely decomposed into atoms. The study focuses on analyzing the characteristics of the most representative atoms and selecting the distinctive features. With the assistance of the classification method Linear Discriminant Analysis (LDA), the features achieved 90% classification accuracy result for two music genres media. After the pre-studies, the scheme adopts Molecular Matching Pursuit (MMP) as a platform from where a series of features (or signatures, fingerprints) are analyzed and selected. The feature of each media is then evaluated through matching with the database using cross-correlation algorithms with and without LDA. The proposed scheme is applied to derive fingerprints of audio media. Experimental results show an identification accuracy between 78.3% and 95% for different types of popular attacks, such as low-pass filter, additive noise and MP3 compression. In order to get a generalized result, the leave-one-out cross validation method is applied. The identification accuracy was accordingly

between 70% and 82%.

The rest of the dissertation is organized as shown in Figure 1.1: The literature survey about the research that is related to the two fingerprinting schemes proposed in this dissertation is first presented in the rest of Chapter 1. The main techniques that are applied in the studies are introduced in Chapter 2. This Chapter also points out the motivations of the two proposed schemes. Chapter 3 describes details about the design of the watermark fingerprinting generation, fingerprint embedding, distribution, recovery and robustness tests for the P2P file sharing networks. Chapter 4 which is the pre-step of Chapter 5, introduces the signal decomposition approach for music classification. Chapter 5 elaborates on the procedure of the fingerprint extraction for the audio legacy content protection, the simulation of attacks that are to distort the audio signal, and the identification procedure. The technical analysis of the studies are discussed in Chapter 6 by exploring the relationships and characteristics among the applied techniques. The conclusions of this dissertation are presented in Chapter 7.

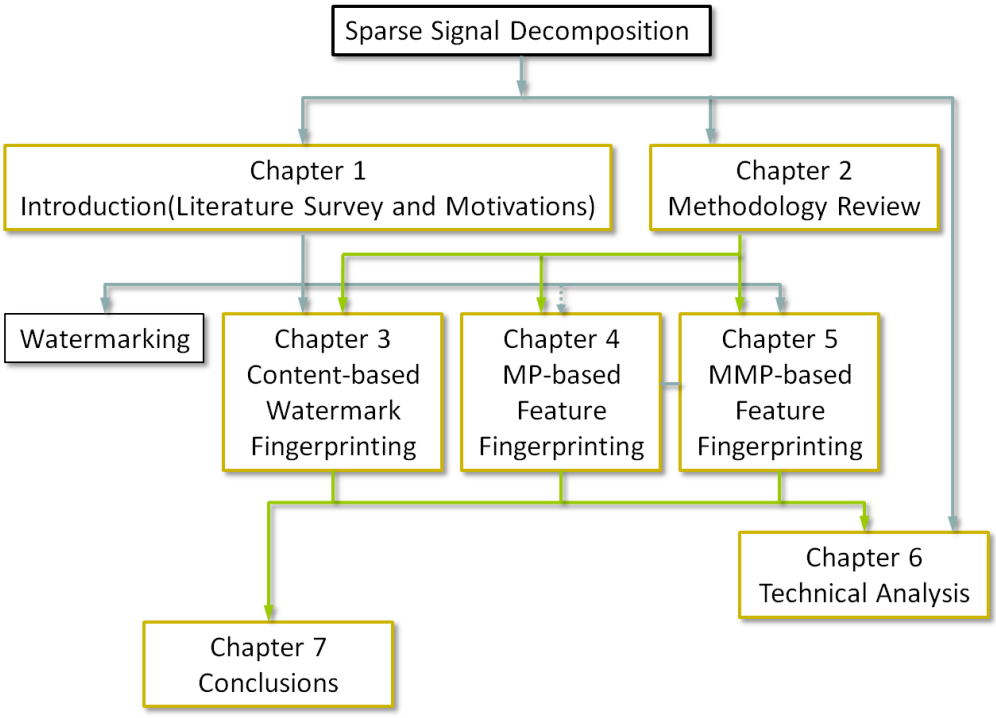


Figure 1.1: Topic Flowchart.

This Chapter will review the design of fingerprinting systems from two different perspectives and introduce the main research that has been done in these fields for copyright control. Then the motivation and the inspiration that lead to the proposed schemes will be presented. Finally, the techniques that are utilized in the studies to meet the criteria will be discussed.

Before describing the Chapter in details, the important contents covered in this Chapter is briefly summarized in Figure 1.2.

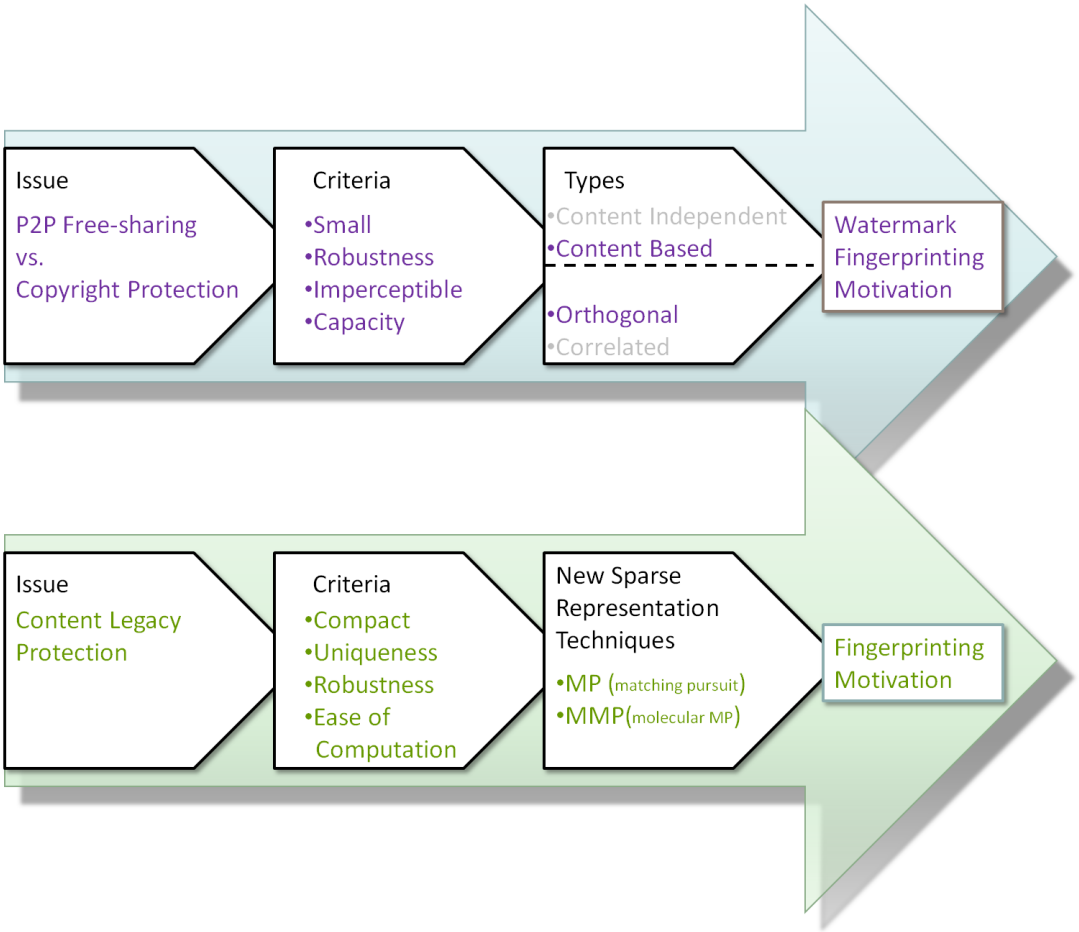


Figure 1.2: Main Contents of the Chapter.

1.2 Definitions of Watermarking and Two Types of Fingerprinting

Before we review the fingerprinting techniques and their current research status, we provide the basic definitions of watermarking and two types of fingerprinting.

Watermarking is a process that embeds a mark, the watermark, into the original digital signal. This mark should not degrade the signal quality, but it should be detectable and nonremovable. Compliant devices should check for the presence of a watermark to prevent copyright infringement before proceeding to operations.

Spread Spectrum (SS) watermarking is an example of spatial embedding of watermarks. In general, most watermarking techniques are considered a variation of the spread spectrum technique [2]. The watermark is modulated by a pseudo-noise signal in order to produce a spread spectrum signal, which is then scaled according to the required power. The modulated watermark is then added to the original signal to produce the watermarked signal. To detect the watermark, a high pass/edge detection or Wiener filter is applied to the watermarked signal to remove irrelevant information. The output of the filter is then correlated with the modulating pseudo-noise signal used at the transmitter side and compared to a predetermined threshold for the detection of the watermark.

Fingerprinting is conceptually quite different from watermarking. Fingerprinting first analyzes a signal and then constructs a "fingerprint" that is uniquely associated with the signal. Thus, it can be used to identify a media file by searching for its fingerprint in a previously constructed database. For example, the fingerprinting technique is being used to monitor music transfers in Napster-like file sharing facilities, blocking transfers of copyrighted material or collecting the corresponding royalties and to track audio content played by broadcasters [3].

At this point, we should clarify the term *fingerprinting* which has been deployed for many years. Traditionally, fingerprinting has been used to generate a short numeric sequence (*fingerprint*) associated with a multimedia signal for the identification of the signal. In the rest of the dissertation, the traditionally defined fingerprinting is named as *feature fingerprinting*.

Thus, it can be distinguished from another type of fingerprinting, called *watermark fingerprinting*, which has been used for a different purpose. Watermark fingerprinting uniquely watermarks (content-related or non-content-related) each legal copy of a media file so that the copy can be traced to the individual who acquired it [4]. The similarity is they both can uniquely represent the signal, which is an analog to a human fingerprint that uniquely represents an individual person, either by the extracted digital *fingerprint* or by the embedded digital *fingerprint*. Both the watermark fingerprinting and the feature fingerprinting approaches are studied and will be described in details in Chapters 4, 5 and 6. The former one will first be reviewed in the next section, and the latter one will be reviewed in Section 1.5.

1.3 Literature Survey of Watermark Fingerprinting

In the following, we will review the development of *watermark fingerprinting* technique for digital multimedia copyright control. Besides the broad sense of watermark fingerprinting, identifying the pirate initiator, known as *traitor tracing*, is another difficulty that multimedia owners/publishers have to resolve. As mentioned at the beginning of this Chapter, watermark fingerprinting as the technique for this type of copyright control that aims at traitor tracing for the prevention of the resource leakage, has been, and is, being explored by researchers. Figure 1.3 depicts the procedure of the watermark fingerprinting system. This figure shows that before the original media file is distributed to the authorized users, each copy will be embedded an unique fingerprint which is corresponding to the user. Even though some user, such as user 5, tries to erase the embedded fingerprint and shares with other un-authorized users without being detected, the robust fingerprinting technique should still be able to help the owner to identify the pirate (user 5). This is possible because the original file is only known by the owner. The pirate, who is not given enough information, will hardly erase the fingerprint completely.

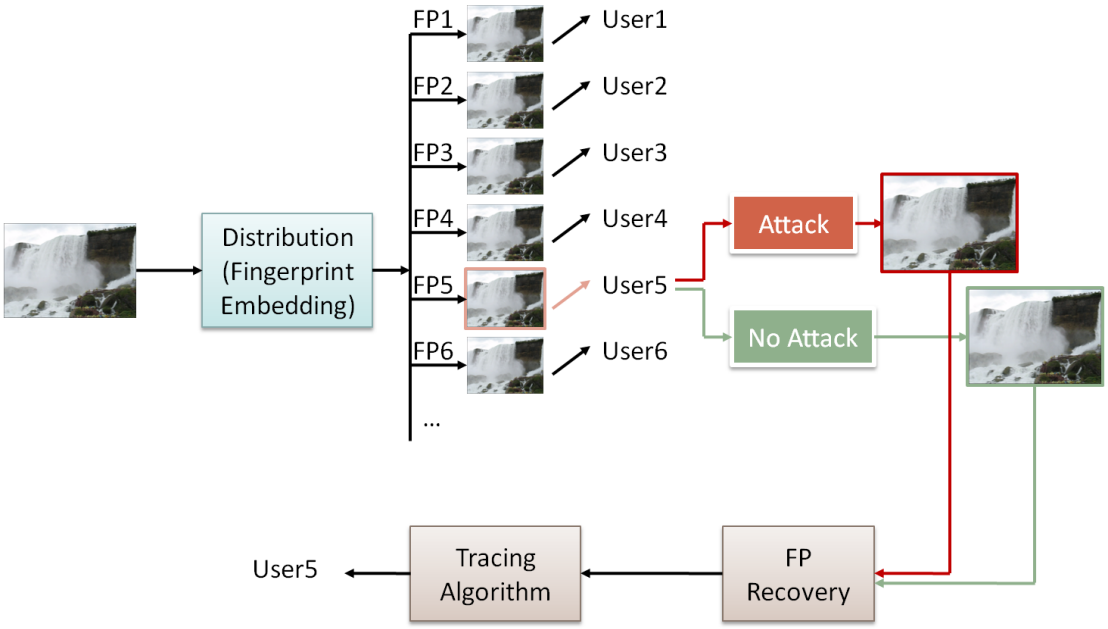


Figure 1.3: Watermark Fingerprinting System

1.3.1 Watermark Fingerprinting Principles

An effective media watermark fingerprinting scheme must be the one that can successfully resolve a set of compromises. These compromises exist among fingerprint capacity, robustness and perceptual quality.

- Capacity is defined as the total number of unique fingerprints that can be embedded and successfully recognized at the receiver
- Robustness reflects the inability for one or more pirates to erase or forge the fingerprint without affecting the commercial quality of the media
- Perceptual quality requires that the fingerprints are hidden in the media so that they are unavailable to the users.

According to different applications, some specific requirements of the embedded fingerprint will be considered. Our study that applies fingerprint embedding in the P2P networks

also requires that the fingerprint should be very compact and essential content related. More details will be discussed in Chapter 4.

1.3.2 Watermark Fingerprinting Design

Luh and Kundur [5] claimed that the content-based fingerprinting technique is more favoured than the fingerprinting over codebook because of its obvious shortcomings related to the security. First, since the codebook is designed independently from the digital entertainment media, it may not be robust against the signal-processing-based collusion attacks. Second, due to the same reason, common collusion attacks on uniquely fingerprinted copies using the codebook design do not result in perceptual changes. Therefore, the fingerprints should be designed based on the media content [5]. The material of the fingerprints that Luh and Kundur used were the features of the media, the motion vector (or the magnitude of the spatial adjustment), under a threshold between two segments correspondingly from two consecutive frames. The result of the content-based fingerprint proved that such fingerprints outperformed the codebook-based fingerprints.

From the perspective of the correlation between fingerprints, two major classes of fingerprinting strategies were proposed [6]. One is called orthogonal fingerprinting and the other one is called correlated fingerprinting. In orthogonal fingerprinting, its simplicity of encoding and embedding are attractive to identification applications. However, the disadvantages that consist of limited number of fingerprints, computational complexity in the detection stage, and weak identification capability as the number of colluders increases should be resolved. Ergun et al [7] used a set of mutually orthogonal watermarks as fingerprints to identify between each other. The problem of using orthogonal fingerprinting was the computational complexity. This is because of the classical method where the correlation of each test signal against each fingerprint is calculated. Trappe et al. [8] developed a recursive detection mechanism to reduce the computational complexity. The idea was to divide the set of fingerprints into two groups and compute the summation of fingerprints in each group. They then calculated the correlation of each test signal with these two summed fingerprints,

respectively. One evaluation method they used is based on which group of fingerprints gives higher correlation value. Therefore, the selected one group will be further divided into two subgroups and the correlation value would be calculated with the test signal in the same way at each iteration until the matched fingerprint was found. The other method was to use a threshold, thus the chance that one or two groups was chosen to get into the next recursion was possible.

On the contrary, correlated fingerprinting can have more dimensionalities than orthogonal fingerprinting, counteract the energy deduction and simplify the detection computation. Therefore, this class is usually applied for a large group of users. However, the procedure of its encoding is more complicated than the orthogonal class. Boneh and Shaw [9] proposed a collusion-secure (resist to collusion attacks) fingerprinting scheme which was designed to find, for each copy of the object, the right set of marks that helped to prevent collusion attacks. They obtained ($c > 1$)-secure codes that were capable of identifying a guilty user (or called pirate) in a coalition of at most c pirates with a probability ε of failing to do so. The construction of Boneh and Shaw scheme composed an inner binary code with an outer random code. The length of the code was considerably large for small error probabilities and a large number of users. However, the main issue was that the identification algorithm involved the decoding of a random code, which is known to be a NP-hard problem [10]. To reduce decoding complexity, the codes in [10], [11], [12] and [13] represented an important improvement compared to the Boneh and Shaw proposal. In [2], Tomàs et al. applied the collusion-secure fingerprinting scheme in [12] for traitor tracing over the YouTube video service. They claimed that this fingerprinting code can trace traitors under no collusion performed, with a very low distortion. If collusion appears, the traitors could be also traced with a low distortion.

Another point of view about fingerprinting codes applied to media has been presented recently by He and Wu [14]. In this system [14], the idea was to divide each segment of the fingerprint, which corresponded to one symbol, into several subsegments, and then to randomly permute these subsegments before embedding. At the detection stage, inverse

permutation was performed on these subsegments, followed by a correlation detector to identify traitors. By taking advantage of prior knowledge that some users are more likely to collude together than with others, possibly due to geographical or cultural reasons, they proposed a group-based joint coding and embedding (GRACE) technique. In GRACE, each fingerprint consisted of a user subcode and a group subcode, and was embedded in the host signal via the SS technique. The fingerprints, which were assigned to different users in the group, were orthogonal to each other.

The last two paragraphs illustrated the correlated fingerprinting schemes, but they were all designed based on codebook. Therefore, they were completely independent of the content to be protected.

1.4 Motivation in Watermark Fingerprinting

The proposed scheme is inspired by the fact that copyright owned media files are much easier to be illegally shared due to the appearances of the P2P and user-generated-content (UGC) networks.

From a technical point of view, content-based watermark fingerprinting is more favorable than codebook watermark fingerprinting. The literature about video feature fingerprinting (shown in Section 1.5.2) also concludes that among four feature domains, the spatial feature performs the best and the transform-domain feature is more resilient to geometric distortions. In fact, the transform-domain feature represents some spatial features as well. Thus our approach will generate content-based fingerprint motivated by these two features.

1.4.1 Free Sharing vs. Copyright Protection

The first P2P application, Napster, appeared in 1999. It was originally a music file sharing application and had gained a big success. Due to the obvious advances in saving bandwidth, P2P systems and applications have been rapidly developed over the past several years. Nevertheless, the literature survey shows that P2P networks have not been well developed. Many aspects of P2P networks need to be enhanced. The P2P file sharing system (such as Napster

(2002), Gnutella wego (2002) and Gnutella news (2002)) creates an environment that its users can distribute copyrighted content without authorization of content owner. Therefore, the proliferation of online multimedia in the P2P networks has brought a renewed challenge in multimedia watermark fingerprinting technique for solving the copyright violation problems. Particularly for enterprisers or individuals who publish their resources in P2P networks to the authorized customers, the copyright control (allows the publishers to trace the source leakage point) and access control become important issues. The failure of protecting the copyrighted file from free sharing resulted in the shutdown of some P2P service companies through prosecution [15][16].

It is strongly suggested that this fingerprinting technique is much needed especially for those multimedia producers who like to share their valuable multimedia with the subscribed customers privately within the public P2P networks. Watermark fingerprinting can help the source owner to trace the traitor who leaks out the multimedia file. This research will focus on addressing this problem. It is the first known study of copyright control based on traitor tracing using a watermark fingerprinting technique for P2P networks.

A good fingerprint technique developed for the P2P application should be: (a) robust against attacks, (b) have negligible impact on the quality of the multimedia file and (c) have higher capacity. In addition, the technique for media distribution in P2P networks should also incorporate the P2P feature in the design, which requires the embedded fingerprint to be as small as possible. In this study, a digital watermark fingerprinting technique for an image/video file based on PCA and wavelet is proposed.

1.4.2 Content-based Watermark Fingerprinting

Literature review shows that very few methodologies and strategies have been proposed in DRM for P2P applications. One research group [17] recently proposed a watermarking scheme for P2P application. In their scheme, there were multiple keys as the watermarks were casted into the image by the pseudorandom sequence of a Gaussian distribution generator. The detection of the watermarks relied on the statistical correlation between the

queried watermark and original stored watermark. This correlation method required strong statistical dependency of the watermark sequence. However, the paper did not mention the robustness of the scheme against common attacks. In addition, this proposed watermark generation is independent of the content.

Literature review in Section 1.3.2 tells that very few researches focus on content-based fingerprinting, except the one done by Luh and Kundur in [5], even though the content-based fingerprinting technique is more favored than the fingerprinting over codebook because of its obvious shortcomings in the security.

Among the schemes applying wavelet techniques, one [18] proposed an algorithm in the PCA/Wavelet-transform domain. They first applied PCA to produce eigenimages and then decomposed them into multi-resolution images. Correspondingly, the watermark image was also decomposed into a multi-resolution image in the same scale. Finally, the Human Visual System (HVS) as the strength parameter was adopted for watermark embedding. The scheme, however, was applicable for embedding one mark because of the uniqueness of the strength parameter. Liu et al. [19] proposed their algorithm based on a Singular Value Decomposition (SVD). The host image is originally presented as USV^{-1} where the matrix S contained the singular values and U, V were the singular vectors. The algorithm added the watermark to the singular values S, thus the modified S was presented by $U_w S_w V_w^{-1}$. Then the newly generated singular values S_w would replace the original S to generate the watermarked image. The singular vectors U_w and V_w were kept by the owner just for watermark detection. Since S_w was approximately equal to S, the visual quality of the image was preserved. To extract the watermark, the watermarked image would be decomposed again using SVD. The corrupted singular values $S_w^{'}$ and the singular vectors U_w, V_w would recover the watermark. The main issue of this method was that the attacker could also claim his/her watermark easily by providing another set of singular vectors, such as U_a, V_a. In other words, the recovered watermark depends more on the selected singular vectors. It proves that embedding a watermark (or fingerprint) only on singular values is unreliable. Hien et al. [20] also proposed a PCA method. The difference was they embedded the watermark into the

eigenvectors. First, the PCA process decomposed the image into eigenvectors and eigenvalues. The image was then projected onto each eigenvector and became a coefficient matrix. The watermark was embedded into the coefficient matrix based on the selected components. Finally, the watermarked image was obtained by applying the inverse PCA process. The robustness became the issue of this method as well. Since the eigenvectors were normalized, the numerical value of each component of the eigenvector was very small and could be easily corrupted by distortion methods.

In our proposed watermark fingerprinting scheme, the system is designed for P2P networks (but not limited to it). It is based on a content-based fingerprint generation by combining the PCA, Discrete Wavelet Transform (DWT) technologies and security concept.

1.5 Literature Survey of Feature Fingerprinting

As explained in Chapter 1, the feature fingerprinting for digital multimedia legacy content protection is the technique that derives the characteristic information (fingerprint) from the multimedia. It is very suitable to deal with the legacy content, that is, with multimedia material released without watermark. The reason is watermark must be embedded in multimedia during the production, and therefore cannot be used to identify the multimedia that missed the procedure and has been in circulation. The idea of using fingerprint for DRM is that the owner can claim the ownership of a piece of multimedia content by registering the content's fingerprint (the compact representation of the content) with the owner's name to a centralized database. If the fingerprint is robust enough to withstand a certain degree of the content alteration, it can be used to prevent anyone else from making a false claim of the ownership of the same piece of multimedia content. Therefore, the feature fingerprinting method can be applied to identify the legacy content, while watermarking cannot.

In 1999, a few graduate students at Stanford University wrote a technical report entitled, "Finding pirated video sequences on the Internet" [21]. The work contained in that technical report later on lead to two distinguished Ph.D. dissertations by Shivakumar [22] and Indyk [23], respectively. This work was the first to use fingerprint in unauthorized copyrighted

video identification on the Internet.

In fact, there are more feature fingerprinting applications than identification. Specifically, it can also be used for verification of content-integrity, similar to fragile watermarks. Since the dissertation mainly focuses on the audio identification for content legacy protection, the following review is more related to identification.

1.5.1 Feature Fingerprinting Principles

In feature fingerprinting, the main objective is to create a mechanism that is able to detect similarities between two media files by comparing the associated fingerprints. A fingerprinting scheme consists of two steps:

1. Construction of the database: in this stage, the signals to be recognized are presented to the system. The system processes them by extracting a unique fingerprint of each of them based on the characteristics of the content. This fingerprint is then associated with its corresponding meta-data (artist, title, ownership information, release dates and genres, etc.) and, finally stored together in the database. An overview of this step according to the designed scheme is illustrated in Figure 1.4.

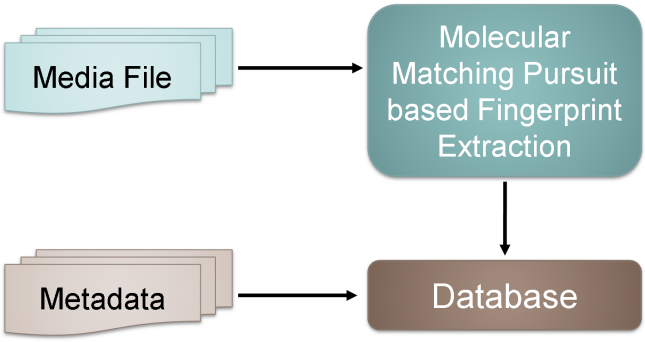


Figure 1.4: Populating the database

2. Identification of a new media file: when a new unknown file is presented to the system, it is processed to extract its fingerprint. This fingerprint is then compared to those of the database. If a match is found, the corresponding meta-data is obtained from the database. Figure 1.5 presents, based on the designed scheme, the procedure of this stage.

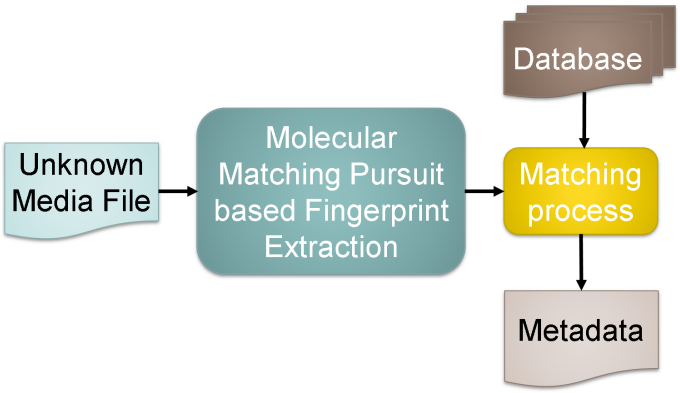


Figure 1.5: Identifying the new file

Generally, an ideal fingerprint should fulfill several requirements, as described in [24][25]:

- Uniqueness (a perceptual digest of the media file). This digest should allow the discrimination over a large number of fingerprints. This may be conflicting with other requirements, such as complexity and robustness.
- Robustness (Invariant to distortions). A fingerprint should be invariant for the same media content under various types of processing, transformations and manipulations, such as format conversion, transcoding, and content editing.
- Compact. It is desirable to keep the size of the fingerprint small since a large number of fingerprints need to be stored and compared. An excessively short representation, however, might not be sufficient to discriminate among recordings, affecting the accuracy, reliability and robustness.
- Ease of computation. The extraction of the fingerprint should not be excessively time-consuming.

The main advantage of feature fingerprinting is the simplicity due to the fact that it is much easier to compare the fingerprints of files than to compare the entire file. Other advantages are:

- it reduces storage cost due to the small sizes of the fingerprints;

- it provides more effective comparison because a fingerprint only carries relevant components so that noise is removed;
- faster searching because of a smaller size of fingerprint database compared to the original file database.

The solutions proposed to fulfill the above requirements imply the trade-offs among uniqueness, compactness and robustness.

The fingerprint extraction consists of two steps. The first step (called front-end in [25]) computes a set of measurements from the signal. The second step (called fingerprint model block in [25]) defines the final fingerprint representation, e.g: a vector, a trace of vectors, a code-book, a sequence of indexes to Hidden Markov Models (HMM) sound classes, a sequence of error correcting words or meaningful high-level attributes.

Given a fingerprint derived from a media file, the matching algorithm searches a database of fingerprints to find the best match. A way of comparing fingerprints, that is a similarity measure, is needed for the matching process. Since the number of fingerprint comparisons is high in a large database and the similarity can be expensive to compute, matching methods are required to speed up the search. Some feature fingerprinting systems use a simple similarity measure to quickly discard the unlikely candidates and then use a more precise, but expensive similarity measure, for the reduced set of candidates.

1.5.2 Front-End

In general, the front-end procedure consists of three steps: preprocessing, feature extraction, and post-processing.

a. Preprocessing

The media file is digitalized and converted to a general format. For a video file, the preprocessing includes segmentation, high-pass filtering or low-pass filtering, color converting, and frame block-dividing. For an audio file, it may be converted to a desirable digital format mono PCM (16bits), with a fixed sampling rate (ranging from 5kHz to 44.1kHz). Sometimes

the audio is preprocessed to simulate the channel, e.g: band-pass filtered in a telephone identification task.

b. Feature Extraction

b.1 Video Almost all the video features that have been proposed to date can be summarized into four types: spatial, temporal, color, and transform-domain. The fingerprint can be derived from one feature, or the combination of features. A spatial fingerprint describes spatial features of a video frame and is computed independent of other frames. Examples of the features comprise luminance patterns, differential luminance or gradient patterns, and edges. A temporal fingerprint characterizes temporal features of a video and is computed between two frames in the temporal direction. Examples of temporal features (or motion features) include frame difference measures, motion vector patterns, and shot durations. A color fingerprint presents color characteristics of a video frame and is derived from a color space, such as RGB or YUV. Many color fingerprints are an abstraction of patterns in the color histogram. A transform-domain fingerprint captures the spatial and/or temporal features in the transform domain of a video frame. It is computed from coefficients of an image or video transform, such as a Fourier Transform, Discrete Cosine Transform (DCT) or wavelet transform. For example, the transforms, such as polar Fourier Transform or Singular Value Decomposition, are chosen is because of the robust properties of the transform to rotation, translation and geometric distortions [26].

Among the four types of fingerprints, judging by the criteria, such as robustness, discriminability, compactness, low complexity, the spatial fingerprint performs the best, particularly the block-based spatial fingerprint. Temporal and color fingerprints, useful in improving discriminability, tend to lack of robustness in comparison to spatial fingerprint. This claim is supported by the experimental result reported in the research literature, the industry evaluation tests, and the successful applications in real commercial systems. Even though the transform-domain fingerprints were not widely adopted in video feature fingerprinting due to their computational complexity, they are the necessities when geometric distortion can

not be effectively compensated by other fingerprints.

b.2 Audio Audio feature extraction is different from video due to its nature. The procedures and available approaches are summarized in the following.

- **Framing and Overlap**: A key assumption in the measurement of characteristics is that the signal can be regarded as stationary over an interval of a few milliseconds. Therefore, the signal can be divided into frames of a size comparable to the variation velocity of the underlying acoustic events. The number of frames computed per second is called a frame rate. A tapered window function is applied to each block to minimize the discontinuities at the beginning and end. Overlap must be applied to assure robustness to shifting (i.e. when the input data is not perfectly aligned to the recording that was used for generating the fingerprint). There is a trade-off between the robustness to shifting and the computational complexity of the system: the higher the frame rate, the more robust to the shifting the system becomes, but at a cost of a higher computational load.

- **Signal Decomposition - Spectral Estimates**: If the transform is suitably chosen, the redundancy is significantly reduced. There are optimal transforms in the sense of information packing and de-correlation properties, like Karhunen-Loéve (K-L) or SVD, which both will be explained in details in Section 2.1. These transforms, however, are problem dependent and computationally complex. For that reason, lower complexity transforms using fixed basis vectors are common. Most content-based audio identification (CBID) methods, therefore, use standard transforms from time to frequency domain to facilitate efficient compression, noise removal and subsequent processing. Lourens [27] uses power measures for computational simplicity and Kurth et al. [28] use power measures to model highly distorted sequences, where the time-frequency analysis exhibits distortions. The power can still be seen as a simplified time-frequency distribution, with only one frequency bin.

The most common transformation is the Discrete Fourier Transform (DFT). Other transforms have been proposed, such as the DCT, the Haar Transform (one of the wavelet transform) or Walsh-Hadamard Transform [29].

Since most of the real-world signals are non-stationary, the study and analysis of non-stationary signals is receiving more and more attention in the scientific community. For signal analysis, time series and frequency spectrum contain all the information about the underlying processes of signals. By themselves, however, the best representations of non-stationary processes may not be well presented. Due to the time-varying behavior, techniques which give joint Time-Frequency (TF) information are needed to analyze non-stationary signals. The TF distribution is best suited for non-stationary signals, which need all the three axes of time, frequency and energy to represent them efficiently. TF decomposition breaks down a signal into elementary building blocks, TF atoms, to represent the inner structure and the processes. It can better reveal the joint TF relationship and can be useful in determining the nature of the many kinds of non-stationary signals. The success of any TF modeling lies in how well it can model the signal on a TF plane with optimal TF resolution. Different analysis techniques to decompose signals into TF atoms (or basis functions) have been developed. Fourier analysis and wavelet transform are the most common examples of such signal analysis models.

Most research was interested in the Fourier Transform and its derivatives. For example, the Fast Fourier Transform was used in [30] by Seo et al. and in [31] by Haitsma and Kalker. In [32], Ramalingam et al. proposed an audio feature fingerprinting scheme based on the Short-Time Fourier Transform, which is also used in [33] by Sungwoong et al..

On the other hand, in [34], Lu used features based on the wavelet coefficients of the Continuous Wavelet Transforms, while Subramanya et al. extracted features from the Walsh-Hadamard Transform in [29]. Coefficients from the Modulated Complex Lapped Transform were used by Mihak et al. in [35] and by Burges et al. in [36],

which exhibited approximate shift invariance properties [35].

In many cases, however, the basis functions in TF domain are orthogonal to each other, such as for the cosines and sines function in Fourier and wavelets bases. Orthogonal basis functions are suitable for data compression applications, but they exhibit drawbacks for modeling non-stationary signals in feature extraction application [37]. In [37], MP with Gabor dictionary is applied to model the signal. Note that atoms in Gabor dictionary can reach the best possible TF resolution. This is due to the fact that the TF resolution is limited to the lower bound of the Heisenbergs uncertainty principle and it has been proven that only Gabor functions or atoms (Gaussian) satisfy the lower bound condition [38]. Gabor dictionary is more flexible and adaptive than wavelets since there is no restriction on windowing patterns and the scaling parameter is independent of frequency. Since the expansions are not constrained to orthonormal bases, MP is better adapted to the time-frequency localization of signal decomposition. In [39], Chu et al. also proposed an MP-based method to classify the ambient environmental sounds. The proposed MP-based method utilizes a dictionary from which features of ambient environmental sounds are selected, resulting in successful classification.

- **Transform-domain Analysis for Feature Extraction**: On a time-frequency representation, additional transformations are applied in order to generate the final acoustic vectors. Even when there is great diversity of algorithms, the objective is to reduce the dimensionality and, at the same time, to increase the robustness to distortions.

  Haitsma and Kalker [31] summarized that the set of relevant features can be broadly divided into two classes: the class of semantic features and the class of non-semantic features. Typical elements in the former class are *genre, beats-per-minute*, and *mood*. These types of features are called high-level features since they usually have a direct interpretation, and are actually used to classify music, generate play-list and more; the features in the latter class are called low-level features because they have a more

mathematical nature and are difficult for humans to 'read' directly from music. In this study, the features are known as low-level features.

It is true that the low-level feature extraction approaches also tend to extract perceptual meaningful parameters by considering the transduction stage of a human auditory system. Therefore, many proposed systems extracted features based on a critical-band analysis of the spectrum. In [40], the choice was the *Spectral Flatness Measure (SFM)*, which was an estimation of the tone-like or noise-like quality for a band in the spectrum. Papaodysseus et al. [41] presented the *band representative vectors*, which were an ordered list of indexes of bands with prominent tones (i.e. peaks with significant amplitude). The feature, *energy of each band*, was used by Kimura et al. [42]. Sukittanon and Atlas [43] proposed a modulation frequency analysis to characterize the *time-varying behavior* of audio signals.

On the other hand, approaches from music information retrieval include features that have proved valid for comparing sounds [24]: *harmonicity, bandwidth*, and *loudness* [44].

c. Post-Processing

Features can be absolute measurements and further derived by performing some procedures, such as, normalization, de-correlation, differentiation, and quantization.

For example, in order to better characterize temporal variations in the signal, higher order time derivatives are added to the signal model. Some systems only use the derivative of the features, not the absolute features [40][28]. Some systems compact the feature vector representation using transforms (e.g. PCA [45][46]).

1.5.3 Fingerprint Modeling

After the post-processing, the fingerprint extraction turns into the second stage, the fingerprint modeling. The fingerprint modeling block usually receives a sequence of feature vectors calculated on a frame by frame basis. Exploiting redundancies in the frame time vicinity,

inside a media file and across the whole database, is useful to further reduce the fingerprint size.

Using audio feature fingerprinting modeling as example, a very concise form of fingerprint is achieved by summarizing the multidimensional vector sequences of a whole song (or a fragment of it) in a single vector. Etantrum [47] calculated the vector out of the means and variances of the 16 bank-filtered energies corresponding to 30s of audio ending up with a signature of 512 bits. The example above was computationally efficient and produced a very compact fingerprint. It was designed for applications like linking mp3 files to meta-data (title, artist, etc.) and were more tuned for low complexity (both on the client and the server side) than for robustness (cropping or broadcast streaming audio). Some systems include high-level musically meaningful attributes, like rhythm [48] or prominent pitch.

1.6 Motivation in Feature Fingerprinting

Designing a uniqueness, robustness, compact, and low complexity fingerprint to fully represent and identify a unique media file is always the goal in a feature fingerprinting scheme. If the transform is suitably chosen, the selected features that characterizing the transformation should approximately fulfill the requirements above.

On the other hand, the newly emerged sparse representation techniques have not been well explored. The proposed scheme is motivated by the desire of looking into the potentials of such techniques for the feature fingerprinting.

1.6.1 Signal Decomposition

Compared to methods based on orthonormal transform, sparse representation techniques, such as MP, is used to find the representation of a signal x as a weighted sum of elements (or atoms) from an over-complete dictionary. It has been proved to offer better performance with its capacity for efficient signal modeling [49] or signal decomposition [50] - using less bases to represent the signal efficiently.

Mallet and Zhang [1] have stated that for a given class of signals, if we can adapt the

dictionary to minimize the storage for a given approximation precision, we are guaranteed to obtain better results by MP than decompositions on orthonormal bases.

Recent research has focused on three aspects of the sparse representation. First, the pursuit methods for solving the optimization problem, such as MP [1], orthogonal matching pursuit [51], and basis pursuit [52]. Second, the design of the dictionary, such as the K-SVD method [53], which is an iterative method that alternates between sparse coding of the examples based on the current dictionary and an update process for the dictionary atoms so as to better fit the data. Third, the applications of the sparse representation for different tasks, such as signal separation [54][55][56]. One research group [55] adopted psychophysiological evidence to build up an over-complete dictionary with 768 atoms for music, and applied the separable atoms on the promise that there is no modulation error stemming from the dictionary to linearly present the different music genres. The features were obtained by utilizing dimensionality reduction methods. The classification accuracy is high when the feature dimension goes up to a certain large number. Since the classification does not analyze the atoms of each genre, its accuracy only relies on the number of features.

The drawbacks of the sparse representation techniques mentioned above are the computational speed and the control of dictionaries. MMP improves the techniques by reducing the matching pursuit iterations and setting explicit dictionary structure. This algorithm was first introduced solely for audio decomposition in 2006 [57] and implemented for audio watermarking embedding in 2008 by Parvaix et al. in [58]. None of the sparse representation methods in the literature so far have used the Molecular Matching Pursuit algorithm for audio feature fingerprinting.

Therefore, MMP was taken as the transform method because it provides sparse representation of the audio signal, so that the features extracted can be more suitable for presenting as the fingerprint of the signal.

1.6.2 Feature Extraction

Once the transformed coefficients are derived, the features can be calculated to further reduce the dimensionality without loss in robustness. According to the methods introduced in Section 1.5.2, some features, such as finding the prominent tones (i.e. peaks with significant amplitude), energy of each band, time-varying behavior of audio signals, eigenvector of the coefficient pattern by PCA technique, harmonicity, bandwidth, and loudness are considered for applying on the MMP transformation domain.

Due to the platform, MMP transformation coefficients that the features are to be derived from, is different from the other approaches, the feature extraction implementation will be correspondingly adjusted.

1.7 Summary

This Chapter outlines the present research situation in the DRM field. The research interests driven by providing better fingerprinting techniques are pointed out. The driven reasons are the existing drawbacks in the application appearing in the latest decade, such as the P2P file sharing network, and the newly emerging fingerprint extraction platform, the MP and MMP sparse representation algorithms.

It is worth noting that, besides the basic requirements for the fingerprint (to be embedded or to be extracted), such as higher capacity, robustness and lower computational complexity, an additional requirement in the proposed DRM scheme is to reduce the dimension of the fingerprint. Since the dimension of decomposition methods determines the dimension of the fingerprint, applying the sparse approximation techniques seems to be very important. The next Chapters will present the key sparse approximation techniques that are applied in the studies.

Chapter 2

Sparse Signal Decomposition Methodologies

This Chapter will briefly review a number of traditional and modern sparse signal representation methodologies (except LDA) that are adopted in the studies are introduced in this Chapter. Figure 2.1 lists the methodologies and their connections with the proposed approaches. In addition, the beginning of each Section concludes where and how the methodology is used in the studies.

2.1 Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)

2.1.1 The Role of PCA (or SVD) in the Studies

This methodology is part of the the approach described in Chapter 6. The main task of this approach is to find the most representable features of a signal. One of these features is designed to be derived by the PCA methodology.

More importantly, the PCA methodology is introduced in the second step of the scheme presented in Chapter 4 to further extract the principal elements of the original signal. The extracted principal elements are then manipulated to generate the to-be-embedded fingerprints.

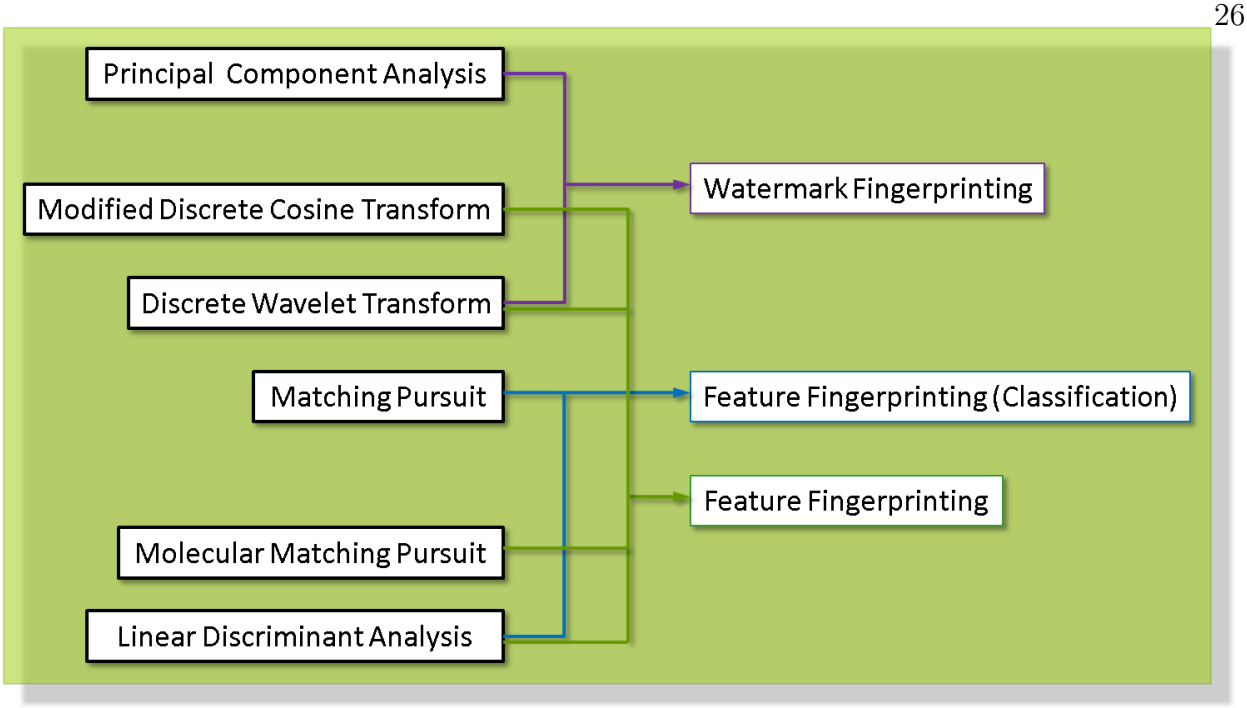


Figure 2.1: Methodologies For the Proposed Approaches.

2.1.2 Introduction of PCA

PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences, in other words, remove the redundancy of the data. The patterns of data can be considered as components of data. For instance, a pair of eigenvector and eigenvalue derived by PCA presents a component of the data. Thus the number of the groups of the eigenvector/eigenvalue pairs determines the number of components (or patterns) the data has. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once these patterns in the data are found, the data can be compressed, i.e., by reducing the number of dimensions (in other words, by eliminating the negligible eigenvectors/eigenvalues), without causing much loss of information. This technique can be used in image compression. Beyond that, PCA has been widely used in many other fields.

From the linear algebra point of view, the procedure of PCA is to find a matrix so that

the original data can be rotated to a new set of coordinates; and these coordinates highlight the correlations among the data as much as possible. The row vectors in the found matrix are called *eigenvectors*.

The implementation steps of PCA can be summarized as:

1. Get some data to build an $m \times n$ matrix X^o, where m represents the number of variables and n represents the number of samples.

2. Subtract the mean from each of the data dimensions, that is, every sample (the row vector of the matrix) subtracts the mean of its corresponding variable, represented by μ_{X^o}, to get the matrix $X = X^o - \mu_{X^o}$.

3. Calculate the covariance matrix of X.

4. Calculate the eigenvectors and eigenvalues of the $m \times m$ covariance matrix XX^T. We know that for an $m \times m$ matrix XX^T, a nonzero vector p is the eigenvector of the matrix if:

$$XX^T p = \lambda p \tag{2.1}$$

   where the scaler λ is called the eigenvalue of XX^T, and p is said to be an eigenvector of XX^T corresponding to λ. Since $XX^T p = \lambda p$ then $(XX^T - \lambda I)p = 0$, where I is called identity matrix. For a unique set of eigenvalues, the determinant of the matrix $(XX^T - \lambda I)$ must be equal to zero. Thus, from the solution of the equation, we are able to obtain the values of λ. When we substitute the variable λ with the value, we will get the ratios which relates the values of the elements of the corresponding eigenvector. The ratios lead to the real values such that the eigenvector is a normalized vector.

Notice that if we resolve all the eigenvalues and eigenvectors, we can define Eq. (2.1) as:

$$XX^T P = P\Lambda \tag{2.2}$$

where the eigenvector in the *ith* column of P is p_i corresponding to the *ith* eigenvalue λ_i of Λ. The matrices P and Λ are called *eigenvector* matrix and *eigenvalue* matrix, respectively,

and Λ is a diagonal matrix. Since $P^{-1} = P^T$, Eq. (2.2) has another form as:

$$P^T X X^T P = \Lambda \tag{2.3}$$

$$P^T X (P^T X)^T = \Lambda \tag{2.4}$$

This equation tells that the covariance matrix of $P^T X$ is a diagonal matrix.

2.1.3 The Significance of Principal Component Analysis

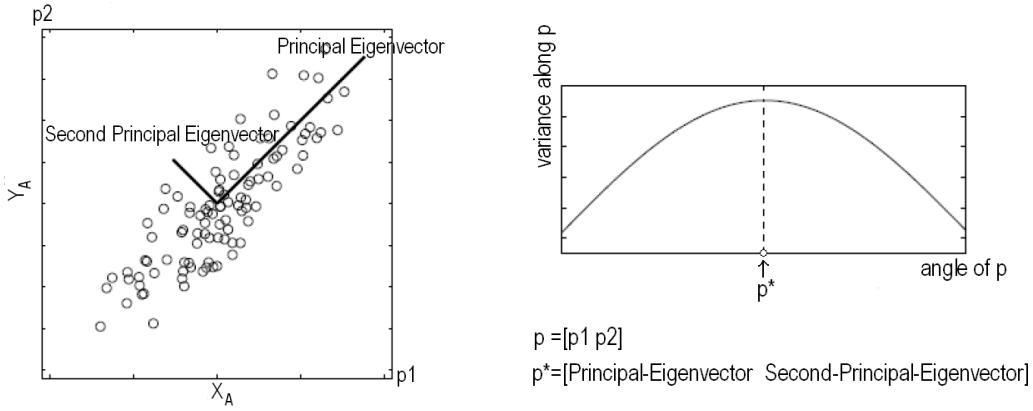
It is worth to mention that the covariance matrix being a diagonal matrix has important meanings. The elements on the diagonal are the variance of the variable itself; the elements on the off-diagonal are the covariance of two variables. Since covariance illustrates the correlation of the variables, if two variables are uncorrelated, the elements on the off-diagonal will be zero.

We can use the concept of variance to reduce the data redundancy. For example, Figure 2.2(a) shows the distribution of a data set which is a function of two variables $\mathsf{X_A}$ and $\mathsf{Y_A}$. By projecting the data set on the set of basis vectors p, ($\mathsf{p} = [\mathsf{p1}\ \mathsf{p2}]$), we obtain different variances. It can be proved that the maximum variance occurs if the data set is projected on the principal eigenvector of the data set, which is denoted by p^* in Figure 2.2(b). Therefore, the covariance matrix of $P^T X$ being a diagonal matrix implies that all rotated data are only distributed along the basis (eigenvectors). The data being distributed on the eigenvector corresponds to a larger eigenvalue among the diagonal elements in Λ means they contribute more energy on that eigenvector. This is why the eigenvector can rotate the original data X to a new set of coordinates; and these coordinates highlight the correlations among the data as much as possible.

Similar to XX^T, $X^T X$ has its eigenvalues and eigenvectors which are defined as Λ' and Q and the equation is represented as follow:

$$X^T X Q = Q \Lambda'. \tag{2.5}$$

Since the determinant of the eigenvector matrix either P or Q can be scaled down to normal bases ($P^T P = Q^T Q = I$) without violating Eqs. (2.3) and (2.5), the determinant of the



(a) (b)

Figure 2.2: Variances of Data. (a) The sampled data. The darker perpendicular lines indicate a set of eigenvectors (the principal eigenvector and the second principal eigenvector). (b) The variance reach to the maximum when the set of basis vectors p rotate 45^o degree. At this moment, the basis vectors p^* align with the eigenvectors.

eigenvalue matrix Λ and Λ' should be the same for the reason that $|XX^T| = |X^TX|$. That is, Λ equals to Λ'. Therefore, we have

$$X^TXQ = Q\Lambda. \tag{2.6}$$

2.1.4 Introduction of SVD

The SVD technique expresses a rectangular matrix $X_{m\times n}$ by a product of three matrices. The SVD theorem states:

$$X_{m\times n} = U_{m\times m}S_{m\times n}V^T_{n\times n} \tag{2.7}$$

where

$$U^TU = I_{m\times m} \tag{2.8}$$

$$V^TV = I_{n\times n}. \tag{2.9}$$

In Eq. (2.7), the columns of U are the left singular vectors; S has singular values and is diagonal; and V^T has rows that are the right singular vectors.

In fact, calculating the SVD consists of finding the eigenvalues and eigenvectors of XX^T and X^TX. The eigenvectors of XX^T make up the columns of U, the eigenvectors of X^TX

make up the columns of V. The singular values in S are square roots of eigenvalues from XX^T and X^TX. The relations can be proved as follows:

$$\begin{aligned} X &= USV^T; X^T = VS^TU^T && (2.10)\\ XX^T &= USV^TVS^TU^T && (2.11)\\ XX^T &= US^2U^T && (2.12)\\ XX^TU &= US^2, && (2.13) \end{aligned}$$

and the same is true for

$$X^TXV = VS^2. \quad (2.14)$$

Thus, the terms in SVD and PCA can be linked as:

$$\begin{aligned} S^2 &= \Lambda && (2.15)\\ U &= P && (2.16)\\ V &= Q, && (2.17) \end{aligned}$$

respectively.

2.1.5 PCA and K-L Transform

PCA is also categorized as a transform technique which is called Karhunen-Loève (K-L) Transform. This transformation technique utilizes PCA algorithm to transform signals, that is, finding an orthogonal matrix P, so that the covariance of the transform $Y = P^TX$ (or $Y = U^TX$, since $U = P$ as stated after Eq. (2.16)) is a diagonal matrix Λ as shown in Eq. (2.4).

K-L transform, similar to DWT and DCT, can be viewed as that X is the linear combination of bases (eigenvectors), and the coefficients are the projections of data X onto the eigenvectors. Assuming X is an $m \times n$ dimension matrix, the following is one way to write

X as Eq. (2.19) which is called the *rank one decomposition*.

$$\begin{aligned}
X_{m\times n} &= U_{m\times m}S_{m\times n}V^T_{n\times n} \\
&= [\overrightarrow{u}_1|\overrightarrow{u}_2|\dots|\overrightarrow{u}_m]\begin{bmatrix} s_1 & 0 & \dots \\ 0 & s_2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}\begin{bmatrix} \overrightarrow{v}_1 \\ - \\ \overrightarrow{v}_2 \\ - \\ \vdots \\ - \\ \overrightarrow{v}_n \end{bmatrix} \\
&= s_1\begin{bmatrix} | \\ u_1 \\ | \end{bmatrix}[v_1] + s_2\begin{bmatrix} | \\ u_2 \\ | \end{bmatrix}[v_2] + \dots \qquad (2.18) \\
&= \sum_i s_i\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}[v_i]\,, \qquad (2.19)
\end{aligned}$$

where the range of i will rely on the number of none zero singular values s_i. In the equation above, u_i and v_i^T are presented as $\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}$ and $[v_i]$, respectively, to tell apart the scale factor s_i. The name, rank one decomposition, is given due to the fact that each summand

$$s_i\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}[v_i] \qquad (2.20)$$

is a rank one matrix.

This equation illustrates that the eigenvector v_i is the base vector. The following will demonstrate that the coefficient $s_i\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}$ is the projection of X onto $\begin{bmatrix} | \\ v_i \\ | \end{bmatrix}$ scaled with s_i.

$$\because X = USV^T \tag{2.21}$$

$$XV = US \tag{2.22}$$

$$XV(1/S) = U \quad (\text{when } s_i \neq 0) \tag{2.23}$$

$$(1/S)XV = U \tag{2.24}$$

$$U = \sum_i \frac{1}{s_i} X \begin{bmatrix} | \\ v_i \\ | \end{bmatrix}, \tag{2.25}$$

$$\therefore \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} = \frac{1}{s_i} X \begin{bmatrix} | \\ v_i \\ | \end{bmatrix}. \tag{2.26}$$

Similarly,

$$\begin{bmatrix} | \\ v_i \\ | \end{bmatrix} = \frac{1}{s_i} X^T \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}. \tag{2.27}$$

If substitutes Eq. (2.19) with Eq. (2.26), we get

$$X = \sum_i s_i \frac{1}{s_i} X \begin{bmatrix} | \\ v_i \\ | \end{bmatrix} [v_i] \tag{2.28}$$

$$= \sum_i X \begin{bmatrix} | \\ v_i \\ | \end{bmatrix} [v_i] \tag{2.29}$$

$$= \sum_i \begin{bmatrix} | \\ y_i^* \\ | \end{bmatrix} [v_i]. \quad \left(\text{if } \begin{bmatrix} | \\ y_i^* \\ | \end{bmatrix} = X \begin{bmatrix} | \\ v_i \\ | \end{bmatrix}\right) \tag{2.30}$$

Likewise, substituting Eq. (2.19) with Eq. (2.27) results in:

$$X = \sum_i s_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \left[\frac{1}{s_i} X^T \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \right]^T \tag{2.31}$$

$$= \sum_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \left[X^T \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \right]^T \tag{2.32}$$

$$= \sum_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [y_i^{**}], \quad \left(\text{if } [y_i^{**}] = \left[X^T \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \right]^T\right) \tag{2.33}$$

which is called K-L expression. Eqs. (2.30) and (2.33) show that X is the linear combination of eigenvectors. The coefficients $\begin{bmatrix} | \\ y_i^* \\ | \end{bmatrix}$ (or $[y_i^{**}]$) is the projection of X (or X^T) onto the corresponding eigenvector.

If X needs to be compressed, a subset in the K-L expression can be neglected. Assume $i = m+1, ..., N-1$ are to be rejected, the approximation of X will become $\hat{X}$ as defined below:

$$\hat{X} = \sum_{i=1}^{m} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [y_i^{**}]\,. \tag{2.34}$$

The mean square error of $\hat{X}$ to X will be

$$\varepsilon = E[X - \hat{X}]^2. \tag{2.35}$$

The minimized error is given by:

$$\varepsilon_{min} = \sum_{i=m+1}^{N-1} \lambda_i, \tag{2.36}$$

where λ_i $(i = m+1, \ldots, N-1)$ are the smallest values in the eigenvalue matrix Λ. Appendix A illustrates how ε_{min} is derived by using the method of Lagrange multipliers.

Therefore, the main advantages of K-L transform can be summarized below:

1. The correlation of original signal X can be completely removed

2. During the compression of the data, the minimum mean square error after the truncation of $y_i^{**}, i = 1, \ldots, m$ equals to the remaining summation of the eigenvalues

3. Because $\lambda_0 \geqslant \lambda_1 \geqslant \ldots \geqslant \lambda_{N-1}$, after truncating $\lambda_{m+1}, \ldots, \lambda_{N-1}$, the rest $\lambda_0, \lambda_1, \ldots, \lambda_m$ remains the maximum energy of the data X.

2.2 Modified Discrete Cosine Transform (MDCT) for Sparse Representation

2.2.1 The Role of MDCT in the Studies

Due to the advantages of MDCT, namely half size frequency transformation and perfect reconstruction, it has been utilized by one of the Sparse Representation techniques, MMP, as a dictionary system. One of the studies is to investigate if the MMP is good for feature extraction for audio signal which is then used to construct a fingerprint for the audio identification for legacy content protection from the perspective of copyright protection.

2.2.2 Kaiser-Bessel Derived (KBD) Windowing

As we know, in order to analysis the signal, we usually transform the signal into frequency domain. In practice, we can not transform a signal with infinite duration, instead we use window to partition the signal into segments and implement the transformation on each segment. The procedure is the window signal will multiply the original signal in time domain. According to the Fourier transform, it is equivalent to the convolution of these two signals in frequency domain. This relationship can be presented as:

$$\mathscr{F}[\mathbb{w}(t)x(t)] = \mathbb{W}(f) * X(f), \tag{2.37}$$

where $\mathscr{F}$ presents the Fourier transform, $\mathbb{w}(t)$ is the window signal and $x(t)$ is the original signal.

Since the window signal is a time limited but frequency unlimited signal, no matter how high the sample rate is, the windowed signal in frequency domain always have aliasing because of the unlimitation of the window signal in frequency-domain. It means the energy leakage of the signal into other frequency. This creates the noise when the signal is inverse-transformed. If the window width (or size) in time domain is increased, its frequency spectrum will be compressed. The frequency response of rectangular window as an example

is demonstrated as follows:

$$\mathbb{W}_R(f) = \int_{-\infty}^{\infty} \mathbb{w}_R(t) e^{-j2\pi ft} dt = \int_0^T e^{-2\pi ft} dt = e^{-j\pi fT} \frac{sin(\pi fT)}{\pi f}, \tag{2.38}$$

where $\mathbb{w}_R$ and $\mathbb{W}_R$ present the time and frequency form of rectangular window respectively. Even though the width of the frequency spectrum is still unlimited, the frequency components other than the central frequency drops-off quickly, which makes the leakage smaller. When the width of the window in time domain goes to infinity, the frequency spectrum becomes a delta function. Any frequency domain signal convolves with a delta signal will remain the same shape. It states that if the window width expands to infinity, i.e., no windowing, there is no energy leakage.

Since we have to choose a window in the real world, which type of window is suitable for the application needs to be considered. The reason is under the restriction of having the same window size, the different window shape will give the different aliasing result. We will use Kaiser-Bessel (KB) and KBD window as examples because the KBD window is suitable to use with MDCT. Let's first give the formula of these two window functions. The time and frequency expressions of the KB window in discrete time are defined as [59]:

$$\mathbb{w}_{KB}(n) = \frac{I_0\left[\pi\alpha\sqrt{1.0 - (\frac{n-N/2}{N/2})^2}\right]}{I_0[\pi\alpha]}, \qquad for \qquad n = 0, ..., N; \tag{2.39}$$

$$\mathbb{W}_{KB}(f) = \frac{N}{I_0(\pi\alpha)} \frac{sinh\left[\sqrt{\pi^2\alpha^2 - (N2\pi f/2)^2}\right]}{\sqrt{\pi^2\alpha^2 - (N2\pi f/2)^2}}. \tag{2.40}$$

I_0 is the 0^{th} modified Bessel function, α is an arbitrary real number that determines the shape of the window, and $sinh$ is called hyperbolic sin that the points of $sinh(a)$ form the right half of the equilateral hyperbola as shown in 2.3. In the frequency domain, it determines the trade-off between main-lobe width and side lobe level, which is a central decision in window design.

The KBD window is the normalized version KB window. Its discrete time function is

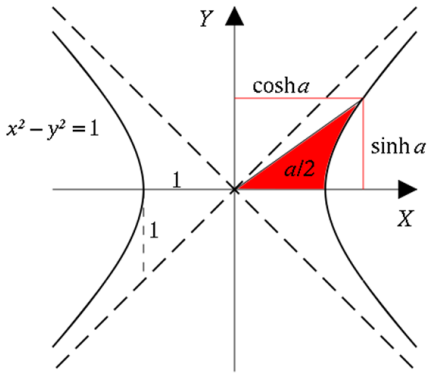


Figure 2.3: Curve graph of sinh function.

defined in terms of the KB window $\mathbb{w}_{KB}$ with length $N+1$, by the formula:

$$\mathbb{w}_{KBD}(n) = \begin{cases} \sqrt{\frac{\sum_{j=0}^{n} \mathbb{w}_{KB}(j)}{\sum_{j=0}^{\frac{N}{2}} \mathbb{w}_{KB}(j)}}, & \text{for } n = 0, ..., \frac{N}{2} - 1 \\ \sqrt{\frac{\sum_{j=n-N/2+1}^{\frac{N}{2}} \mathbb{w}_{KB}(j)}{\sum_{j=0}^{\frac{N}{2}} \mathbb{w}_{KB}(j)}}, & \text{for } n = \frac{N}{2}, ..., N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.41}$$

In the Figure 2.4, both KB and KBD windows have the same width of 128, with different α settings. The parameter α can be used to tune the window shape so that it has appropriate frequency resolution and leakage properties. The left hand side of the Figure 2.4 shows that, for the same type of window, the higher the α is the narrower the window and the less smooth the edges of the window will be. Accordingly, in the frequency domain, the higher the α is the wider the main lobe and the lower the side lobe level will be. Figure 2.4 also indicates that the different types of windows with the same value of α have the corresponding frequency responses according to their shapes in the time domain. Having the same value of α, the less smooth edges of the KBD window leads to the worse ultimate rejection, i.e., the amount of attenuation in the side lobes energy, than the ultimate rejection of the KB window. In other words, the KBD window creates more energy leakage than KB window. On the other hand, given the same value of α, the KBD window has wider average width than KB window, which leads to better frequency localization, i.e., its main lobe is narrower than the KB window.

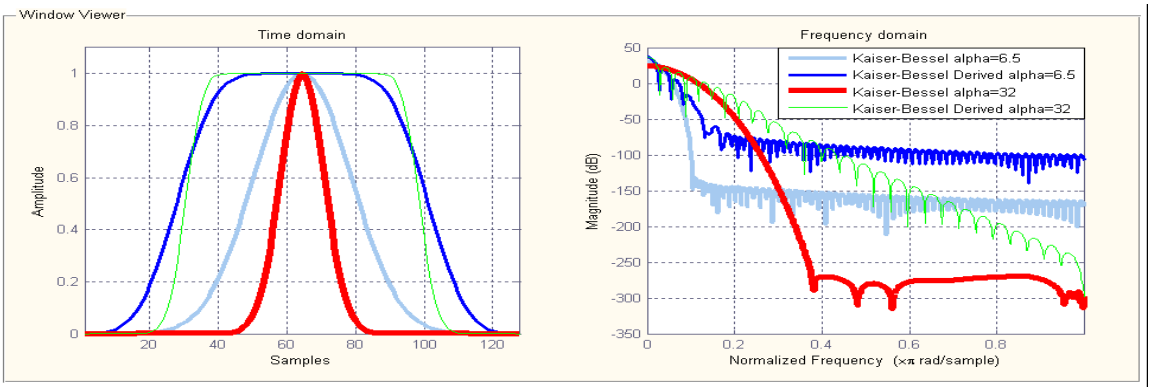


Figure 2.4: Windows comparison for the frequency resolution and leakage properties.

KBD window is normalized version of Kaiser-Bessel window and designed for the overlap-and-add approach as shown in Figure 2.5, which means its window shape follows the rule of:

$$\mathrm{w}^{i}_{KBD_a}[n] * \mathrm{w}^{i}_{KBD_s}[n] + \mathrm{w}^{i-1}_{KBD_a}[M+n] * \mathrm{w}^{i-1}_{KBD_s}[M+n] = 1, for\ n = 0, ..., \frac{N}{2} - 1 \quad (2.42)$$

where $M = N - M = \frac{N}{2}$ in this case; w_{KBD_a} and w_{KBD_s} present the analysis and synthesis KBD windows respectively; and i and $i-1$ stands for the current block and previous block. If we choose the identical analysis and synthesis windows, then the rule above will be simplified to:

$$\mathrm{w}^{i}_{KBD}[n]^2 + \mathrm{w}^{i-1}_{KBD}[M+n]^2 = 1, for\ n = 0, ..., \frac{N}{2} - 1. \quad (2.43)$$

This rule, $\mathrm{w}^{i}[n]^2 + \mathrm{w}^{i-1}[M+n]^2 = 1$, is called Princen-Bradley condition. Compared to the frequency localization, the leakage problem becomes trivial, therefore we choose KBD window with $\alpha = 6.5$ in the study. The value gives a good balance between the frequency resolution and the leakage.

Since KDB window follows the Princen-Bradley condition, it can be utilized in MDCT so that MDCT uses two times of the window: analysis and synthesis, and can still perfectly reconstruct the signal.

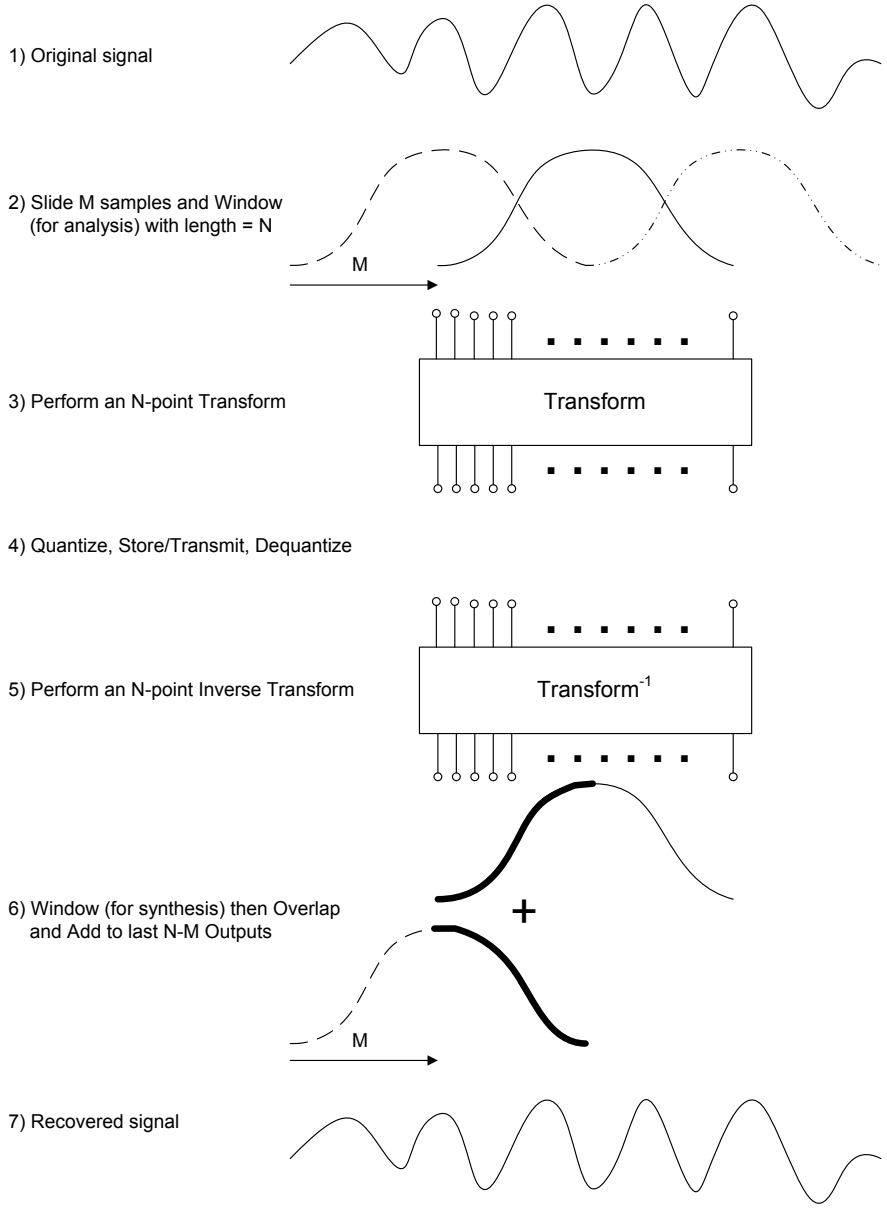


Figure 2.5: Schematic of the window and overlap-and-add approach utilized to encode-decode signal

2.2.3 Superiority of MDCT/Inverse-MDCT (IMDCT) Transformations

As mentioned in the previous section, we can develop a good frequency representation of a signal by wisely taking finite length blocks of time-sampled data and transforming the data into a finite length of discrete frequency domain samples. However, implementing the overlap-and-add procedure increases the data rate of the frequency-domain signal prior to any coding gains. With 50% overlap between adjoining blocks we end up doubling the data rate. MDCT is an alternative transform to the Discrete Fourier Transform (DFT) to deal with the problem by allowing a 50% overlap between blocks without increasing the data rate. In other words, for a real-valued signal, only half of the N frequency-domain samples from an N-point MDCT are independent indicating that the transform for such signal only requires $N/2$ frequency samples from each data block for full signal recovery. The MDCT transform can be defined as:

$$X(k) = \sum_{m=0}^{N-1} \mathrm{w}^a(m)x(m)\cos(\frac{\pi}{2N}(2m+1+\frac{N}{2})(2k+1)),$$

$$for\ k = 0 \sim \frac{N}{2} - 1, \quad (2.44)$$

where $\cos(\frac{\pi}{2N}(2m+1+\frac{N}{2})(2k+1))$ is the kernel of the transformation, $x(m)$ represents a signal with length N, $X(k)$ is correspondingly the transformed signal, w^a denotes analysis window, and m and k are the index of time and angular frequency, respectively. If we set the signal length N to be 16, there will be total $m = 1 \sim 16$ cosine series in time domain and each one of them is corresponding to 16 different angular frequencies; but the first eight ones $k = 1 \sim 8$ shown in Figure 2.6 and the second eight ones $k = 9 \sim 16$ shown in Figure 2.7 are the same but opposite in sign. In other word, only 8 of cosine series in this case are independent. Therefore, IMDCT only considers $k = 1 \sim 8$ frequencies, which will be presented later.

Given $N = 16$, the characteristics of MDCT bases can be summarized as:

1. only 8 orthogonal bases among 16 ones; Within each cosine base, the left 8 time sequences are antisymmetric and the right 8 time sequences are symmetric around the

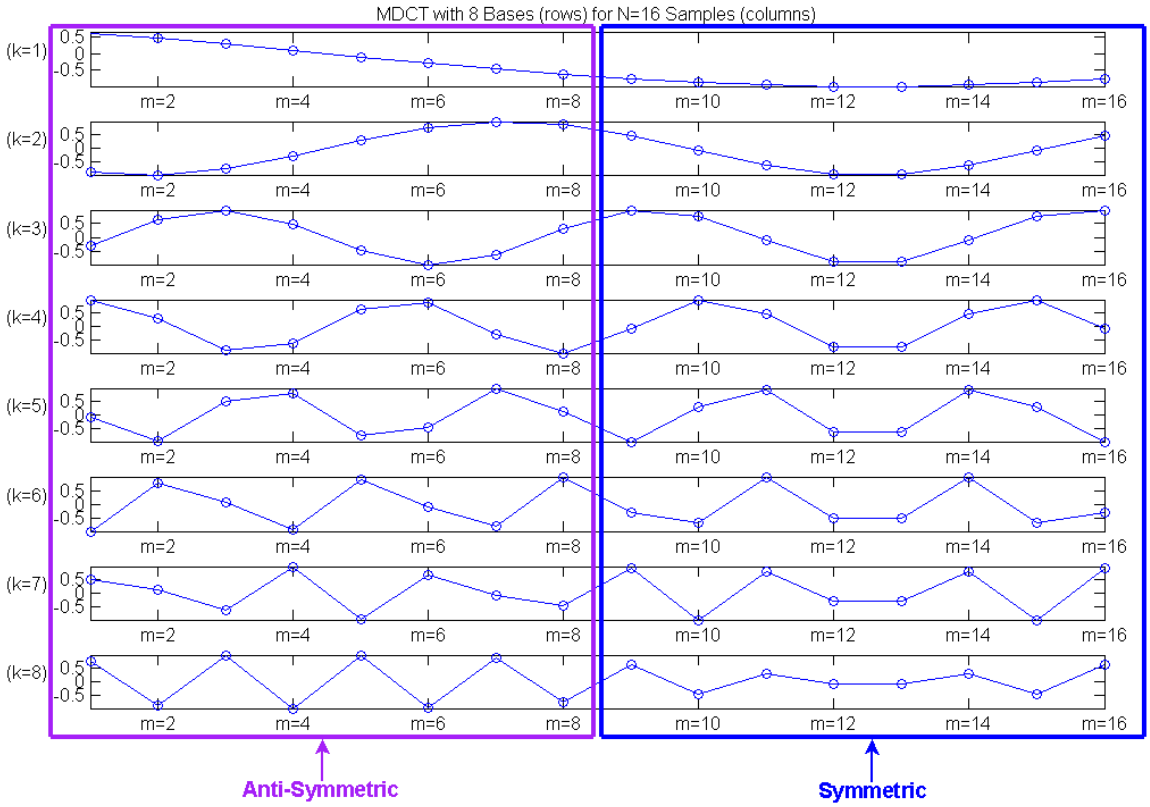


Figure 2.6: MDCT bases with length N=16 for frequencies k=1∼8

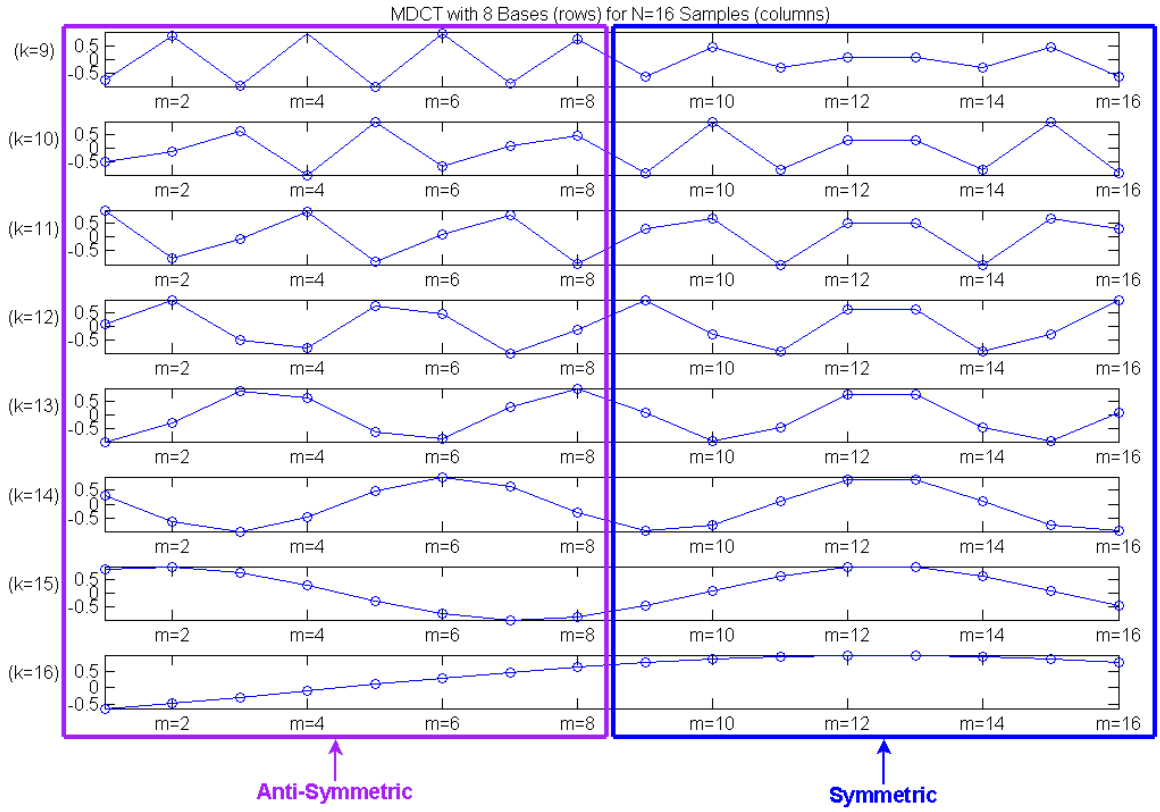


Figure 2.7: MDCT bases with length N=16 for frequencies k=9∼16

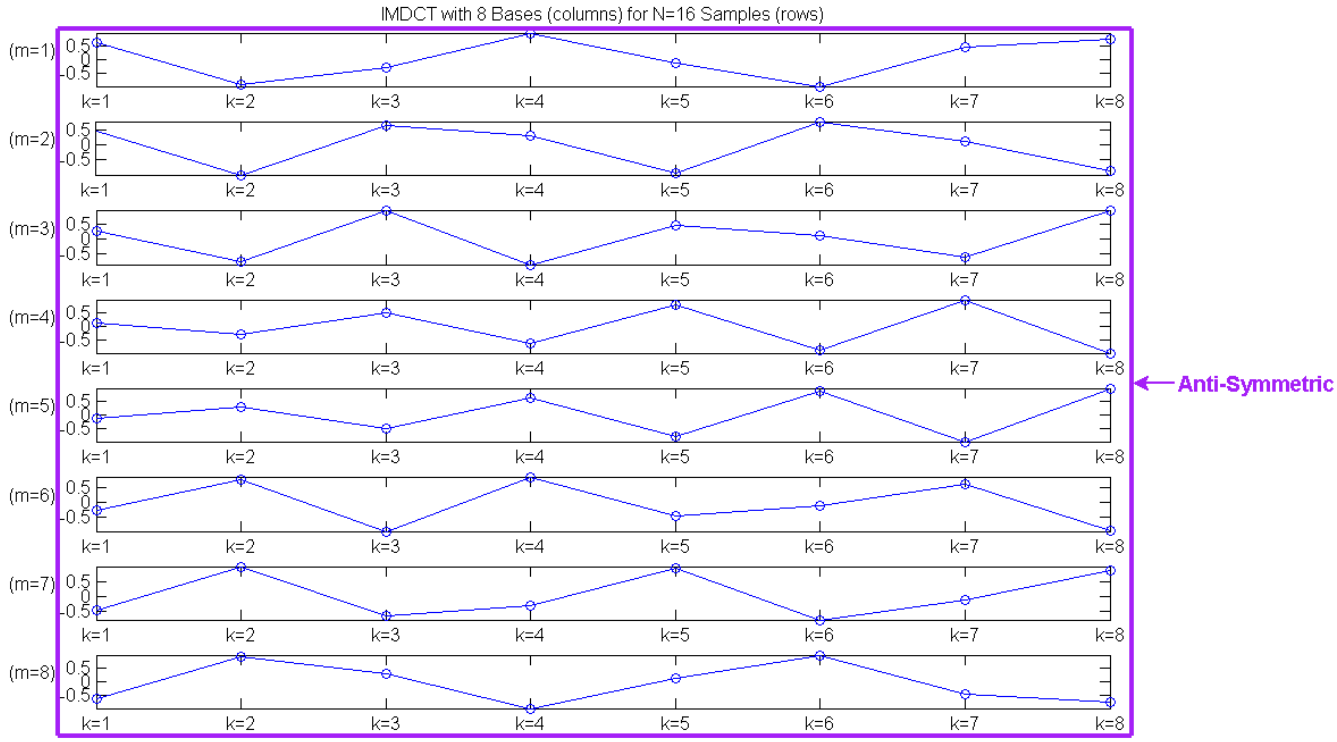


Figure 2.8: IMDCT bases with frequencies k=1∼8 for samples m=1∼8

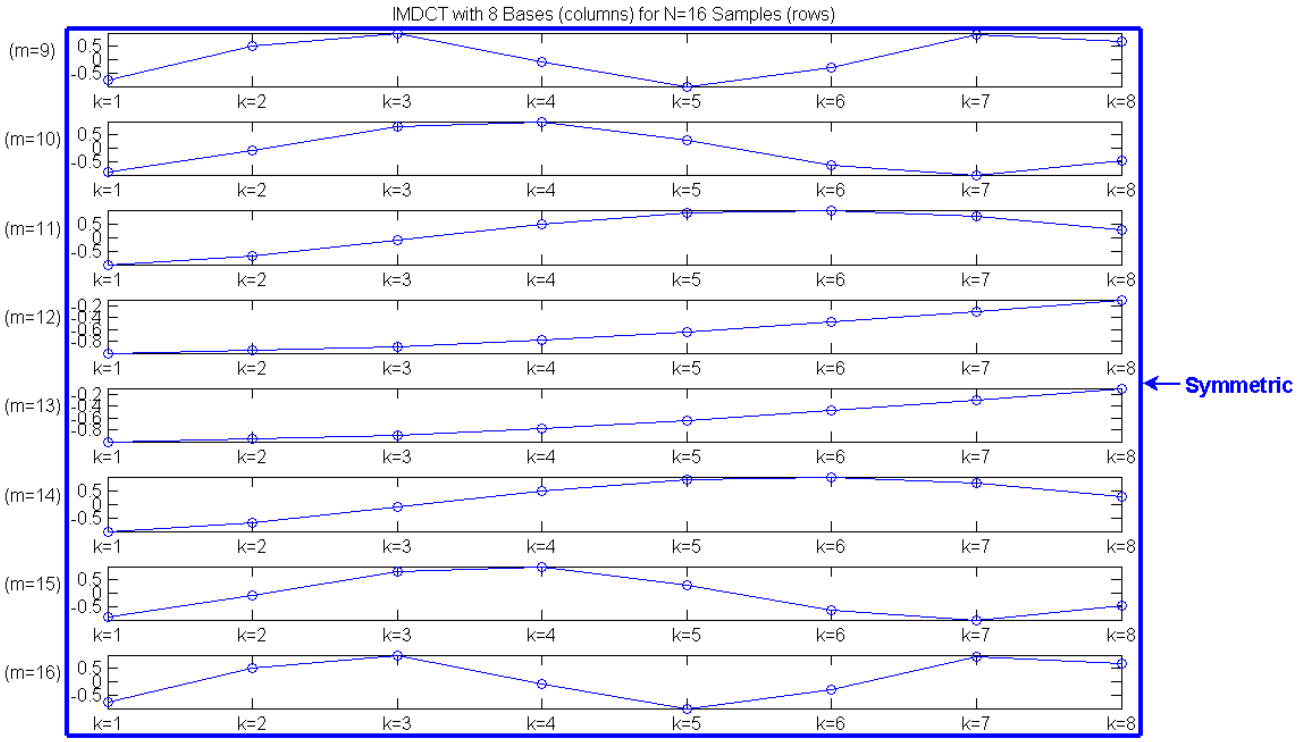


Figure 2.9: IMDCT bases with frequencies k=1∼8 for samples m=9∼16

center as indicated in Figures 2.6 and 2.7. We define the antisymmetric series as matrix An and the symmetric series as matrix Sy

2. within each base, the summation of the power of each point in the symmetric side and the corresponding point in the antisymmetric side equal to 1, e.g. $\cos^2(\mathrm{m}=1,\mathrm{k}=1)+\cos^2(\mathrm{m}=9,\mathrm{k}=1)=1$.

Accordingly, the IMDCT transform is given by the equation

$$x'(m)=\mathrm{w}^s(m)\frac{4}{N}\sum_{k=0}^{N/2-1}X(k)\cos(\frac{\pi}{2N}(2m+1+\frac{N}{2})(2k+1)),$$
$$for\ m=0\sim N-1, \quad (2.45)$$

where $\frac{4}{N}$ is the normalization coefficient, the transform kernel - $\cos(\frac{\pi}{2N}(2m+1+\frac{N}{2})(2k+1))$ remains the same, and w^s indicates the synthesis window. This equation indicates that IMDCT only take into account half of the frequencies, i.e., $k=1\sim 8$ if $N=16$. The IMDCT bases example for $k=1\sim 8$ are displayed in the Figures 2.8, 2.9.

Similarly, IMDCT for $N=16$ also has some characteristics:

1. the bases in Figure 2.8 is the transform of the bases in An and we name it An^T; the bases in Figure 2.9 is the transform of the bases in Sy and we name it Sy^T

2. each base in An^T is orthogonal to all the others in both An^T and Sy^T except its peer in An^T; each base in Sy^T is orthogonal to all the others in both Sy^T and An^T except its peer in Sy^T; for example, $\cos(\mathrm{m}=1,\mathrm{k}=1\sim 8)\not\perp\cos(\mathrm{m}=8,\mathrm{k}=1\sim 8)$, but $\cos(\mathrm{m}=1,\mathrm{k}=1\sim 8)\perp\cos(\mathrm{m}=2\sim 7,9\sim 16,\mathrm{k}=1\sim 8)$

3. the inner product of the bases itself in An^T and the inner product of the corresponding peer bases in Sy^T are equal to each other and the value is 4

4. the inner product of the peer bases in An^T and the inner product of the corresponding peer bases in Sy^T are equal to each other but opposite in sign, i.e., $\cos(\mathrm{m}=1,\mathrm{k}=1\sim 8)\times\cos(\mathrm{m}=8,\mathrm{k}=1\sim 8)+\cos(\mathrm{m}=9,\mathrm{k}=1\sim 8)\times\cos(\mathrm{m}=16,\mathrm{k}=1\sim 8)=0$.

$$
\begin{aligned}
MDCT_{N=16} &= \begin{matrix} \\ k_1 \\ k_2 \\ \vdots \\ k_8 \end{matrix}
\begin{matrix} \begin{matrix} m_1 & m_2 & \cdots & m_8 & m_9 & m_{10} & \cdots & m_{16} \end{matrix} \\
\begin{bmatrix} \cos(k_1 m_1) & \cos(k_1 m_2) & \cdots & \cos(k_1 m_8) & \cos(k_1 m_9) & \cos(k_1 m_{10}) & \cdots & \cos(k_1 m_{16}) \\ \cos(k_2 m_1) & \cos(k_2 m_2) & \cdots & \cos(k_2 m_8) & \cos(k_2 m_9) & \cos(k_2 m_{10}) & \cdots & \cos(k_2 m_{16}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \cos(k_8 m_1) & \cos(k_8 m_2) & \cdots & \cos(k_8 m_8) & \cos(k_8 m_9) & \cos(k_8 m_{10}) & \cdots & \cos(k_8 m_{16}) \end{bmatrix} \end{matrix}
\times \begin{bmatrix} x(m_1)\mathbb{w}^a(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x(m_{16})\mathbb{w}^a(16) \end{bmatrix} \\
&= [An \quad Sy] \times \begin{bmatrix} x(m_1)\mathbb{w}^a(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x(m_{16})\mathbb{w}^a(16) \end{bmatrix} \\
&= \begin{bmatrix} X(k_1) \\ X(k_2) \\ \vdots \\ X(k_8) \end{bmatrix} \\
&= X(\boldsymbol{k})
\end{aligned}
$$

Figure 2.10: MDCT for N=16

The signals $x'(m)$ can be obtained by the following procedures in the Figures 2.10, 2.11 and 2.12.

If $x^{'}(m_1), x^{'}(m_1), ..., x^{'}(m_8)$ represent the left side of current ith window samples $x_{i,L,1}$, $x_{i,L,2}$, ..., $x_{i,L,8}$, and $x^{'}(m_9), x^{'}(m_{10}), ..., x^{'}(m_{16})$ represent the right side of previous $(i-1)$th window samples $x_{i-1,R,1}, x_{i-1,R,2}, ..., x_{i-1,R,8}$ respectively, due to the features of the MDCT, IMDCT transforms and normalized version of Kaiser-Bessel window, the pair-wise summation will perfectly reconstruct the overlapped samples as shown in the Figure 2.12.

2.3 Wavelet Transform and Discrete Wavelet Transform (DWT)

2.3.1 The Role of DWT in the Studies

Since DWT is a very good tool to obtain multi-resolution of 1-D and 2-D data, it is possible that the data in large size can be quickly compressed into small size without loss of the main semantic feature. Besides, DWT uses the bases that are more sparse to present percussive signal because they can be similar to impulse signal.

In the studies, DWT was mainly involved in the approach which is described in Chapter 4. In this approach, DWT was utilized to reduce the size of the image so that the very key information of the image is remained. This preparation step is very important because not only it does meet the requirement of the application, but also make the following Principal Component Analysis (PCA) calculation faster. The approach in Chapter 6 applies the DWT

$$
\begin{aligned}
IMDCT_{N=16} &= \frac{4}{N}\times\begin{bmatrix} w^s(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^s(16) \end{bmatrix}\times \begin{matrix} m_1 \\ m_2 \\ \vdots \\ m_8 \\ m_9 \\ m_{10} \\ \vdots \\ m_{16} \end{matrix} \begin{matrix} k_1 & k_2 & \cdots & k_8 \end{matrix} \\
&\quad \begin{bmatrix} \cos(k_1m_1) & \cos(k_2m_1) & \cdots & \cos(k_8m_1) \\ \cos(k_1m_2) & \cos(k_2m_2) & \cdots & \cos(k_8m_2) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(k_1m_8) & \cos(k_2m_8) & \cdots & \cos(k_8m_8) \\ \cos(k_1m_9) & \cos(k_2m_9) & \cdots & \cos(k_8m_9) \\ \cos(k_1m_{10}) & \cos(k_2m_{10}) & \cdots & \cos(k_8m_{10}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(k_1m_{16}) & \cos(k_2m_{16}) & \cdots & \cos(k_8m_{16}) \end{bmatrix}\times X(\boldsymbol{k}) \\
&= \frac{4}{N}\times\begin{bmatrix} w^s(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^s(16) \end{bmatrix}\times\begin{bmatrix} An^T \\ Sy^T \end{bmatrix}\times\begin{bmatrix} X(k_1) \\ X(k_2) \\ \vdots \\ X(k_8) \end{bmatrix} \\
&= \frac{4}{N}\times\begin{bmatrix} w^s(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^s(16) \end{bmatrix}\times\begin{bmatrix} An^T \\ Sy^T \end{bmatrix}\times [An \quad Sy]\times\begin{bmatrix} x(m_1)w^a(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x(m_{16})w^a(16) \end{bmatrix} \\
&= \begin{bmatrix} x'(m_1) \\ x'(m_2) \\ \vdots \\ x'(m_8) \\ x'(m_9) \\ x'(m_{10}) \\ \vdots \\ x'(m_{16}) \end{bmatrix} \\
&= x'(\boldsymbol{m})
\end{aligned}
$$

Figure 2.11: IMDCT for N=16

$$
\begin{bmatrix} x'(m_1)+x'(m_9) \\ x'(m_2)+x'(m_{10}) \\ \vdots \\ x'(m_8)+x'(m_{16}) \end{bmatrix} = \begin{bmatrix} x_{i,L,1}+x_{i-1,R,1} \\ x_{i,L,2}+x_{i-1,R,2} \\ \vdots \\ x_{i,L,8}+x_{i-1,R,8} \end{bmatrix} = \begin{bmatrix} x(overlap_1) \\ x(overlap_2) \\ \vdots \\ x(overlap_8) \end{bmatrix}
$$

Figure 2.12: Data Recovery

as part of the redundant dictionary system to capture the percussive signal, but further investigation needs to be done in the next stage.

2.3.2 Time-frequency Analysis

In order to process signals in different applications, different domain transformation methods has been developed, such as Fourier Transform, Short Time Fourier Transform (STFT), Wavelet Transform. Fourier Transform explores the frequency components of a signal but lacks time information. Short Time Fourier Analysis (was initiated by Gabor in 1946), segments the signal by window function and then using Fourier Transform to analysis each segmentation. The STFT of a signal $x(t)$ screened by a window function $\mathbb{w}(t)$ is defined as follows:

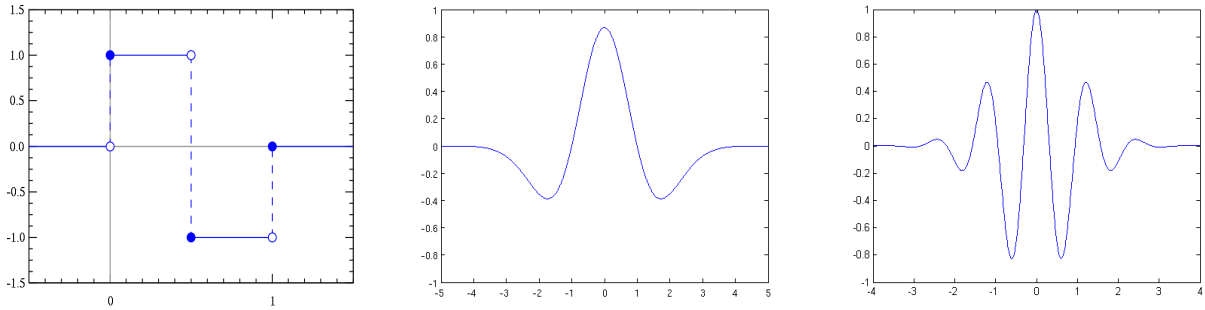
$$STFT(f,\tau) = \int_{-\infty}^{\infty} x(t)\mathbb{w}(t-\tau)e^{-j2\pi ft}dt \tag{2.46}$$

where τ is the shifting in time. In the frequency domain, $STFT(f,\tau)$ can be recognized as the convolution of $X(f)$ with the Fourier Transform of $\mathbb{w}(t-\tau)$ which is $e^{-j2\pi f\tau}\mathbb{W}(f)$. Even though STFT presents both the frequency and time information, the length of the segmentation (time information) is fixed which makes the analysis inadaptable to the non-stationary signal. According to the Heisenberg explanation in [60][61], it is impossible to obtain a high resolution for time and frequency at the same time. The trade-off between time resolution and frequency resolution is: if we use a window of length T then we have a time-resolution of T but our frequency resolution is $1/T$. Therefore, when the length of the segmentation is relative long, the transform method provides good frequency resolution by sacrificing the time resolution; when the length is relative short, the transform more precisely presents the time but less precisely in frequency.

The Continuous Wavelet Transform (CWT) is able to solve both time and scale (frequency) events better than the STFT by involving the scale parameter into the transformation basis as defined in the following formula:

$$C(s,\tau) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\boldsymbol{w}(\frac{t-\tau}{s})dt \tag{2.47}$$

The function $w(t)$ is called the *mother wavelet*. It is taken to be a "small wave". For example, the Haar wavelet is a single cycle of the square wave of period 1. There are several other wavelet, such as Mexican hat wavelet, Morlet wavelet as shown in the Figure 2.13. The graph of $w(\frac{t-\tau}{s})$ is obtained by stretching the graph of $w(t)$ by the factor s, called the scale, and shifting in time by τ, called the translation. The time-shifted and time-scaled wavelet is sometimes called a *baby wavelet*.



(a) Haar (b) Mexican Hat (c) Morlet

Figure 2.13: Examples of Wavelets.

The CWT is the inner product or cross correlation of the signal $x(t)$ with the scaled and time shifted wavelet $\frac{1}{\sqrt{s}}w(\frac{t-\tau}{s})$. This cross correlation is a measure of the similarity between signal and the scaled and shifted wavelet. For a fixed scale, s, the CWT is the convolution of the signal $x(t)$ with the time reversed wavelet $\frac{1}{\sqrt{s}}w(\frac{-t}{s})$. That is, the CWT is the output when a signal is fed to the filter with impulse response $\frac{1}{\sqrt{s}}w(\frac{-t}{s})$ as shown in the Figure 2.14.

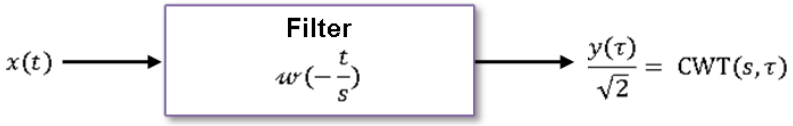


Figure 2.14: Wavelet Filtering

Similarly, STFT also can be interpreted as a signal $x(t)$ pass through a filter system. In the equation below, we rewrite the STFT as:

$$STFT(\zeta,\tau) = \int_{-\infty}^{\infty} x(t)\mathrm{w}(t-\tau)e^{-j2\pi\zeta t}dt \quad = e^{-j2\pi\zeta\tau}\int_{-\infty}^{\infty} x(t)\mathrm{w}(t-\tau)e^{-j2\pi\zeta(t-\tau)}dt \quad (2.48)$$

Notice that the term $\int_{-\infty}^{\infty} x(t)\mathrm{w}(t-\tau)e^{-j2\pi\zeta(t-\tau)}dt$ is the convolution of the signal, $x(t)$, with the frequency shifted and time reversed window function, $e^{j2\pi\zeta t}\mathrm{w}(-t)$ as denoted in Figure 2.15.

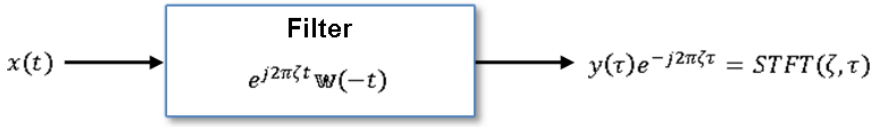
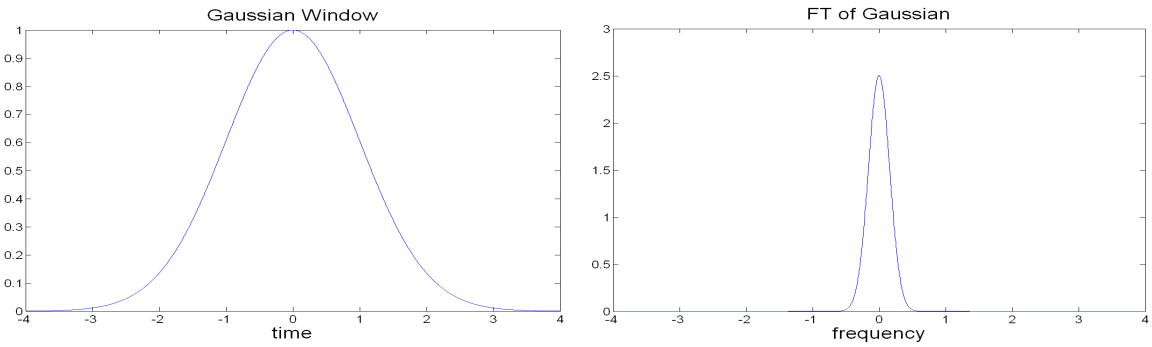


Figure 2.15: STFT Filtering

To better understand the significance of the filter interpretations of CWT and STFT, we can consider the case of the Morlet wavelet, $w(t) = e^{\frac{-t^2}{2}}\cos(5t)$, and the STFT with Gaussian window function, $\mathrm{w}_G(t) = e^{\frac{-t^2}{2}}$.

The Fourier Transform of the Gaussian window function is: $\mathbb{W}_G(f) = \sqrt{2\pi}e^{\frac{-(2\pi f)^2}{2}}$. Figure 2.16 show the graphs according to the function $\mathrm{w}_G(t)$ and $\mathbb{W}_G(f)$, respectively. Need to note that the window function in frequency domain is a low pass filter which blocks all frequencies above $f = \frac{3}{2\pi} \approx 0.5Hz$.



(a) (b)

Figure 2.16: Gaussian window and its Fourier Transform.

Therefore, the frequency response of the filter in the STFT is this transform shifted by frequency ζ, which is $\mathbb{W}_G(f-\zeta)$. This is a band pass filter centered at frequency ζ with bandwidth 1 Hz.

In the case of CWT, the frequency response of the filter when the scale $s = 1$, is:

$(1/2)(\mathbb{W}_G(f-5/(2\pi))+\mathbb{W}_G(f+(5/2\pi)))$. This is a band pass filter centered at frequency $5/2\pi \approx 0.8Hz$ with bandwidth 1 Hz. Accordingly, at scale s, the frequency response is $(1/2)(\mathbb{W}_G(sf-5/(2\pi s))+\mathbb{W}_G(sf+5/(2\pi s)))$. This is band pass filter centered at frequency $5/(2\pi s)$ with a bandwidth of $1/s$ Hz. This shows that the essential difference between the STFT and the CWT. In the STFT the frequency bands have a fixed width (1 Hz for Gaussian). In the CWT the frequency bands grow and shrink with the frequency (scale) being used. This allows good frequency resolution at low frequencies and good time resolution at high frequencies.

2.3.3 DWT - Analysis and Synthesis

Instead of computing STFT (f,τ) for all frequency f and all time shift τ, we can restrict the calculation to $f_n = n/T$ and $\tau_m = mT$. Therefore,

$$STFT(n/T, mT) =< x(t), \upsilon_{n,m}(t) > \tag{2.49}$$

where

$$\upsilon_{n,m}(t) = e^{j2\pi n/T}\mathbb{w}(t-mT). \tag{2.50}$$

Since $\upsilon_{n,m}$ is non-zero only for $mT \leq t \leq (m+1)T$, it is clear that these are orthogonal functions. Notice that T is the width of the window box $\mathbb{w}$.

Along the same line, we can get an orthogonal basis of functions in the CWT case by choosing the scales to be powers of 2 and the shifting times to be an integer multiple of the scales. That is, for integers j and $\varkappa$, the wavelet coefficient will be:

$$C(1/2^j, \varkappa/2^j) = 2^{j/2}\int_{-\infty}^{\infty} x(t)w(2^j t - \varkappa) \tag{2.51}$$

where the $1/2^j$ and $\varkappa/2^j$ are the scales and the shifting times, respectively. Correspondingly, the set of baby wavelet is defined as follows:

$$w_{j,\varkappa} = 2^{j/2}w(2^j t - \varkappa) \tag{2.52}$$

So that the values $C(1/2^j, \varkappa/2^j)$ are the analysis coefficients for the baby wavelet functions and can be rewritten as:

$$C(1/2^j, \varkappa/2^j) =< x(t), w_{j,\varkappa}(t) > \tag{2.53}$$

Notice that the set of baby wavelets is an orthogonal basis.

In the case of an orthogonal wavelet, the analysis formula is called the *Discrete Wavelet Transform*.

$$DWT(Analysis) : c_{j,\varkappa} = \int_{-\infty}^{\infty} x(t) w_{j,\varkappa}(t) dt \tag{2.54}$$

The recovery of the signal through the synthesis formula is called the *Inverse Discrete Wavelet Transform*.

$$IDWT(Synthesis) : x(t) = \sum_{j} \sum_{\varkappa} c_{j,\varkappa} w_{j,\varkappa} \tag{2.55}$$

Figure 2.17, called time-scale diagram, illustrates the orthogonality. In this figure, the baby wavelets at different scales are shown on the right side and their corresponding time shifting positions are shown as a tree structure on the left. The time-scale diagram for DWT is a set of samples of the time-scale diagram for the CWT. The samples are quite "sparse" for large scale and more "dense" for small scale. There are number of such orthogonal wavelets and the simplest of these is the Haar wavelet.

2.3.4 Multi-resolution Analysis

This section will describe how the orthogonal baby wavelets can give rise to a *Multiresolution Analysis*. Given the signal $x_j(t)$ is in the space $\mathcal{W}_j$, the DWT synthesis formula can be expanded as follows:

$$x(t) = \sum_{j=-\infty}^{\infty} x_j(t) \quad where \quad x_j(t) = \sum_{\varkappa=-\infty}^{\infty} c_{j,\varkappa} w_{j,\varkappa}(t), \tag{2.56}$$

There can be another expression way, that is, the space V_j is defined as the set of all signals, $x(t)$, which can be synthesized from the baby wavelets $w_{i,\varkappa}(t)$ where $i < j$ and $-\infty < \varkappa \leq \infty$. That is

$$x(t) = \sum_{i=-\infty}^{j-1} \sum_{\varkappa} c_{i,\varkappa} w_{i,\varkappa}(t). \tag{2.57}$$

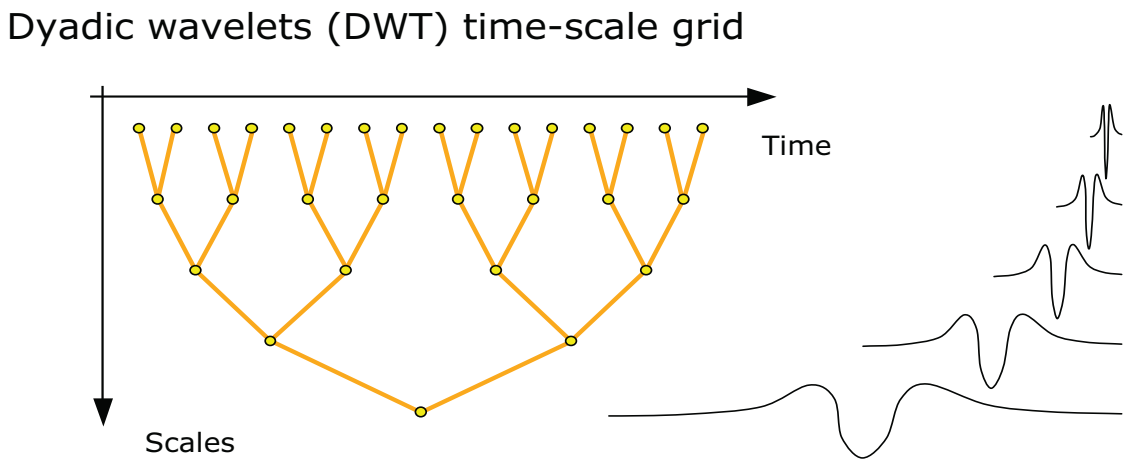


Figure 2.17: Time-scale diagram for the Discrete Wavelet Transform

Every signal in space V_j is a summation of a signal in V_{j-1} and $\mathcal{W}_{j-1}$ because the Eq. (2.57) can be further expanded as:

$$x(t) = \sum_{i=-\infty}^{j-1} \sum_{\varkappa} c_{i,\varkappa} w_{i,\varkappa}(t) = \sum_{i=-\infty}^{j-2} \sum_{\varkappa} c_{i,\varkappa} w_{i,\varkappa}(t) + \sum_{\varkappa} c_{j-1,\varkappa} w_{j-1,\varkappa}(t) \tag{2.58}$$

This shows that the spaces $\mathcal{W}_{j-1}$ are the differences between adjacent spaces $\mathcal{V}_j$ and $\mathcal{V}_{j-1}$. It also tells that the spaces V_j are nested inside each other:

$$\{0\} \subset \ldots \subset V_{-3} \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset V_3 \ldots \subset L^2. \tag{2.59}$$

For j goes to infinity, V_j enlarges to become all energy signals (L^2); in contrast, for j goes to negative infinity, V_j shrinks down to the zero signal.

Table 2.1: Resolution Change Along With the Change of Level and Scale

Level i	10	9	...	2	1	0	-1	-2
Scale	1024	512	...	4	2	1	1/2	1/4
Resolution	$1/2^{10}$	$1/2^9$	...	1/4	1/2	1	2	4

Similarly, every signal $x(t)$ in space V_j can be iteratively unfolded as follows:

$$\begin{aligned} \mathcal{V}_j &= \mathcal{V}_{j-1} + \mathcal{W}_{j-1} \qquad (2.60) \\ &= \mathcal{V}_{j-2} + \mathcal{W}_{j-2} + \mathcal{W}_{j-1} \\ &= \mathcal{V}_{j-3} + \mathcal{W}_{j-3} + \mathcal{W}_{j-3} + \mathcal{W}_{j-1} \\ &= \vdots \\ &= \mathcal{V}_{j-n} + \sum_{i=j-n}^{j-1} \mathcal{W}_i \qquad (2.61) \end{aligned}$$

This leads to various decompositions:

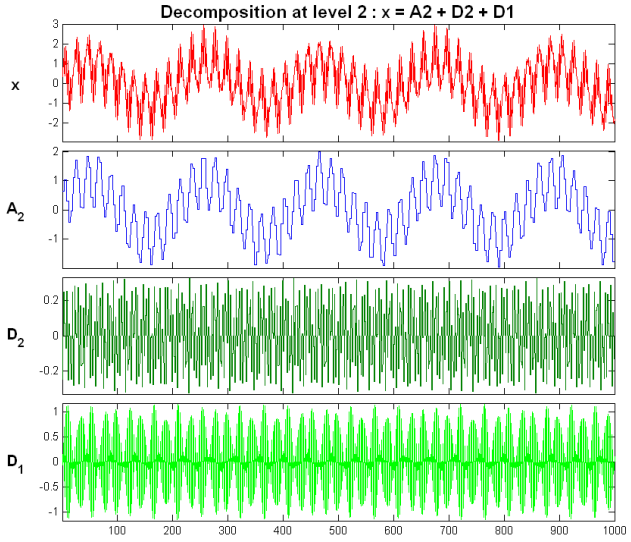
$$\begin{aligned} x(t) &= \mathbb{A}_1(t) + \mathbb{D}_1(t) \qquad (2.62) \\ &= \mathbb{A}_2(t) + \mathbb{D}_2(t) + \mathbb{D}_1(t) \\ &= \mathbb{A}_3(t) + \mathbb{D}_3(t) + \mathbb{D}_2(t) + \mathbb{D}_1(t) \end{aligned}$$

where $\mathbb{D}_i(t)$, in $\mathcal{W}_{j-i}$, is called the detail at level i and $\mathbb{A}_i(t)$, in $\mathcal{V}_{j-i}$, is called the approximation at level i. Table 2.1 indicates that as i goes larger, the approximation $\mathbb{A}_i(t)$ of the signal $x(t)$ becomes smoother (lower resolution of original signal) and its size is smaller. Therefore, the different size of the compressed signal can be obtained for different applications.

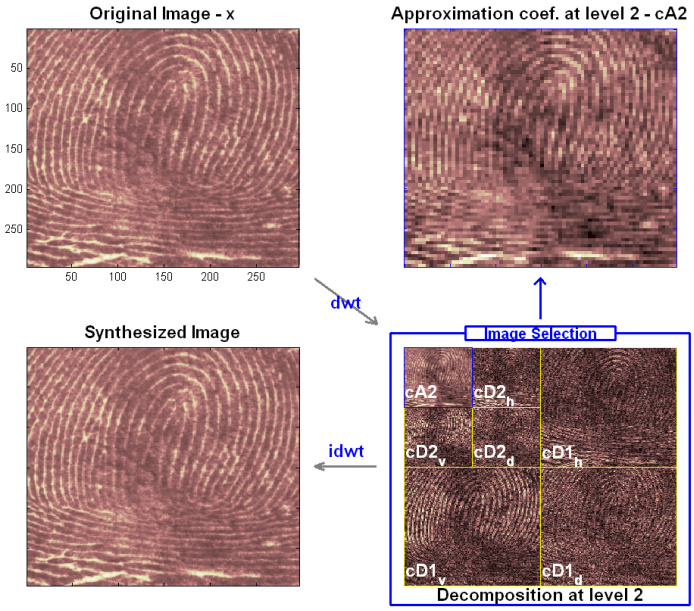
Figures 2.18(a) and 2.18(b) depict the decomposed signals at level 2 in 1 dimension and in 2 dimensions, respectively.

2.3.5 Scaling Function and Filter Banks

Investigation of the multiresolution analysis leads to a *scaling function*, a pair of *discrete time filters*, and a *perfect reconstruction filter bank* which can be used to calculate the DWT quickly.



(a) x is a 1D signal - the sum of sines



(b) x is a 2D signal - finger

Figure 2.18: The decomposed signals at level 2.

In order to express the multiresolution subspaces $\mathcal{V}_j$ directly, a scaling function $\phi(t)$ is introduced and the *baby scaling functions* is defined as:

$$\phi_{j,\varkappa}(t) = \sqrt{2^j}\phi(2^j t - \varkappa) \tag{2.63}$$

where $-\infty < j < \infty$ and $-\infty < \varkappa < \infty$. Just as for the wavelet, the "scale" of $\phi_{j,\varkappa}(t)$ is $1/2^j$ and the "shifting position" is $\varkappa/2^j$. Notice that $\phi(t - \varkappa)$ correlates to the signal in $\mathcal{V}_0$. That is, if $\phi(t)$ can be found, the signal in $\mathcal{V}_0$ can be synthesized from the integer translates $\phi(t - \varkappa)$ of the scaling function. Accordingly, the spaces $\mathcal{V}_j$ can be derived from $\mathcal{V}_0$ by time compression or dilation by powers of 2.

There are *discrete time filter* coefficients $h_0(n)$ such that:

$$\phi(t) = \sum_n h_0(n)\sqrt{2}\phi(2t - n) \tag{2.64}$$

which connects the scaling function to itself at two different time scales. This formula (2.64) is called the *Two Scale Equation*. Since $\mathcal{W}_0$ is also a subset of $\mathcal{V}_1$, there is another two scale equation for the wavelet which gives rise to another filter $h_1(n)$, such that:

$$w(t) = \sum_n h_1(n)\sqrt{2}\phi(2t - n) \tag{2.65}$$

Since the spaces $\mathcal{V}_j$ are getting larger and larger as j goes to ∞, Any signal, $x(t)$, can be approximated by choosing a large enough value of $j = J$ and projecting the signal into $\mathcal{V}_J$ using the basis $\phi_{J,\varkappa}$. That is

$$cA_0(\varkappa) = \int_{-\infty}^{\infty} x(t)\phi_{J,\varkappa}(t)dt \tag{2.66}$$

From these we can approximately recover the signal as:

$$x(t) \approx \sum_{\varkappa} cA_0(\varkappa)\phi_{J,\varkappa}(t) \tag{2.67}$$

In effect, we replace the signal, $x(t)$, by the approximate signal given by the projection coefficients, $cA_0(\varkappa)$.

Since $\mathcal{V}_J = \mathcal{W}_{J-1} + \mathcal{V}_{J-1}$, Eq. (2.67) can be further expressed as:

$$\begin{aligned} x(t) &= \sum_{\varkappa} cA_0(\varkappa)\phi_{J,\varkappa}(t) \\ &= \sum_{\varkappa} cA_1(\varkappa)\phi_{J-1,\varkappa}(t) + \sum_{\varkappa} cD_1(\varkappa)w_{J-1,\varkappa}(t) \\ &= \mathbb{A}_1(t) + \mathbb{D}_1(t) \end{aligned} \tag{2.68}$$

where the basis $w_{J-1,\varkappa}(t)$ in $\mathcal{W}_{J-1}$ and $\phi_{J-1,\varkappa}(t)$ in $\mathcal{V}_{J-1}$. The coefficients $cA_1(\varkappa)$ and $cD_1(\varkappa)$ are called the approximation coefficients and the detail coefficients at level 1. As before, the signals $\mathbb{A}_1(t)$ and $\mathbb{D}_1(t)$ are called the approximation and detail at level 1.

From the equation above, it can be derived that the signal can be decomposed using the bases $\phi_{j,\varkappa}(t)$ and $w_{j,\varkappa}$. Therefore, as long as the approximation coefficients and detail coefficients are available, the signal in any resolution level can be obtained. These coefficients at many different scales can be computed through a filter bank.

As the wavelets and the scales at each index level are orthogonal, we can compute the coefficients $cA_1(\varkappa)$ and $cD_1(\varkappa)$ by the inner product formulae:

$$\begin{aligned} cA_1(\varkappa) &= < x(t), \phi_{j-1,\varkappa}(t) > \\ &= < \sum_n cA_0(n)\phi_{j,n}(t), \phi_{j-1,\varkappa}(t) > \\ &= \sum_n cA_0(n) < \phi_{j,n}(t), \phi_{j-1,\varkappa}(t) > \end{aligned} \tag{2.69}$$

$$\begin{aligned} cD_1(\varkappa) &= < x(t), w_{j-1,\varkappa}(t) > \\ &= < \sum_n cA_0(n)\phi_{j,n}(t), w_{j-1,\varkappa}(t) > \\ &= \sum_n cA_0(n) < \phi_{j,n}(t), w_{j-1,\varkappa}(t) > \end{aligned} \tag{2.70}$$

If we bring the $< \phi_{j,n}(t), \phi_{j-1,\varkappa}(t) >= h_0(n-2\varkappa)$ and $< \phi_{j,n}(t), w_{j-1,\varkappa}(t) >= h_1(n-2\varkappa)$ to Eqs. (2.69) and (2.70), we can complete the calculations:

$$cA_1(\varkappa) = \sum_n h_0(n-2\varkappa)cA_0(n) \tag{2.71}$$

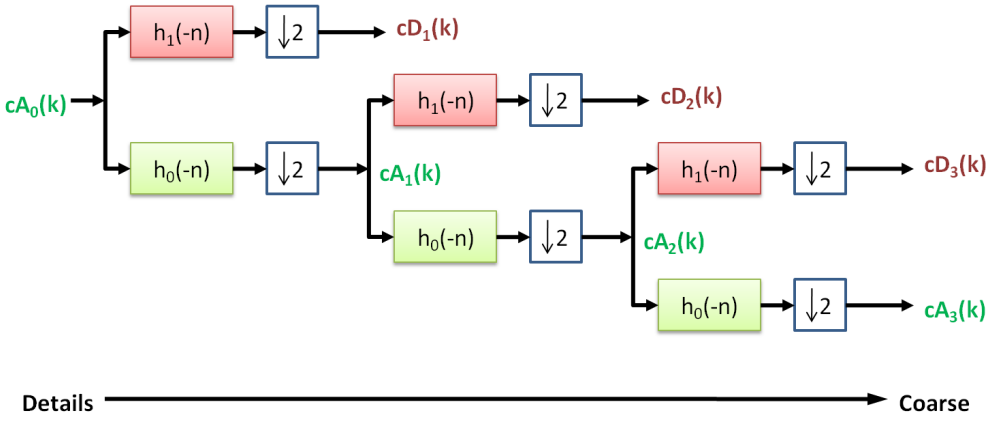


Figure 2.19: Three-Stage Analysis Filter Bank

and

$$cD_1(\varkappa) = \sum_n h_1(n - 2\varkappa) cA_0(n) \tag{2.72}$$

where $h_0(n-2\varkappa)$ and $h_1(n-2\varkappa)$ are the shifted filters in Eqs. (2.64) and (2.65).

Because of the space limitation, the derivation steps of the equations $< \phi_{j,n}(t), \phi_{j-1,\varkappa}(t) >= h_0(n-2\varkappa)$ and $< \phi_{j,n}(t), w_{j-1,\varkappa}(t) >= h_1(n-2\varkappa)$ will not be represented. For more details, please see Proofs 1 and 2 in Appendix B.

It is clear that as long as we get the coefficient cA_0, the other coefficients in different index levels can be derived by filters. These filters are called filter bank and the connections between the filters and the coefficients are illustrated in the Figure 2.19.

On the other hand, the decomposition of a signal into an approximation and a detail can be reversed. That is

$$\begin{aligned} cA_0(n) &= < x(t), \phi_{j,n}(t) > \\ &= < \sum_\varkappa cA_1(\varkappa)\phi_{j-1,\varkappa}(t) + \sum_\varkappa cD_1(\varkappa) w_{j-1,\varkappa}(t), \phi_{j,n}(t) > \\ &= \sum_\varkappa cA_1(\varkappa) < \phi_{j-1,\varkappa}(t), \phi_{j,n}(t) > + \sum_\varkappa cD_1(\varkappa) < w_{j-1,\varkappa}(t), \phi_{j,n}(t) > \\ &= \sum_\varkappa cA_1(\varkappa) h_0(n-2\varkappa) + \sum_\varkappa cD_1(\varkappa) h_1(n-2\varkappa) \end{aligned} \tag{2.73}$$

This synthesis formula can be understood in terms of upsampling and filtering. It tells that the signal can be synthesized if we keep the approximation coefficient and detail coefficients

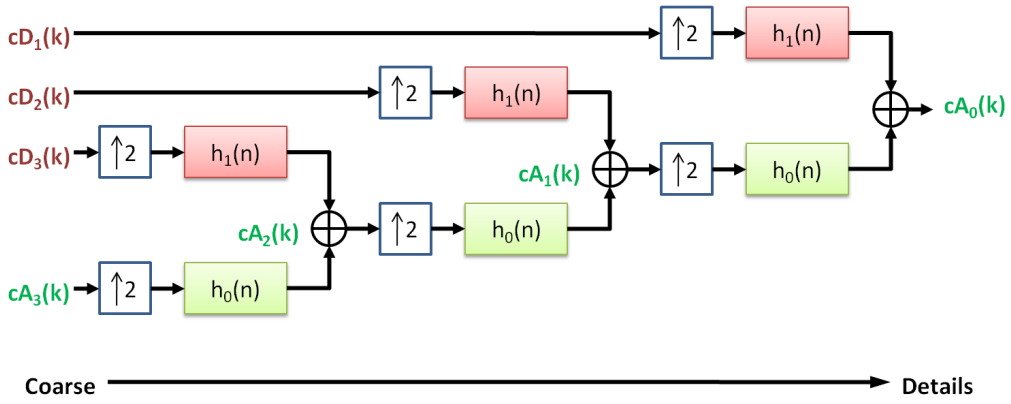


Figure 2.20: Three-Stage Synthesis Filter Bank

at different level. Figure 2.20 depicts the relations.

All above illustrate how a pair of *discrete time filters*, and a *perfect reconstruction filter bank* which can be used to calculate the DWT quickly.

2.3.6 Properties of the Filters, the Scale and Wavelet Functions

In order to make the section more complete, the properties of the filters, and the scale and wavelet functions need to be listed briefly. The details of designing a wavelet which are out of the scope of this thesis can be found in [62]. Because these properties lead to the criteria of designing wavelet bases. All available wavelets so far, have to obey these properties including Haar, which is used in our scheme.

All the properties are based on the following assumptions:

1. **Orthogonal for the Wavelet:** the baby wavelets at different scale and shifting time are orthogonal and every baby wavelet itself has unit energy.

$$\int_{-\infty}^{\infty} w_{j,\varkappa}(t) w_{m,n}(t) dt = \delta(j-m)\delta(\varkappa - n) \tag{2.74}$$

2. **Orthogonal for the Scale:** the translates, $\phi(t-\varkappa)$, of the scaling function, $-\infty < \varkappa < \infty$, are orthogonal and every translate itself has unit energy $(< \phi(t), \phi(t) >= 1)$.

$$\int_{-\infty}^{\infty} \phi(t)\phi(t-n)dt = \delta(n) \tag{2.75}$$

3. **Completeness:** the translates $\phi_{t-\varkappa}$, $-\infty < \varkappa < \infty$, span the same space as the wavelets $w_{j,\varkappa}(t)$, $-\infty < j < 0$ and $-\infty < k < \infty$.

From the two scale equation, it can be easily derived that:

$$\phi_(t-n) = \sum_{\varkappa} h_0(\varkappa - 2n)\sqrt{2}\phi(2t-\varkappa) \tag{2.76}$$

The same is true of the wavelet function and the filter $h_1(\varkappa)$.

$$w(t-n) = \sum_{\varkappa} h_1(k-2n)\sqrt{2}\phi(2t-\varkappa) \tag{2.77}$$

The orthogonality of the wavelet and the scale also applies to the filters, which is called the *double shift orthogonality relations* of the filters, as shown in the followings:

$$\int_{-\infty}^{\infty} \phi(t)\phi(t-n)dt = \delta(n) = \sum_{\varkappa} h_0(\varkappa)h_0(\varkappa-2n) \tag{2.78}$$

$$\int_{-\infty}^{\infty} w(t)w(t-n)dt = \delta(n) = \sum_{\varkappa} h_1(\varkappa)h_1(\varkappa-2n) \tag{2.79}$$

$$\int_{-\infty}^{\infty} \phi(t)w(t-n)dt = 0 = \sum_{\varkappa} h_0(\varkappa)h_1(\varkappa-2n) \tag{2.80}$$

The previously listed assumptions and the double shift orthogonality relations lead to a number of other properties of the filters:

1. Normalization: $\sum_{\varkappa} h_0(\varkappa)^2 = 1$ and $\sum_{\varkappa} h_1(\varkappa)^2 = 1$

2. Sums of the filters: $\sum_{\varkappa} h_0(\varkappa) = \sqrt{2}$ and $\sum_{\varkappa} h_1(\varkappa) = 0$
   For a proof of this equation, please see Proof 3 in Appendix B.

3. Even and odd sums of filters:
   $\sum_{\varkappa} h_0(2\varkappa) = \sum_{\varkappa} h_0(2\varkappa+1) = \frac{1}{\sqrt{2}}$
   $\sum_{\varkappa} h_1(2\varkappa) = -\sum_{\varkappa} h_1(2k\varkappa+1) = \pm\frac{1}{\sqrt{2}}$

4. Alternating flip of filters: $h_1(\varkappa) = (-1)^{\varkappa}h_0(N-\varkappa)$

In summary, there are two types of wavelet transforms: Discrete Wavelet Transforms (DWTs) and Continuous Wavelet Transforms (CWTs). Both DWT and CWT are continuous-time (analog) transforms. They can be used to represent continuous-time (analog) signals. CWTs operate over every possible scale and translation whereas DWTs use a specific subset of scale and translation values. In applications, CWTs are more suitable for similarity detection and DWTs are good at signal compression.

2.4 Matching Pursuit (MP) and Molecular Matching Pursuit (MMP)

2.4.1 The Role of MP and MMP in the Studies

In the field of the signal decomposition, MP and MMP are quite different from the previously described techniques. They are aiming to represent a signal in a more sparse way by searching the components (atoms) in a redundant dictionary. That is, the number of atoms used in the signal decomposition can be much less than the number of bases used from the orthogonal dictionary.

For these two techniques, MP was first studied and applied for music classification (see Chapter 5). In this study, the MP algorithm was utilized to obtain the most representative atoms. Thus, the features that can be extracted from the atoms will be more robust and condensed.

MMP was also utilized for gaining sparse decomposition atoms but in a faster way and applied for audio feature fingerprinting (see Chapter 6). Due to the difference from MP, the obtained atoms are different, such as the atoms from MDCT in Section 2.2 and DWT in Section 2.3. This leads to the differences in feature extraction for applications.

2.4.2 MP Algorithm

The MP algorithm was developed in 1993 by Mallat and Zhang [1]. This is a sparse signal representation algorithm based on a redundant dictionary. MP is a greedy signal approximation algorithm, selecting at least one atom at each iteration to best match the inner

structures of a signal. Let u_λ be the elements of a dictionary D and α_λ be the coefficient of the original file x projection on the element u_λ. The original signal x can be written in the form:

$$x = \sum_{\lambda} \alpha_\lambda u_\lambda \tag{2.81}$$

In most of the cases, it is sufficient to approximate most of the energy of the signal with a small subset of elements such that x can be expressed by N elements and the residual (error) R_N:

$$x = \sum_{\lambda=0}^{N-1} \alpha_\lambda u_\lambda + R_N \tag{2.82}$$

Each element of the dictionary is also called an *atom*. The MP algorithm is used to find a set of atoms and their corresponding coefficients to minimize the residual R_N. The algorithm is given as follows:

1. Initialization step: $R_0 = x$ and $i = 0$. Computation of each coefficient $\alpha_{\lambda_i} =< R_0, u_{\lambda_i} >$ for all the elements of D;

2. Find the maximum among all the coefficients: $\alpha_{\lambda_i} = \max|\alpha_\lambda|$;

3. New residual calculation: $R_{i+1} = R_i - \alpha_{\lambda_i} u_{\lambda_i}$;

4. Coefficients updating: $\alpha_\lambda =< R_{i+1}, u_\lambda >$

5. Stop criterion: stop if $\alpha_{\lambda_i} < \epsilon$ otherwise $i \Leftarrow i + 1$ and return to step 2.

Notice that other stopping criteria can be used. For example, the iteration times is predecided empirically or the criterion can be based on a certain amount of the energy of the initial signal.

2.4.3 Drawbacks and Faster Solutions of MP Algorithm

The main limitation of the above method is its intrinsic complexity. At every iteration there are two stages that may be computationally expensive: step (2) that looks for the maximum of the inner product, which can be lengthy if the dictionary is large; step (4) when one has to update the inner products of the residual with every element of the dictionary.

Instead of searching in a very redundant dictionary, the search for the atoms that best match the signal residues can be limited to a sub-dictionary, which can be much smaller than the original dictionary. This faster version of MP is implemented as follows: the pursuits are performed only on a set of maximum atoms which correspond to the most dominant local maxima, i.e., the small areas on the spectrogram of a signal or its residue with the highest energy concentration (both in time and frequency). When no qualified atoms are left (either because they have all been selected or because after a few iterations their energy is too low), then the corresponding spectrograms are updated (using the residual) and a new set of maximum atoms are selected [63]. The algorithm performs the pursuit on this new set and so on. To use this faster decomposition, the number of maxima in the set needs to be specified. If the number is 1, then this method is exactly equivalent to the regular MP, which is searching the best match in the whole dictionary. The more maxima put in the set, the faster the algorithm and the less accurate the signal approximation will be. Another fast approach, weak matching pursuit, stops the search as long as

$$\alpha_{\lambda_i} \geq \rho \max|\alpha_\lambda| \tag{2.83}$$

where $\rho \in [0, 1]$ is a fixed weak parameter. Therefore, the atom is nearly optimal.

However, the fast methods still require high computation, and so far none of them have been realistically applied globally to very large dataset. Existing practical solutions use only local searches on a frame-by-frame basis [57].

2.4.4 MMP Algorithm

The MMP algorithm was initially proposed as another faster approach and applied for audio decomposition in 2006 [57]. The difference between the fast methods above and MMP is that: the scheme can be faster because several atoms can be subtracted from the residual at the same time for each iteration. Besides, in MP decompositions, the localization of the selected atoms in the time-frequency/time-scale planes is not uniform but reveals some of the intrinsic structure of the analyzed signal; MMP is to make use of this structural information, by grouping together atoms of the same class (i.e., belonging to the same orthonormal basis) with neighboring time-frequency/time-scale parameters. This results in the simple structure of the corresponding grouped coefficients such that the further encoding procedure can be simpler because less information is needed.

In order to obtain most of the energy of the initial signal concentrated in a small number of elements, the choice of an appropriate dictionary is very important. Since the scheme was designed for audio decomposition, the following description of this scheme will base on audio signal.

Most of the audio signals can be modeled as the sum of two elementary components: the tonal part (sum of sinusoids), and the transient part (sum of Diracs). That is why MDCT with atoms u_λ and DWT with atoms v_λ are used to construct a 2-times redundant dictionary $\mathcal{D}$ for MMP, with $\mathcal{D} = \mathcal{C} \cup \mathcal{W}$, where $\mathcal{C}$ is an orthogonal basis of lapped cosines (also called an MDCT basis), and $\mathcal{W}$ is an orthogonal basis of discrete wavelets. MDCT atoms are used to present the tonal part, and DWT atoms are used to present the transient part.

The last step (step 4 mentioned below) of each iteration is to group atoms within a certain window around the significant molecule found. The MMP algorithm could be summarized as follows:

1. Initialization: $R_0 = x$, and $i = 0$. Compute each MDCT coefficient $c_\lambda =< R_0, u_\lambda >$ for all the elements of $\mathcal{C}$ and each DWT coefficient $w_\lambda =< R_0, v_\lambda >$ for all the elements of $\mathcal{W}$;

2. Compute the molecule index $\mathcal{T}$ that a set of u_λ defines and the molecule index $\mathcal{K}$ that a set of v_λ defines; find $K = \max\mathcal{K}$ and $T = \max\mathcal{T}$.

3. Identify the most significant structure. If $T \geq K$, then the most significant structure is of type "tonal molecule"; otherwise $K > T$, it is of type "transient molecule".

4. For a tonal molecule, identify atoms that define the most significant tonal molecule: $M_i = \sum_{\lambda=1\ldots m_i} c_\lambda$. Update the residual: $R_{i+1} = R_i - \sum_{\lambda=1\ldots m_i} c_\lambda u_\lambda$. Update the coefficients from the new residual: $c_\lambda =< R_{i+1}, u_\lambda >$.
   For a transient molecule, identify atoms that define the most significant transient molecule: $M_i = \sum_{\lambda=1\ldots m_i} w_\lambda$. Update the residual: $R_{i+1} = R_i - \sum_{\lambda=1\ldots m_i} w_\lambda v_\lambda$. Update the coefficients from the new residual: $w_\lambda =< R_{i+1}, v_\lambda >$.

5. Stop criterion: stop if $\max(K, T) < \epsilon$, otherwise $i \Leftarrow i + 1$ and return to step 2.

By comparing with the MP algorithm, it can be observed that only step 3 was added to this algorithm. This significantly reduces the number of times of step (4).

MMP is mainly designed for audio signal because the dictionary it chooses is based on the characteristics of the audio signal. Therefore, for the other signal type, its characteristics can guide us to find another appropriate redundant dictionary.

2.5 Linear Discriminant Aanlysis (LDA)

2.5.1 The Role of LDA in the Studies

Breiman [64] summarizes two types of statistical modelings. One assumes that the data are generated by a given stochastic data model; the other uses algorithmic models and treats the data mechanism as unknown. LDA is optimal for normally distributed data and thus belongs to the model-based method. Support Vector Machine (SVM) and neural network are counted as arithmetic methods which have no knowledge about the data.

In the study, after the data characteristics of the given dataset have been analyzed and features have been selected, LDA method was used to assist the dataset classification and

identification.

2.5.2 Introduction of LDA

Assume the ith dataset is presented by d-dimensional discriminating features, and the features are denoted as $\mathbf{x}_i \in \mathbb{R}^d$. LDA method tries to find the linear combinations of these discriminating features that best separate the groups of dataset (samples). These combinations are called canonical discriminant functions and have the form:

$$\mathsf{x}_\mathsf{i}^{'} = \mathsf{w}_1\mathsf{x}_{\mathsf{i}1} + \mathsf{w}_2\mathsf{x}_{\mathsf{i}2} + ... + \mathsf{w}_\mathsf{d}\mathsf{x}_{\mathsf{id}} + \mathsf{w}_0, \tag{2.84}$$

The equation above can be expressed in matrix form,

$$\mathsf{x}_\mathsf{i}^{'} = \mathbf{w}^\mathsf{T}\mathbf{x} \tag{2.85}$$

where $\mathsf{x}_\mathsf{i}^{'}$ indicates the co-ordinate of the projection of $\mathbf{x}_i$ onto the discriminating vector $\mathbf{w}$. Thus, the scale value, $\mathsf{x}_\mathsf{i}^{'}$, represents the mapping from $\mathbb{R}^d$ to $\mathbb{R}$, that is, from the original d-dimensional space to a one dimensional space (along $\mathbf{w}$). The scale value, $\mathsf{x}_\mathsf{i}^{'}$, is then associated with the original class label y_i where $y_i \in \{class_1, class_2, \ldots, class_k\}$ given k classes. When the classes are maximally separated, the scale values belong to different classes have least overlapping. To meet this requirement, the discriminating vector $\mathbf{w}$ are derived during the LDA procedure.

The maximization lies in two aspects: maximizing the difference between classes and minimizing the variance within class. In other words, between-class distance should be as large as possible, meanwhile the within-class scatter should be as small as possible. Because a large variance would lead to possible overlaps among the points of the classes due to the large spread of the points, and thus one may fail to have a good separation.

Suppose there are $k = 2$ classes and thus the dataset $\mathcal{G}$ can be partitioned into $\mathcal{G}_1$ and $\mathcal{G}_2$. Let $|\mathcal{G}_1| = n_1$ and $|\mathcal{G}_2| = n_2$.

LDA defines the maximization criterion as:

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}, \tag{2.86}$$

where

$$\mathsf{m}_1 = \frac{\sum_{\mathbf{x}_i \in \mathcal{G}_1} \mathsf{x}_i'}{n_1} \quad (\mathsf{m}_\mathsf{i} \text{ is the mean of the projected points in } \mathcal{G}_i) \tag{2.87}$$

$$= \frac{\sum_{\mathbf{x}_i \in \mathcal{G}_1} \mathbf{w}^T \mathbf{x}_i}{n_1} \tag{2.88}$$

$$= \mathbf{w}^T \left(\frac{\sum_{\mathbf{x}_i \in \mathcal{G}_1} \mathbf{x}_i}{n_1} \right) \tag{2.89}$$

$$= \mathbf{w}^T \boldsymbol{\mu}_1 \quad (\boldsymbol{\mu}_1 \text{ is the mean of all the points in } \mathcal{G}_1) \tag{2.90}$$

$$\mathsf{m}_2 = \mathbf{w}^T \boldsymbol{\mu}_2 \quad (\boldsymbol{\mu}_2 \text{ is the mean of all the points in } \mathcal{G}_2) \tag{2.91}$$

$$|\mathsf{m}_1 - \mathsf{m}_2|^2 = |\mathbf{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)|^2 \tag{2.92}$$

$$= \mathbf{w}^T \mathbf{S}_B \mathbf{w} \quad (\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\mathsf{T} \text{ denotes the } \textit{between class scatter matrix}) \tag{2.93}$$

$$\mathsf{s}_1^2 = \sum_{\mathbf{x}_i \in \mathcal{G}_1} (\mathsf{x}_i' - \mathsf{m}_i)^2 \quad (\mathsf{s}_i^2, \text{ called } \textit{scatter}, \text{ represents the total squared deviation from the mean}) \tag{2.94}$$

$$= \mathbf{w}^T \left(\sum_{\mathbf{x}_i \in \mathcal{G}_1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T \right) \mathbf{w} \tag{2.95}$$

$$= \mathbf{w}^T \mathbf{S}_1 \mathbf{w} \quad (\mathbf{S}_1 \text{ is called } \textit{scatter matrix} \text{ for } \mathcal{G}_1) \tag{2.96}$$

$$\mathsf{s}_2^2 = \mathbf{w}^T \mathbf{S}_2 \mathbf{w} \quad (\mathbf{S}_2 \text{ is called } \textit{scatter matrix} \text{ for } \mathcal{G}_2) \tag{2.97}$$

$$\mathsf{s}_1^2 + \mathsf{s}_2^2 = \mathbf{w}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w} \tag{2.98}$$

$$= \mathbf{w}^T \mathbf{S}_W \mathbf{w} \quad (\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2 \text{ denotes the } \textit{within class scatter matrix} \text{ for the pooled data}) \tag{2.99}$$

Therefore, the maximization criterion (2.86) can be rewritten as follows:

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{2.100}$$

By using the Lagrange multiplier with the condition $\mathbf{w}^T \mathbf{S}_W \mathbf{w} = 1$, the best vector $\mathbf{w}$ satisfies the following equations

$$\mathbf{S}_B \mathbf{w} = \mathbf{S}_W \mathbf{w} J(\mathbf{w}) \tag{2.101}$$

$$(\mathbf{S}_W^{-1} \mathbf{S}_B) \mathbf{w} = \lambda \mathbf{w} \quad (\text{if } \mathbf{S}_W \text{ is invertible}) \tag{2.102}$$

In Eq. (2.102), $\lambda = J(\mathbf{w})$ is the eigenvalue, and $\mathbf{w}$ is the eigenvector of the matrix $\mathbf{S}_W^{-1}\mathbf{S}_B$.

For the $k > 2$ classes scenario, the maximization criterion is defined as following:

$$\max_{\mathbf{W}} J(\mathbf{W}) = \frac{|\mathbf{W}^T\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_W\mathbf{W}|} \tag{2.103}$$

where

$$\mathbf{S}_B = \sum_{i=1}^{k} n_i(\mathsf{m}_i - \mathsf{m})(\mathsf{m}_i - \mathsf{m})^T, \quad \mathsf{m} = \frac{1}{n}\sum_{\mathbf{x}_i \in \mathcal{G}} \mathbf{x}_i \tag{2.104}$$

$$\mathbf{S}_W = \sum_{i=1}^{k} \mathbf{S}_i \tag{2.105}$$

The solution of Eq. (2.103) is equivalent to resolving the following equation:

$$\mathbf{S}_B\mathbf{W} = \mathbf{S}_W\mathbf{W}\Lambda \tag{2.106}$$

where Λ is the eigenvalue matrix.

If we are only interested in the projection to k'-dimensional spaces, the first k' eigenvectors (corresponding to the largest k' eigenvalues) are the discriminant vectors. Thus, these k' discriminant vectors define k' discriminant functions. The significance of the functions follow the order of the vector significance.

2.6 Summary

This Chapter describes and illustrates the sparse decomposition techniques from the perspectives related to the studies. The techniques will be applied to our proposed schemes, which are designed for watermark fingerprinting and feature fingerprinting and depicted in the next three Chapters.

Chapter 3

Content-based Watermark Fingerprinting

3.1 Introduction

Figure 3.1 illustrates the schematic diagram of the fingerprint embedding and identification in the proposed watermark fingerprinting approach.

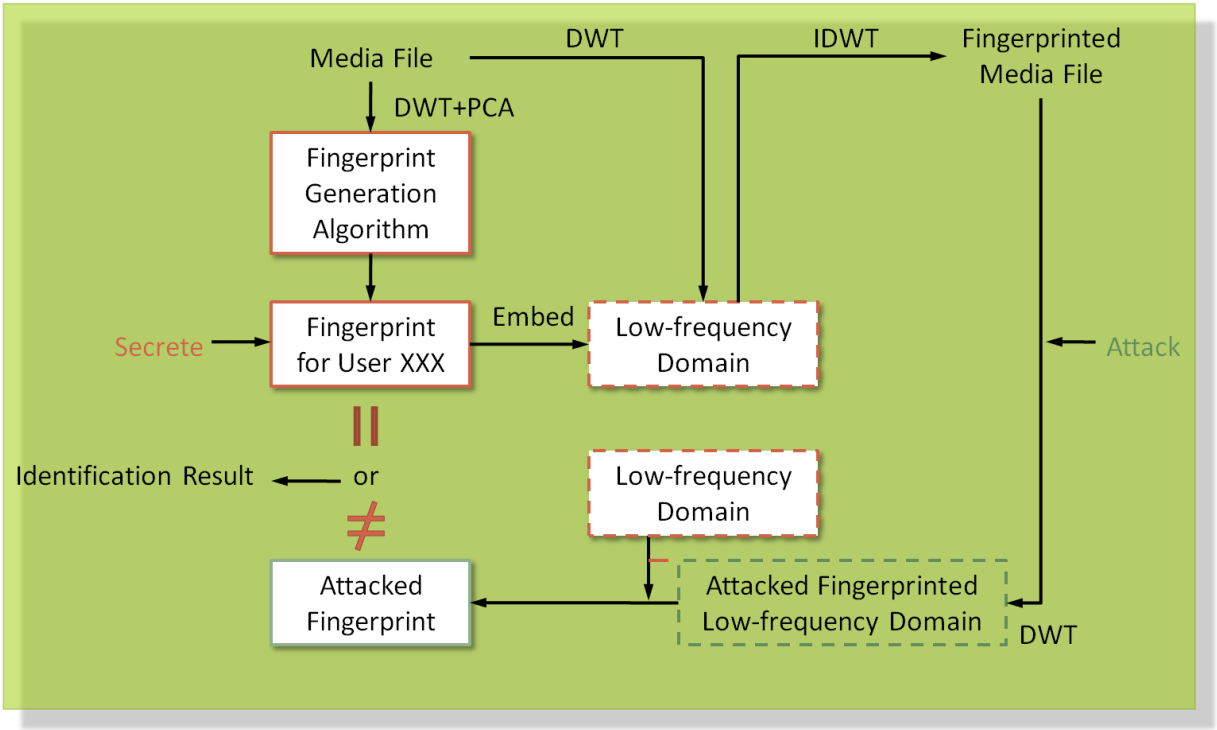


Figure 3.1: Miniature Schematic Diagram of the Proposed Approach.

In this proposed watermark fingerprinting method [65], the fingerprint (to be embedded) causes the changes in both eigenvalues and eigenvectors. Since any attack that corrupts the image will change both the eigenvalues and eigenvectors, if the attack is not strong enough, the fingerprint remains relatively intact. Hence, the proposed method is more robust to common attacks compared to the other two relevant methods: where the fingerprint changes either eigenvalues [19] or eigenvectors [20] only.

Before getting into details of the proposed fingerprinting approach for P2P network, an outline of the approach is given in the following. Unlike the traditional server-client mode networks, the node in P2P networks is not only client but also the server. This feature makes P2P networks maximize resources utilization of the networks. On the other hand, this feature raises a question: since each peer finally has the same copy of the shared file, how does the fingerprint uniquely identify each peer and prevent the peer from sharing it with other peers? To resolve this issue, the source file is decomposed into two parts: *base file* and *supplementary file*. The base file then will carry the embedded unique fingerprint for each peer and be distributed using the traditional server-client mode, while the supplementary file will be freely distributed in P2P networks. Thus, it resolves the conflict of traitor tracing and free sharing. This solution, however, requires the base file and the embedded fingerprint to be small enough to alleviate the load of the server, and at the same time the fingerprint to be kept robust and invisible. The structure of the fingerprint distribution is shown in Figure 3.2. This study proposes sparse signal representation techniques: DWT and PCA to fulfill the criteria.

Before introducing the approach in detail, the technique analysis and comparison between the other two schemes [19][20] (see Section 1.4.2 for details) and the proposed approach in PCA is given in Section 3.1.1. The analysis and comparison are very important because its conclusion predicts that the proposed scheme outperforms these two schemes in robustness. Section 3.1.2 illustrates how the proposed fingerprinting approach generates the fingerprints at small size and embeds them such that the fingerprints can be suitable for the distribution structure and also robust against some attacks and imperceptible to humans. The fingerprint

identification method is introduced in Section 3.1.3. The sharable fingerprinting, as an enhancement method, is introduced in Section 3.1.4. Along the design steps, the design of fingerprint generation using PCA technique is the key in this study. The analytical comparison between ours and other two PCA methods implies that ours should give the best outcome. And the results presented in Section 3.1.5 prove that the approach outperforms the other fingerprint generation methods in robustness against the attacks.

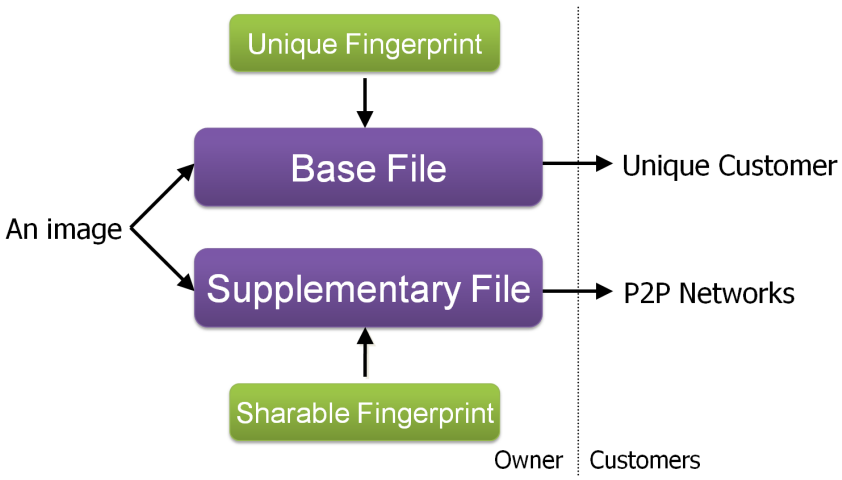


Figure 3.2: Structure of Newly Proposed P2P Fingerprinting Method.

3.1.1 Technique Analysis and Comparison From Different Aspects of PCA

Analysis of Eigenvector and Eigenvalue

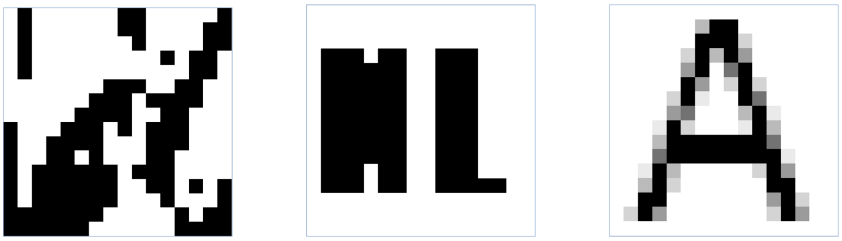
Case 1. The Consequence of Singular-value/Eigenvalue Perturbation As demonstrated in Section 2.1.4, every matrix has a SVD decomposition. Also it has a rank one decomposition form (see Section 2.1.5) and this form is rewritten as follows:

$$X = \sum_i s_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [v_i], \tag{3.1}$$

Within each summand, s_i is a scale factor. If it is very small, the corresponding summand term can be ignored and the reconstructed matrix based on the remaining terms will still

be close to X. The conclusion above explains that PCA is a lossy compression method. Similarly, if all s_i are perturbed a small percentage, the distortion of X will not be perceptible as well. Therefore, this small percentage perturbation can be utilized as a watermark. The drawback is that the altered s_i, however, can be reproduced (or found) according to the rank one decomposition because of its uniqueness. Thus, directly adding a watermark on singular values can be easily attacked by adding another watermark in the same way.

Case 2. The Consequence of A Change on the Matrix of Singular-values/Eigenvalues This section further shows that if the whole matrix of singular values is altered (and this matrix will have a new decomposition) by a watermark, this watermark can possibly not be recovered, and can be completely distorted.



(a) (b) (c)

Figure 3.3: Three 16×16 matrices.(a)a 16×16 image matrix Img. (b)a 16×16 matrix presents the content 'NL'. (c)a 16×16 matrix presents the content 'A'.

SVD theorem states that a matrix can be expressed by concatenating three matrices U, S and V^T. For example, a 16×16 image matrix Img as shown in Figure. 3.3(a) is decomposed as

$$Img = \left[\begin{array}{c}\\\\\\\\\\\\\\\\\end{array}\right] \tag{3.2}$$

$$= U_{Img} S_{Img} V_{Img}^T. \tag{3.3}$$

$$= \begin{bmatrix} -0.32 & 0.25 & \dots & -0.00 \\ -0.29 & 0.33 & \dots & -0.21 \\ \dots & \dots & \dots & \dots \\ -0.15 & 0.07 & \dots & 0.21 \end{bmatrix} \begin{bmatrix} 2431 & 0 & \dots & 0 \\ 0 & 1170 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0.00 \end{bmatrix} \begin{bmatrix} -0.25 & 0.24 & \dots & 0.71 \\ -0.20 & -0.44 & \dots & 0.00 \\ \dots & \dots & \dots & \dots \\ -0.24 & -0.24 & \dots & 0.00 \end{bmatrix} \tag{3.4}$$

Suppose we want to embed a content 'NL' shown in Figure 3.3(b) in matrix *Img*. One approach is to add the content to the matrix of singular values S_{Img} as given by Eq. 3.5:

$$\begin{aligned} & S_{Img} + \alpha[NL] \\ & = U_{INL} S_{INL} V_{INL}^T \end{aligned} \tag{3.5}$$

$$= \begin{bmatrix} -0.99 & 0.03 & \dots & -0.0016 \\ -0.02 & -0.99 & \dots & -0.01 \\ \dots & \dots & \dots & \dots \\ -0.01 & -0.03 & \dots & -0.96 \end{bmatrix} \begin{bmatrix} 2462 & 0 & \dots & 0 \\ 0 & 1204 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 4.12 \end{bmatrix} \begin{bmatrix} -0.99 & 0.03 & \dots & -0.00 \\ -0.02 & -0.99 & \dots & -0.00 \\ \dots & \dots & \dots & \dots \\ -0.01 & -0.23 & \dots & 0.96 \end{bmatrix} \tag{3.6}$$

The real coefficient α is set to be a small value so that $S_{Img} \approx S_{INL}$ as depicted in Eqs. 3.4 and 3.6. By reversing the Eq. 3.5, the embedded content can be recovered.

The watermarked *Img* is defined by Img_w and denoted as:

$$Img_w = U_{Img} S_{INL} V_{Img}^T. \tag{3.7}$$

Let 'NL' be an originally embedded watermark and 'A' an attack watermark shown in Figure 3.3(c). If 'A' attacks the singular value, S_{INL}, following the same way as 'NL' embedded in S_{Img}, we get:

$$S_{INL} + \alpha[A] \;=\; U_{IA} S_{IA} V_{IA}^T \tag{3.8}$$

If U_{IA} and V_{IA}^T respectively replace U_{INL} and V_{INL}^T in Eq. 3.5, the recovered watermark '$\widetilde{A}$' as shown in Figure 3.4(a) is similar to 'A'. This process is depicted in Eq. 3.9:

$$U_{IA} S_{INL} V_{IA}^T - S_{Img} = \alpha[\widetilde{A}] \approx \alpha[A] \tag{3.9}$$

The example above shows that if the singular values S_{Img} is altered by embedding a watermark 'NL', this watermark can be destroyed by replacing its singular vectors with the ones of another watermark 'A'. That is, even if S_{INL} is intact, the embedded watermark 'NL' can be proved and claimed as another watermark 'A'.

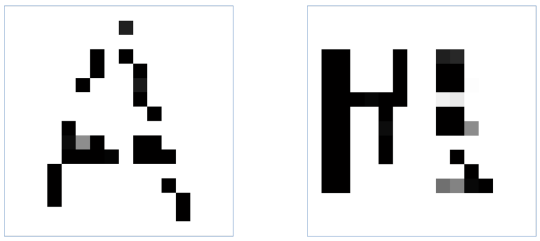
The similar scenario applies the other way around, that is, 'A' is treated as an originally embedded watermark and 'NL' an attack watermark as depicted in Eqs. 3.10 and 3.11.

$$S_{Img} + \alpha[A] \;=\; U_{IA'} S_{IA'} V_{IA'}^T \tag{3.10}$$

$$S_{IA'} + \alpha[NL] \;=\; U_{INL'} S_{INL'} V_{INL'}^T \tag{3.11}$$

After the respective replacements of $U_{IA'}$ and $V_{IA'}^T$ in Eq. 3.10 by $U_{INL'}$ and $V_{INL'}^T$, the reconstructed watermark '$\widetilde{NL}$' seems like 'NL' as displayed in Figure 3.4(b). Eq. 3.12 depicts the watermark reconstruction procedure and illustrates that if S_{Img} is altered by the watermark 'A', this watermark can be demolished by another watermark, such as 'NL', due to the same reason as shown above.

$$U_{INL'}S_{IA'}V_{INL}^{T'} - S_{Img} = \alpha[\widetilde{NL}] \approx \alpha[NL] \tag{3.12}$$



(a) (b)

Figure 3.4: Watermark reconstruction results. (a)a 16×16 matrix presents the content '$\widetilde{A}$'. (b)a 16×16 matrix presents the content '$\widetilde{NL}$'.

Cases 1 and 2 are saying that eigenvalues (singular values) by themselves are not robust enough to conserve an embedded content, such as a watermark.

Case 3. The Consequence of Singular-vector/Eigenvector Perturbation In this section, we will discuss about the weakness of the perturbation on singular vector (eigenvector).

First, we studied an approach on how a watermark is embedded into and recovered from the projected data using the eigenvectors of the data. In the approach, watermark embedding involves the original data X being projected onto one of eigenvectors, v_i, followed by adding an original watermark, Δ_1. The operation is given by Eq. 3.13:

$$Xv_i + \Delta_1 \tag{3.13}$$

This watermark can be recovered using the same eigenvector v_i as shown in Eqs. 3.14 and

3.15:

$$(Xv_i + \Delta_1)v_i^T = X + \Delta_1 v_i^T, \tag{3.14}$$

$$(X + \Delta_1 v_i^T - X)v_i = \Delta_1. \tag{3.15}$$

The watermark embedding approach, however, will fail to identify the original watermark if an attack watermark, Δ_2, is added, not on the projected data but the watermarked data as shown in Eqs. 3.16 and 3.17.

$$X + \Delta_1 v_i^T + \Delta_2 \tag{3.16}$$

$$\begin{aligned}(X + \Delta_1 v_i^T + \Delta_2 - X)v_i &= \Delta_1 + \Delta_2 v_i \\ &\neq \Delta_1.\end{aligned} \tag{3.17}$$

This observation proves that the attack watermark Δ_2 can easily prevent the recognition of the original watermark Δ_1 if we rely on eigenvector(s) alone.

Next, we will study another watermark embedding process where a watermark is directly added on one of the singular vectors (eigenvectors) as given by equation:

$$v_i^{'} = v_i + \Delta_1 \tag{3.18}$$

where Δ_1 is the embedded watermark. With the embedded watermark, the original data X is changed to watermarked data X^+ as given by Eq. 3.19:

$$X^+ = X + s_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [\Delta_1]. \tag{3.19}$$

The problem is, once X^+ is re-decomposed, the expression will not include the term $s_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [\Delta_1]$ any more in that the factors of every term in rank one decomposition appear in

pairs, such as $s_k \begin{bmatrix} | \\ u_k \\ | \end{bmatrix} [v_k]$. It means the re-decomposition after the watermark embedding will not re-produce the terms in the previous decomposition. Therefore, the embedded watermark is not reproducible. As a result, all or part of the decomposition expression will change accordingly in order to make each singular vector normalized. Furthermore, the energy created by $s_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [\Delta_1]$ has been distributed in the new singular values.

Even if sometimes the eigenvectors are not required to obey normalization, the re-decomposition after the watermark embedding will not re-produce the terms in the previous decomposition as well. This is because the eigenvectors must obey the orthogonality among each other. The modification in one eigenvector will cause the changes to the other eigenvectors.

The studies above depict that according to rank one decomposition, the watermark over the singular vector can not be reconstructed. The reasons are the normalization and orthogonal properties of the singular vectors make themselves more sensitive to any change.

It seems that there is a good solution to solve this issue: by mapping the overall changes to the perturbation (watermark). It is, however, too ideal. This mapping is vulnerable to an attack watermark if it behaves the same way as the original watermark Δ_1.

As can be concluded from above, either relying on a data projection on a singular vector (eigenvector) to track a change or tracking a change of vector itself is not possible.

To summarize the discussion of this section:

1. a perturbation of singular values:

    - After the alteration of singular values and original matrix re-decomposition, the alteration can be reproduced and further destroyed by any random singular value changes.
    - After the alteration of the matrix of singular values by adding a pattern matrix, the newly generated singular values matches the pattern, but it can also match the other patterns. That is, the change of singular values can not uniquely match

to a pattern.

2. a perturbation of singular eigenvectors: Any modification with respect to the singular vectors can be distorted by an attack watermark.

In the rest of the Chapter, a new approach is proposed such that it manipulates both singular values (eigenvalues) and singular vectors (eigenvectors). The goal is to make it possible that the robustness and the unique identification of the alteration can be sustained. The drawback is that it may cause a perceptible distortion of the matrix. The approach considers the defect and does a good balance in between. The results show that it performs much better than the other two approaches where the perturbation manipulation is only applied to either singular values (eigenvalues) or singular vectors (eigenvectors).

3.1.2 Fingerprint Generation and Embedding

To give the description of the proposed fingerprint generation and embedding approach for P2P a focus, we concentrate on how the approach works on images. The proposed approach, of course, can be generalized and applied to other types of media files.

Since the base file will be distributed from the central server to all the clients, it should be designed to have small size but contain the most important information. Thus, the load of the server can be alleviated to some extent, while the supplementary file can be larger but contains less important information. By doing this, the peer who has the supplementary file has no commercial motivation to leak the supplementary file alone because of its low quality without the base file. One possible approach to derive a small size base file is to decompose the file into two parts - the base pixel matrix and the detail pixel matrix. The base pixel matrix can give us a rough outline of the image. Since the base pixel matrix has higher correlation information, its entropy value is small so that it can be compressed into a very small size with no quality loss.

The P2P watermark fingerprinting method employs wavelet transform to model the low frequency feature of the image and PCA to further decompose it into eigenvectors. After the preprocessing, any one vector can be adopted to generate one fingerprint by following a rule.

Literature review shows that wavelet [66] and PCA [67] techniques as features were utilized to detect the image information. This is the first time that wavelet and PCA techniques are employed as sparse representation methods for fingerprint generation and embedding.

In wavelet transform, an image is split into one approximation (also called approximation coefficient) w_a and three details in horizontal, vertical and diagonal directions which are named w_h (or horizontal coefficient), w_v (vertical coefficient) and w_d (diagonal coefficient). The approximation is then itself split into a second-level approximation and details, and the process is repeated. For a J-level decomposition, the approximation and the details are described in Eq. 3.20,

$$\begin{aligned} w_{aJ} &= \langle I \cdot A_J \rangle, && \\ w_{hj} &= \langle I \cdot H_j \rangle, && j = J, ..., 1 \\ w_{vj} &= \langle I \cdot V_j \rangle, && j = J, ..., 1 \\ w_{dj} &= \langle I \cdot D_j \rangle, && j = J, ..., 1 \end{aligned} \tag{3.20}$$

where I denotes image and A_J, H_j, V_j, and D_j are wavelet bases. For image decomposition, even the size of the coefficients in different level is different, but the coefficients are still a 2-D matrix. Eq. 3.21 indicates how the image is recovered:

$$\begin{aligned} I &= a_J + \sum_{j=1}^{J} d_j \\ &= w_{aJ} A_J + \sum_{j=1}^{J} w_{hj} H_j + \sum_{j=1}^{J} w_{vj} V_j \\ &\quad + \sum_{j=1}^{J} w_{dj} D_j. \end{aligned} \tag{3.21}$$

The original definition can be found in [68].

In this method, the fingerprint is small but strong and robust compared to the multimedia file. Figure 3.5 shows the brief procedure of the fingerprint generation and embedding. In this research, we only implement the fingerprint method on red channel of the image. Even the message in the blue channel is less sensitive to the human eyes [69][70], the channel

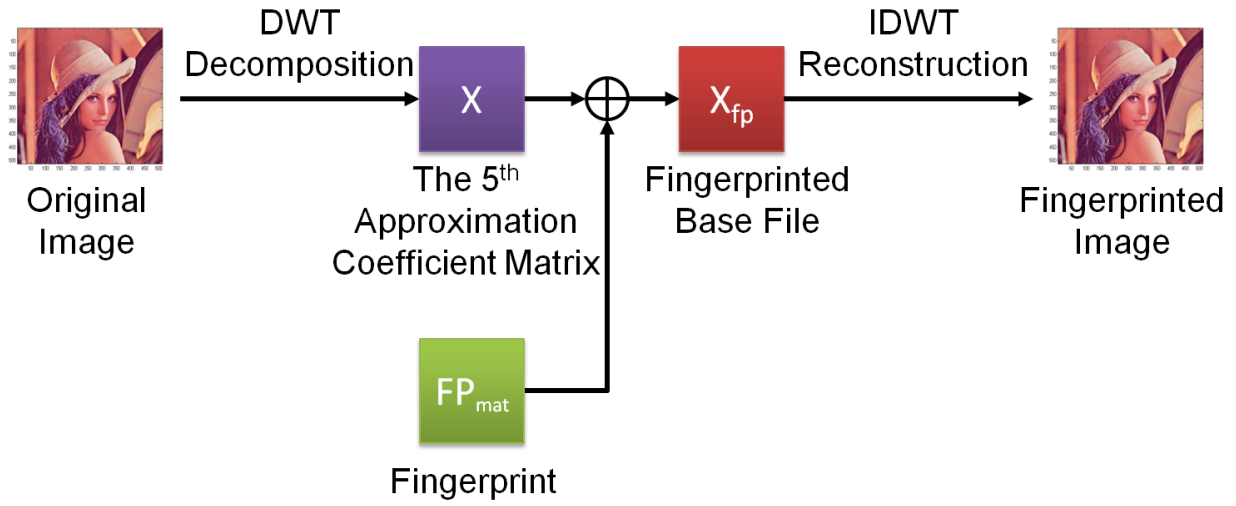


Figure 3.5: Fingerprint Embedding Flowchart.

selection is not very important at the current stage for showing the feasibility and the advantages of the method. To create a fingerprint, first, the image is decomposed into N^{th} level by DWT such that the final decomposition is a 16×16 matrix. The fingerprint is then generated based on the matrix. The choice of 16×16 is the balance between the reduction of PCA computational time and the fingerprint identification accuracy. In our study, we use images with a size 512×512. The image is decomposed into 5^{th} level by DWT. For 5-level decomposition, the coefficient set is $W = [w_{a5}, w_{h5}, w_{v5}, w_{d5}, w_{h4}, w_{v4}, w_{d4}, ..., w_{h1}, w_{v1}, w_{d1}]$. At the 5^{th} level, the size of the approximate coefficient w_{a5} is significantly reduced from the original to 16×16. This coefficient, called X from now on, is defined as the base file as mentioned before. Correspondingly, the other coefficients are defined as supplementary file. The base file was then used to calculate its principal vectors. It goes through three steps according to Eqs. 3.22-3.24.

$$step\ 1. \qquad X' = X - \overline{X}, \tag{3.22}$$

$$step\ 2. \qquad X'' = Cov(X', X'^T), \tag{3.23}$$

$$step\ 3. \qquad X''P = P\Lambda, \tag{3.24}$$

where $\overline{X}$ denotes the mean of X. P and Λ present the set of eigenvectors and the set of

eigenvalues of X'' (or $X'X'^T$). Since X'' is a reversible matrix in this case, there are a total of 16 eigenvectors which make up the columns of P. It is presented as

$$P = [\vec{p}_{16}|\vec{p}_{15}|...|\vec{p}_1], \tag{3.25}$$

where the eigenvectors are arranged in descending order according to their principal components and each eigenvector is a 16x1 vector. Eq. 3.26 illustrates how the preparatory fingerprint matrix FP^{pre}_{mat} is derived:

$$FP^{pre}_{mat} = Y \times (\vec{S} \times \vec{p}^{T}_{m})_{full}, \tag{3.26}$$

where a scale vector defined as $\vec{S}$, which is a 16x1 vector, is multiplied with Y and one of the eigenvectors, for instance $\vec{p}_m$ $(m = 1, 2, ..., 16)$. Y is a visually meaningful full matrix with all positive elements, and only known by the source owner. Thus, it can be a Company's logo, another low resolution of a portion of the original host image, or simply a portion of the host image. It is utilized to prove the right ownership fingerprint. The elements of $\vec{S}$ can be viewed as the coefficient of the fingerprint amplitude. The bigger the values of $\vec{S}$ are, the more visible distortion the fingerprint creates; the smaller the values of $\vec{S}$ are, the weaker the fingerprint energy will be. The value s_i $(i = 1, 2, ..., 16)$ in the scale vector is chosen on the basis of empirical optimization. T indicates the transpose operation. $full$ denotes the transformation operation of the nonreversible matrix to reversible matrix by fine tuning the matrix of singular values of the nonreversible matrix. After the multiplication, the matrix size is 16×16 which is the same as Y. Also, even the data items in each column have different magnitudes but they have the same sign, and this sign matches with the corresponding entries of the eigenvector $\vec{p}^{T}_{m}$. The following Eqs. 3.27-3.29 provide the

explanation:

$$\because y_{ij} > 0, \quad (i,j = 1,...,16; \quad y_{ij} \in Y) \tag{3.27}$$

$$s_j \geqslant 0, \qquad (j = 1,...,16; \quad s_j \in \vec{S}) \tag{3.28}$$

$$\therefore for\ i,j = 1\ to\ 16,$$

$$p_{mi} \times fp^{pre}_{matji} \geqslant 0. \quad (i,j = 1,...,16; \quad p_{mi} \in \vec{p}_m, \quad fp^{pre}_{matji} \in FP^{pre}_{mat}) \tag{3.29}$$

According to Eqs. 3.27-3.28, $Y \times \vec{S}$ is a 16×1 vector with all elements positive. Hence, the ith component p_{mi} of $\vec{p}_m$ determines the sign of the ith column of the matrix $FP^{pre}_{mat} = Y \times \vec{S} \times \vec{p}^{T}_{m}$. This feature implies that the preparatory fingerprint in the horizontal direction follows the trend of the applied eigenvector. The only problem is that it makes the distortion visible after the image is reconstructed. The reason is data items in each column of matrix Y have a similar stretch scale because of the matrix $(\vec{S} \times \vec{p}^{T}_{m})$. The solution can be done by fine adjustment on each element such that the obvious boundary between columns is invisible. This procedure can be modeled as Eq. 3.30. We call the procedure *Column Unify*, because the fine adjustment coefficients from c_{1j} to c_{nj} are created to adjust all the elements in the jth column of matrix FP^{pre}_{mat} to have the same value. The rule of unification not only ensures that the value keeps the sign as before but also maintains a certain difference between two columns. To prevent the previous steps of implementation from creating visual distortion, the variation of the whole matrix was limited by an empirical-based perceptually lossless threshold. Thus, the scale vector and the fine adjustment coefficients should be adjusted

accordingly. The generated fingerprint matrix is named FP_{mat}:

$$\begin{aligned} FP_{mat} &= Column - Unify(FP_{mat}^{pre}) \\ &= diag[c_{11}, c_{21}, \ldots, c_{n1}][FP_{mat}^{pre}]_{COL1} \\ &\quad + diag[c_{12}, c_{22}, \ldots, c_{n2}][FP_{mat}^{pre}]_{COL2} \\ &\quad + \ldots \\ &\quad + diag[c_{1n}, c_{2n}, \ldots, c_{nn}][FP_{mat}^{pre}]_{COLn}, \end{aligned} \tag{3.30}$$

where n equals to 16 in this case. The term $[FP_{mat}^{pre}]_{COLi}$ presents the ith column of the matrix FP_{mat}^{pre}.

FP_{mat} as well as $\overline{X}$ are then added to X' and the fingerprinted image can be reconstructed based on the fingerprinted 5^{th} approximation matrix X_{fp} (alternatively called the fingerprinted base file) using an inverse DWT as shown in Figure 3.5 along with Eq. 3.21.

Liu and Tan [19] mentioned that the singular values of an image have very good stability; that is, when a small perturbation is added to an image, its singular values do not change significantly. In our case, the singular values are eigenvalues. Also, the same concept applies to the eigenvectors, which also means that the differences of each term in SVD of X and X_{fp} are similar. For example, the SVD of X is defined as

$$X = P\lambda Q^T, \tag{3.31}$$

where P as mentioned in Eq. 3.24 is the set of eigenvectors of $X'X'^T$. The eigenvectors of X'^TX' make up the columns of Q. The singular values in λ are square roots of the eigenvalues from $X'X'^T$ or X'^TX'. It can be defined as

$$\Lambda = \lambda\lambda^T, \tag{3.32}$$

since X' is a square matrix, λ equals λ^T. while the SVD of X_{fp} can be similarly defined as

$$X_{fp} = \tilde{P}\tilde{\lambda}\tilde{Q}^T. \tag{3.33}$$

Since the elements of FP_{mat} are small enough compared to the base file X, by adding the fingerprint, the vectors between P and $\tilde{P}$, and Q and $\tilde{Q}$ are very close based on their

correlation coefficients. It can be proved by calculating the overall correlation coefficient of X and X_{fp} as well. For images Lena, Baboon and Peppers, their corresponding correlation coefficients between before and after fingerprinting are 0.9988, 0.9997, and 0.9995. Therefore, the visual aspect of the image is preserved after the fingerprint is embedded.

The fingerprint embedding method not only involves the wavelet but also uses the PCA technique. The reason the wavelet technique is used is because of the advantages that it can provide a scalable approximation matrix associated with the scalable precision, and also because the approximation matrix contains the most important low frequency information in small size. The PCA technique, on the other hand, finds out the orthogonal eigenvectors that a pattern can project into.

Under the unification operation, the robustness of the fingerprint is enhanced. Because the absolute magnitude of elements in each column is replaced by one value, the sign of each column is kept which means the discrimination feature between columns is maintained. The result is presented in Section 3.1.5.

Ideally, X'' has 16 eigenvectors and the same number of fingerprints according to the host image with size 512×512. Such a small number of fingerprints is evidently not enough to represent the large number of users around the world. Fortunately, the multimedia file does not have one frame only; it has many frames in sequence. Any number of frames, for example, 20 out of a total number of 1000 frames can be chosen as the target images and their 16 local eigenvectors, assuming each frame has the size of 512×512, can be determined. If one out of 16 different eigenvectors is chosen from one image, and also other eigenvectors are chosen respectively from the other 19 images to label each customer, there will be 16^{20} different combinations for labeling. The mapping of the labeling and the customer will only be known by the owner.

3.1.3 Fingerprint Identification

Since only the owner, e.g. the media producer, keeps the mapping between the fingerprint and the customer, as long as the producer successfully tracks back the fingerprint for a

suspect video, for example, the pirate customer can be revealed. The suspected video is defined as a video which is freely distributed out of the scope of owners' authorized P2P networks.

In this case, to identify the embedded fingerprint, the multimedia producer needs to decompose the fingerprinted image into level 5 using the wavelet technique so that a 16×16 approximate matrix X'_{fp} is obtained. By deducting X, the difference is the fingerprint matrix FP^*_{mat}. Then the signs (based on the majority rule) of the columns in this matrix will be compared to the signs of each eigenvector using the Hamming distance. The eigenvector that has the minimum Hamming distance to the matrix will be claimed as the embedded fingerprint.

To prove the ownership of the fingerprint issued by the right source owner, the matrix Y^*, approximately equal to Y is first extracted. Then the correlation coefficient is used to decide if the matrix Y exists. Eqs. 3.34-3.35 denote the operations:

$$Y^* = FP^*_{mat} \times (\vec{S} \times \vec{p_m^T})^{-1}_{full}, \tag{3.34}$$

$$C_{y,y^*} = \frac{Y \cdot Y^*}{\| Y \|}. \tag{3.35}$$

3.1.4 Fingerprint Distribution

The fingerprint generation method described in the previous section can be used to generate a unique fingerprint or a sharable fingerprint. The only difference is the unique fingerprint will be individually distributed with the base file, while the sharable fingerprint will be distributed with the image in which it is embedded. As mentioned above, the central server, which is the owner's server, will only distribute the unique fingerprint embedded base files to the customer, while the P2P networks deliver the supplementary files, sharable fingerprint embedded files and regular files. Figure 3.6 depicts the relationship between unique fingerprints and sharable fingerprints in a video with 9 I frames. To simplify the illustration, P frames and B frames are neglected in the figure. These frames are included into the scope

of regular file. The reason of choosing I frame is that these type of frames are encoded independently, while P frames and B frames are encoded on the prediction from I frames. The 1st, 4th and 7th frames are decomposed into base and supplementary files. Within each of these frames, the upper half presents the base file and the lower half presents the supplementary file. The frames that are not decomposed, such as the 2nd, 3rd, 5th, 6th, 8th, and 9th are also counted as supplementary files. The scenario in Figure 3.6 shows that the base files of 1st, 4th and 7th frames are embedded with unique fingerprints and some of the supplementary files, for example the 2nd and 6th frames, are embedded with sharable fingerprints. Finally, the base files in green color will be delivered by center server only and the supplementary files in purple will be distributed in P2P networks.

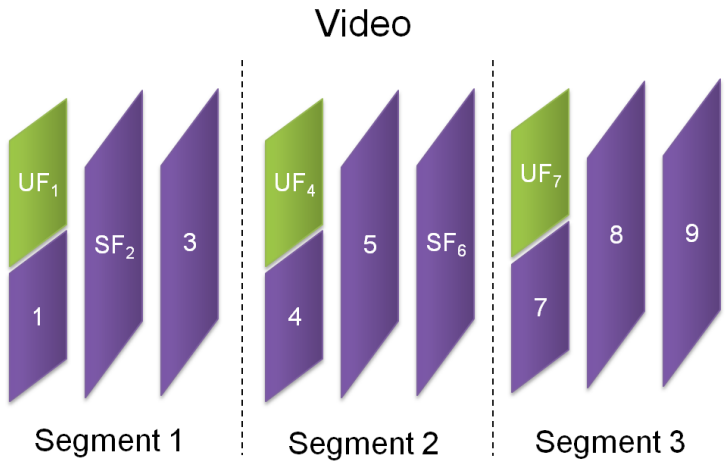


Figure 3.6: Two kinds of fingerprints in a video. UF denotes a unique fingerprint is embedded and SF denotes a sharable fingerprint is embedded.

In distribution stage, when a new multimedia source file is available for its customers to download, the peers that join the P2P networks at very beginning most possibly download the whole video file from the central server, but the peers who joined after may have the file from different peers. For example, a partial video file is downloaded from Peer A with A's sharable fingerprint; the rest is downloaded from Peer B with B's sharable fingerprint or from the owner's server directly. The procedure is illustrated in Figure 3.7. The purpose of involving a sharable fingerprint in the P2P sharing networks is to enhance the efficiency of traitor

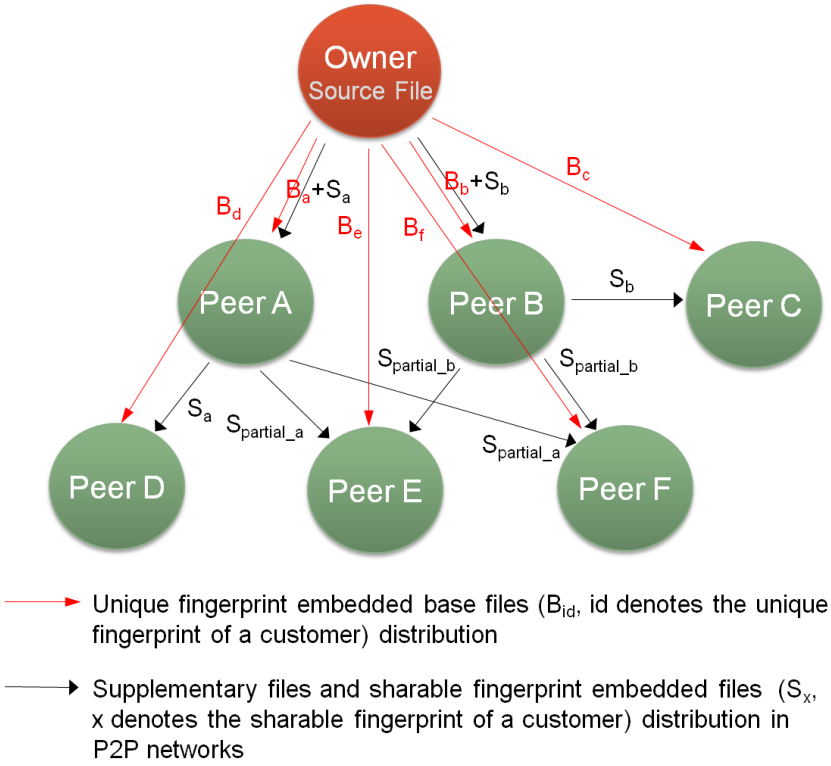


Figure 3.7: The Topology of Base File and Supplementary File Distribution.

tracing. The sharable fingerprint functions as an assistance to provide the source owner a hint about from which peers this video was downloaded. This is especially useful if the attacker successfully removed or attacked the unique fingerprint. It is because a multimedia file consists of many segments and is usually obtained through the P2P networks. Different segments may come from different peers and have different sharable fingerprints. These sharable fingerprints together can be treated as a fingerprint, which may not be unique, but can still help to narrow down the suspected traitors and to provide further evidence to support the result derived from the unique fingerprint.

3.1.5 Attacks

The fingerprint method is studied on a series of common image attack processes that include Gaussian noise, median filter, lossy compression and geometric distortion. The test images - Lena, Baboon, and Peppers - are 512×512, and the fingerprinted images are obtained as

the example shown in Figure 3.5. Under the same subject visual quality of the fingerprinted image, the results show that this method is far more resistant to many common attacks than other methods—Liu [19] and Hien [20]—which also use the concepts of eigenvalues and eigenvectors. This is due to the fact that the watermarks of Liu's and Hien's methods were solely embedded in an eigenvalue or an eigenvector as explained in Section 3.1.1 cases 2 and 3. The drawbacks of those types of embedding have been illustrated in Section 3.1.1. The measurement of the resistance for each image among Liu, Hien and the proposed methods is detection rate. For Liu and Hien methods, the detection rate is defined as the number of correctly detected embedded eigenvalues (or eigenvectors) divided by 16. For the proposed method, the detection rate is the number of successfully detected embedded fingerprints divided by 16.

In term of complexity performance, Liu, Hien and our schemes are approximately same because all of them need to derive the eigenvalues and eigenvectors first and then embed the watermark within one time operation.

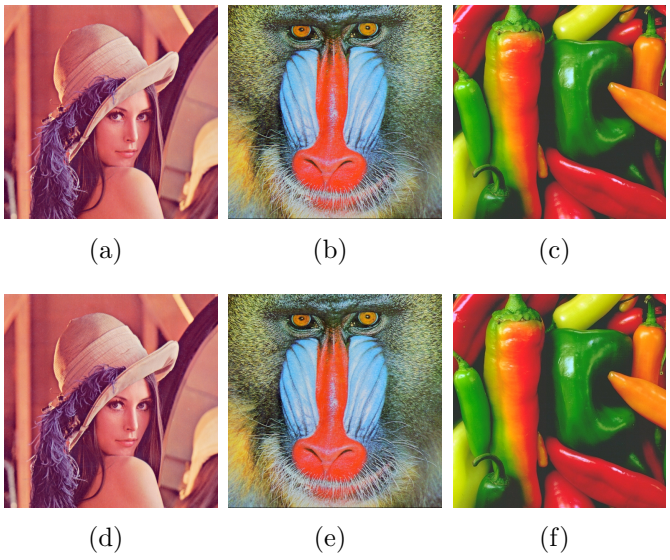


Figure 3.8: Images comparison before and after fingerprinting. (a)Original Lena. (b)Original Baboon. (c)Original Peppers. (d)Fingerprinted Lena. (e)Fingerprinted Baboon. (f)Fingerprinted Peppers.

The experiments suggested that the wrong claims are usually derived from those less

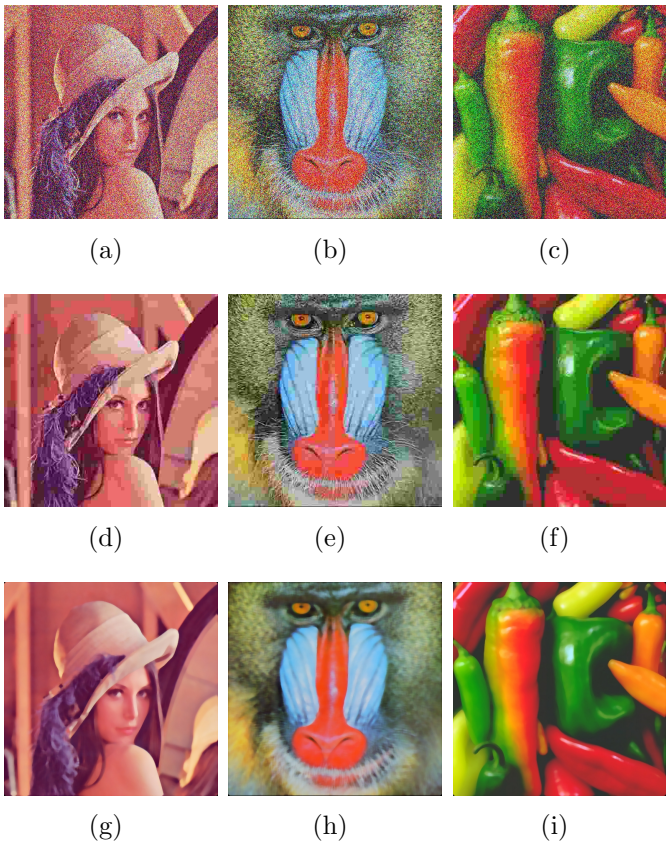


Figure 3.9: Images after Gaussian white noise, compression and median filter.(a)Lena with noise power 7000 (or SNR=4∼5dB).(b)Baboon with noise power 7000 (or SNR=4∼5dB).(c)Peppers with noise power 7000 (or SNR=4∼5dB). (d)Lena at quality 5 of JPEG compression.(e)Baboon at quality 5 of JPEG compression.(f)Peppers at quality 5 of JPEG compression.(g)Lena with median filter [9 9].(h)Baboon with median filter [9 9].(i)Peppers with median filter [9 9].

principal eigenvectors. It implies that the most principal eigenvectors which have a larger Hamming distance gap should be chosen as the vectors to generate the fingerprints for better robustness performance. The eigenvectors which are utilized for fingerprints in this study are selected according to their principals and their Hamming distances. Figures 3.8(a)-(c) and (d)-(f) present the comparison between original images and fingerprinted images, where the fingerprints are included in red channel in this case.

Anti-collusion attacks are another concern in fingerprint design. Some research groups [9][71][8] proposed their schemes against collusion attack. They use the binary codes based on marking assumption or linear combination of a set of orthogonal vectors. At the current stage of our study, we did not consider the anti-collusion attack. However, the sharable fingerprint described above does provide some form of anti-collusion function.

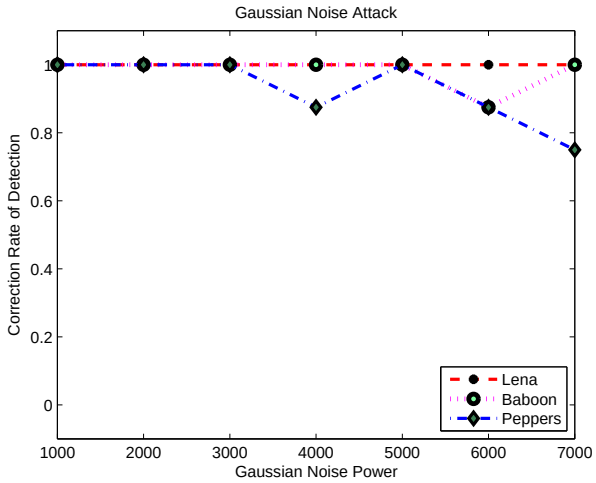
Gaussian Noise Attack

Figures 3.9(a)-(c) present the results of the images after adding Gaussian noise. In these figures, the mean of the additive Gaussian white noise is zero and the power (which is the variance) of the noise is 7000. In other words, the signal noise ratio (SNR) is between $4dB$ and $5dB$. Under this noise, the detection rate of the proposed method starts dropping.

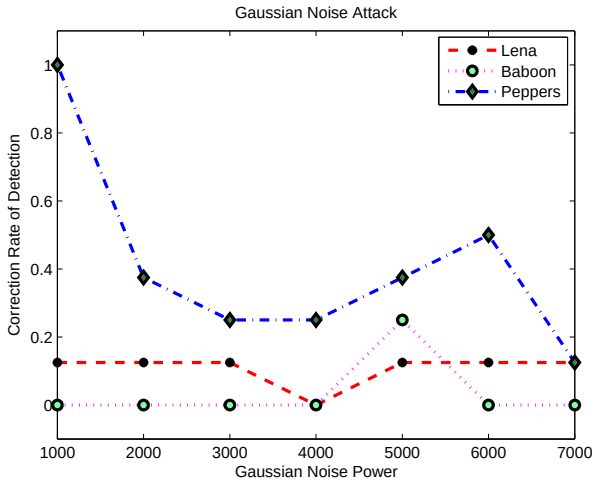
The correction rate of detection corresponding to the different noise power is demonstrated in Figure 3.10. The results show that the proposed method is highly robust against Gaussian noise. The Liu method is relatively worse and its detection rate drops at noise power 2000. Of the methods mentioned above, the Hien method is the weakest method against Gaussian noise attack.

Lossy Compression Attack

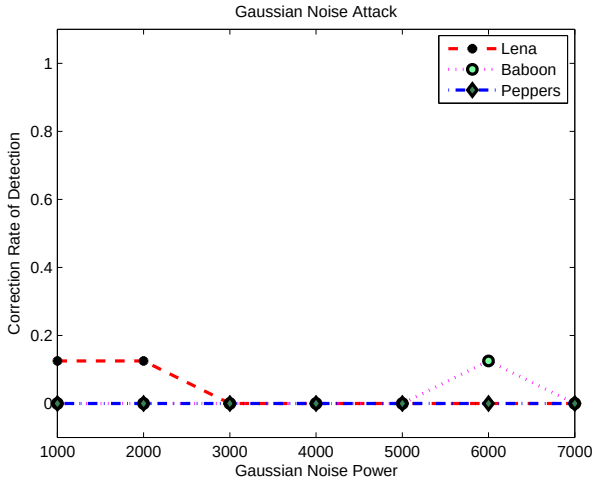
In the experiments with Matlab, the JPEG compression operation is fulfilled by setting the quality level which corresponds to reverse scale of the quantization table. Therefore, the lower the JPEG quality level, the higher the compression ratio will be. But setting the same quality level for different images can correspond to different compression ratios. Figures 3.9(d)-(f) show the three images after JPEG compression. In these figures, the correction rate of



(a)



(b)



(c)

Figure 3.10: Robustness to Gaussian white noise. (a) The proposed method. (b) Liu method. (c) Hien method.

detection starts to drop severely at quality level equals to 5. In this case, the corresponding compression ratios on three images are 102, 63, and 66, respectively. Figure 3.11 illustrates the robustness results of different compression rates. It shows that the fingerprint can be highly detected when the quality is reserved above 10 where the compression ratios on three images are 75, 38 and 50, respectively. The robustness of the Liu method to JPEG compression is highly dependent on the image. In the best case, the detection rate starts decreasing when the image quality is maintained at 35. The corresponding compression ratios of three images at this quality level are 37, 16, and 26, respectively. Relatively, the Hien method performs the worst in robustness to JPEG compression. Note that the latest generation of video compression algorithm, H.264, uses the same transformation technique, DCT transform, as JPEG does to transform the I-frames, the compression result above is valid for H.264 video files.

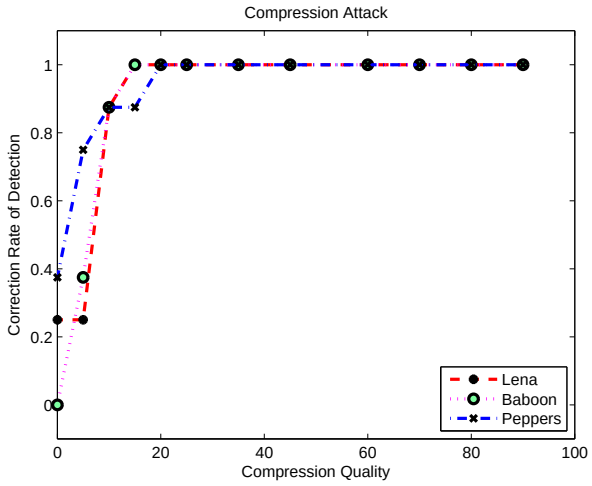
Median Filter Attack

Figures 3.9(g)-(i) indicate the three images after a median filter at size [9 9]. Figure 3.12 shows the robustness results against median filter at different sizes. The correction rate of detection of the proposed method starts falling significantly for Baboon only when the median filter size is up to [9 9]. This figure shows that the detection rate of the Liu method relies still more on images. Among the three methods, the proposed method outperforms the other two.

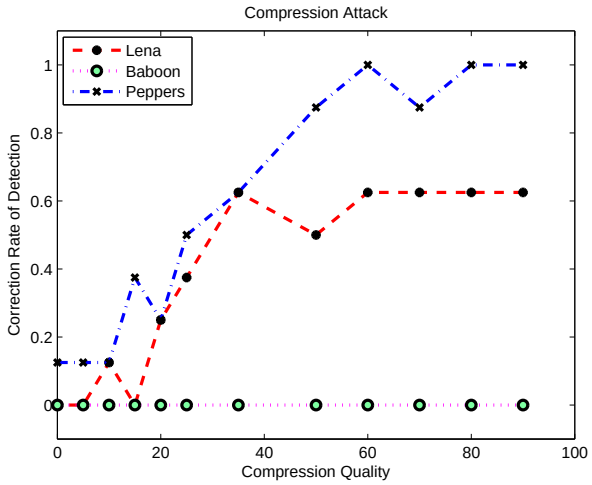
Geometric Attack

Among geometric attacks, image rotation is first tested in the study. The images are rotated by 1 and 2 degrees only. Figure 3.13 shows the robustness results. It indicates that the method does not perform very well for rotation attack. Compared to the other two methods, the proposed method works much better.

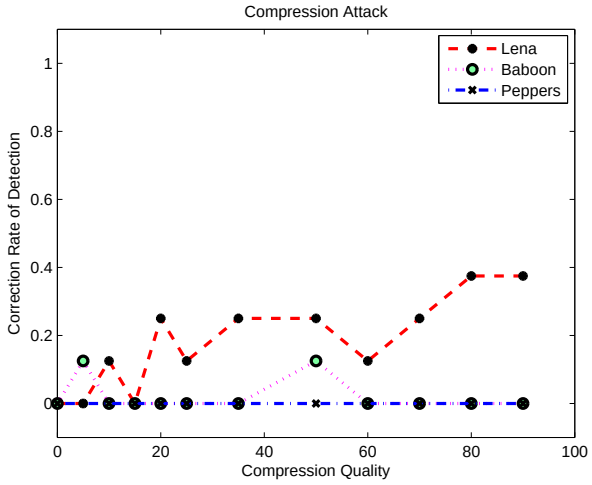
The second geometric attack tested is shift. The images are shifted to the lower right corner by 1, 2 and 3 lines, respectively. Figure 3.14 shows the robustness results. It indicates that the method has better performance when the image is shifted less than or equal to 2.



(a)



(b)



(c)

Figure 3.11: Robustness to JPEG compression. (a) The proposed method. (b) Liu method. (c) Hien method.

Table 3.1: Fingerprint method average robustness on Lena, Baboon and Peppers at size 512×512

Attacks	Correction Rate of Detection		
	Proposed method	Liu method	Hien method
AWGN at SNR=4∼5dB	92%	12.5%	0%
Compress to quality 10	87.5%	8%	4%
Median filter at size [7 7]	91.7%	21%	0%
Rotate 1 degree	75%	8%	4%
Shift 2 lines	79.2%	12.5%	0%
Border crop 5 lines	100%	12.5%	4%

Again, the proposed method has higher robustness to shift than the other two methods.

The last geometric attack tested is border cropping. The four borders of each image were cropped by 1 to 5 lines respectively. Figure 3.15 presents the robustness results. It shows the proposed method has very high robustness against cropping. The other two methods are significantly vulnerable to the border cropping attack.

3.2 Conclusion

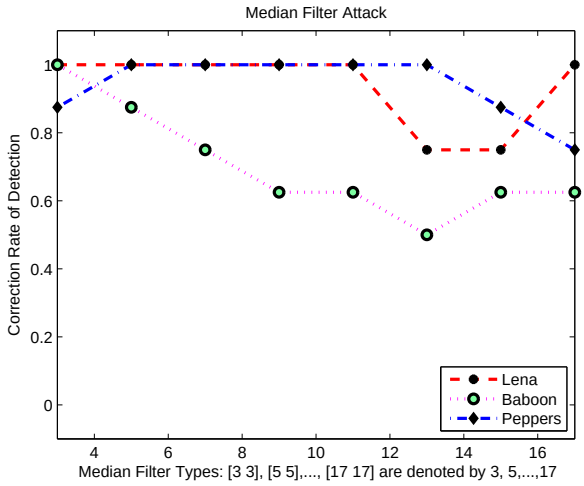
In order to achieve an approximately optimal design of the fingerprint generation, embedding, and distribution suitable for P2P file sharing networks, several key factors had been effectively balanced, such as the robustness, uniqueness, compactness, low computational time. This claim is proofed by testing results on the common attacks.

Due to the space limitation, all the attacks will not be described individually. Instead, the results are summarized in Table 3.1. It indicates that the method is strongly invulnerable to common attacks such as Gaussian white noise, lossy compression, median filter, and border cropping. It also shows that the way of choosing the eigenvectors to generate the fingerprints are very important. In this test, the eigenvectors are chosen based on their principals and their Hamming distance. The fingerprints derived from these chosen eigenvectors outperform the fingerprints derived from all the available eigenvectors.

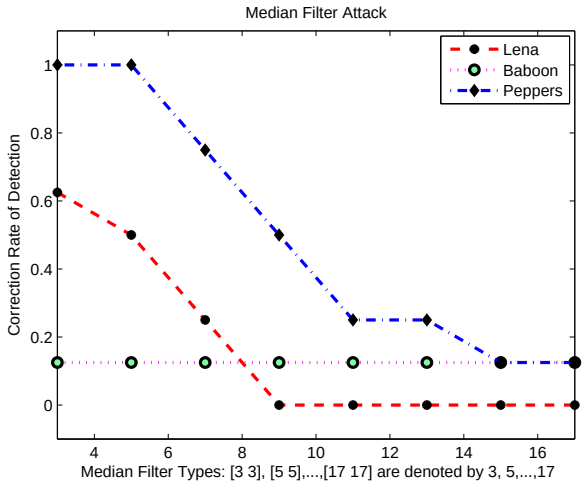
The proposed fingerprint generation method is not only suitable for the watermark fin-

gerprinting applications but also for the watermarking applications. In that case, only the most principal vector needs to be chosen to generate the watermark.

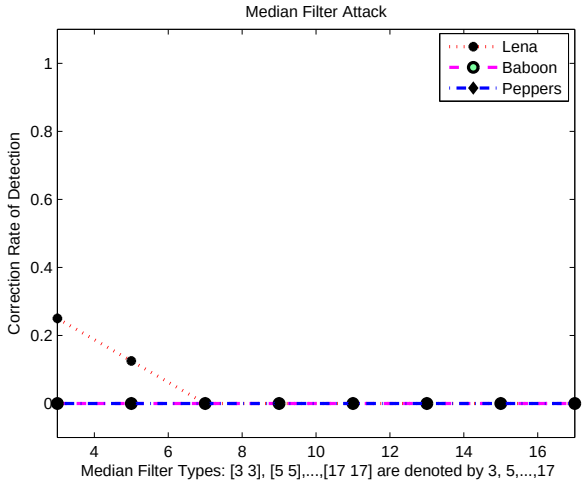
The successful investigation in this Chapter motivates us to think of developing a feature fingerprinting approach as well to enhance the other aspect of DRM, which is legacy content protection. In addition, the concept of a newly available sparse representation technique family provides a possibility of designing a good feature fingerprinting. The next two Chapters will look into these techniques and explore their potentials for the feature fingerprinting by extracting features from the sparse representations.



(a)

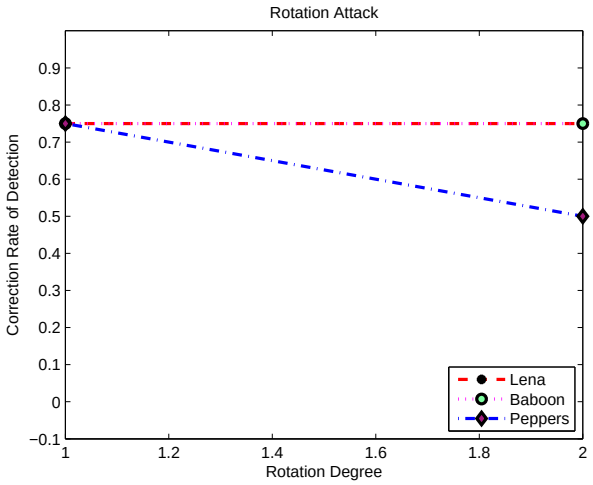


(b)

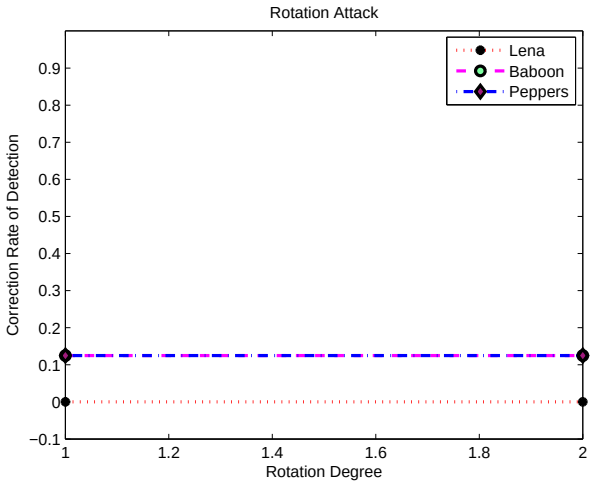


(c)

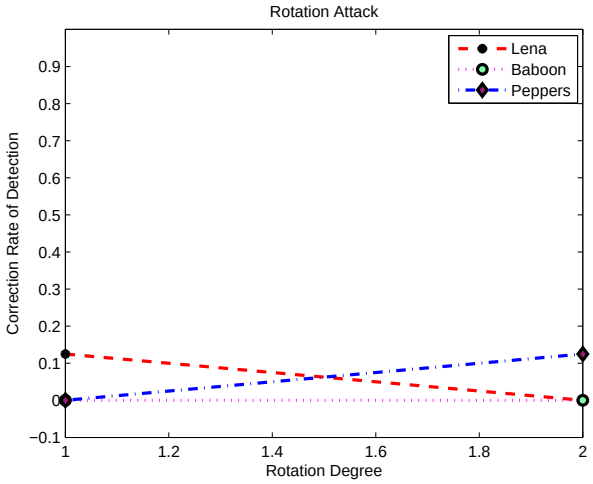
Figure 3.12: Robustness to median filter. (a) The proposed method. (b) Liu method. (c) Hien method.



(a)

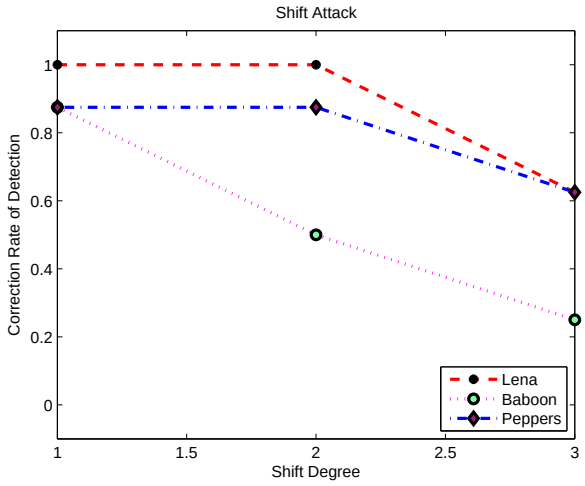


(b)

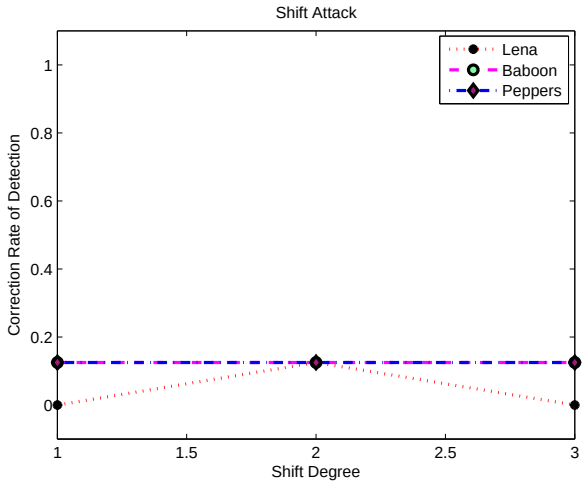


(c)

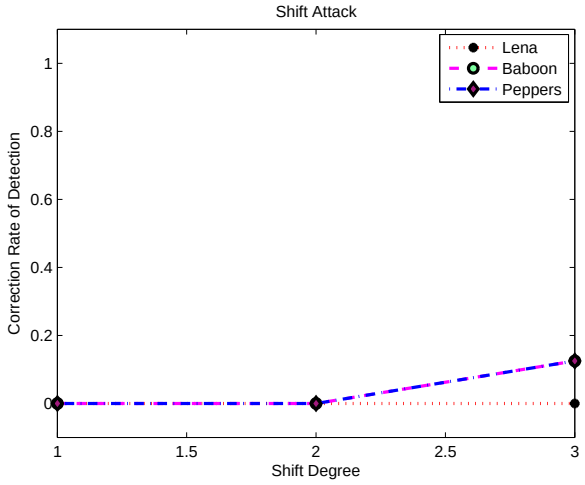
Figure 3.13: Robustness to rotation. (a) The proposed method. (b) Liu method. (c) Hien method.



(a)

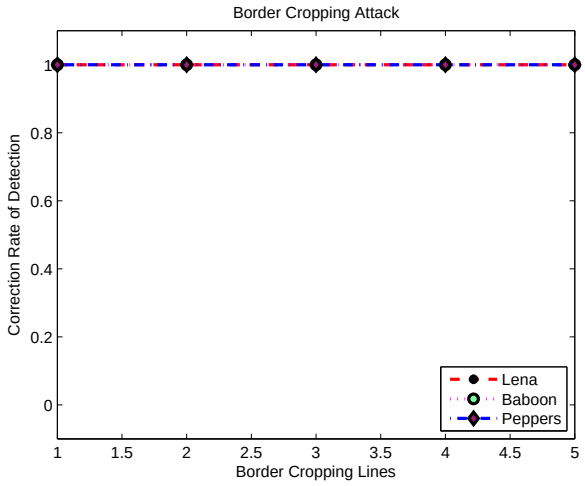


(b)

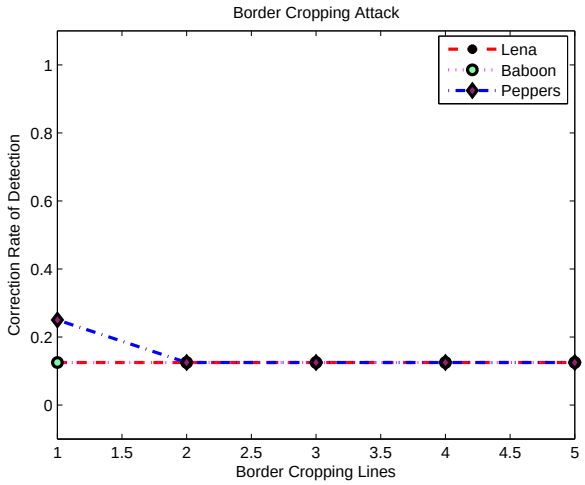


(c)

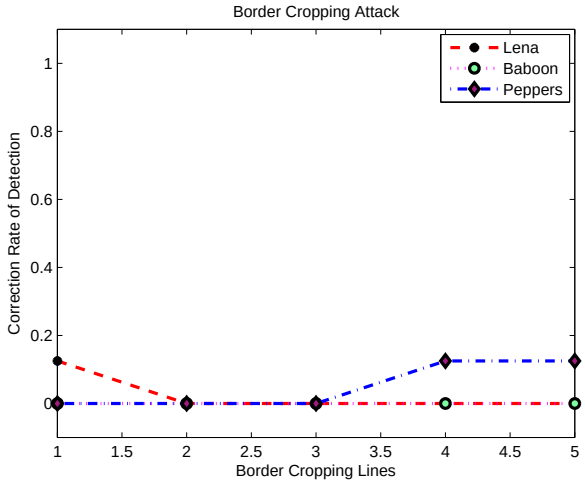
Figure 3.14: Robustness to shift. (a) The proposed method. (b) Liu method. (c) Hien method.



(a)



(b)



(c)

Figure 3.15: Robustness to cropping. (a) The proposed method. (b) Liu method. (c) Hien method.

Chapter 4

MP-based Audio Feature Fingerprinting

4.1 Introduction

Figure 4.1 outlines the procedure of the pre-study of the proposed feature fingerprinting approach. Due to the highly non-stationary and multi-component nature of the audio signals,

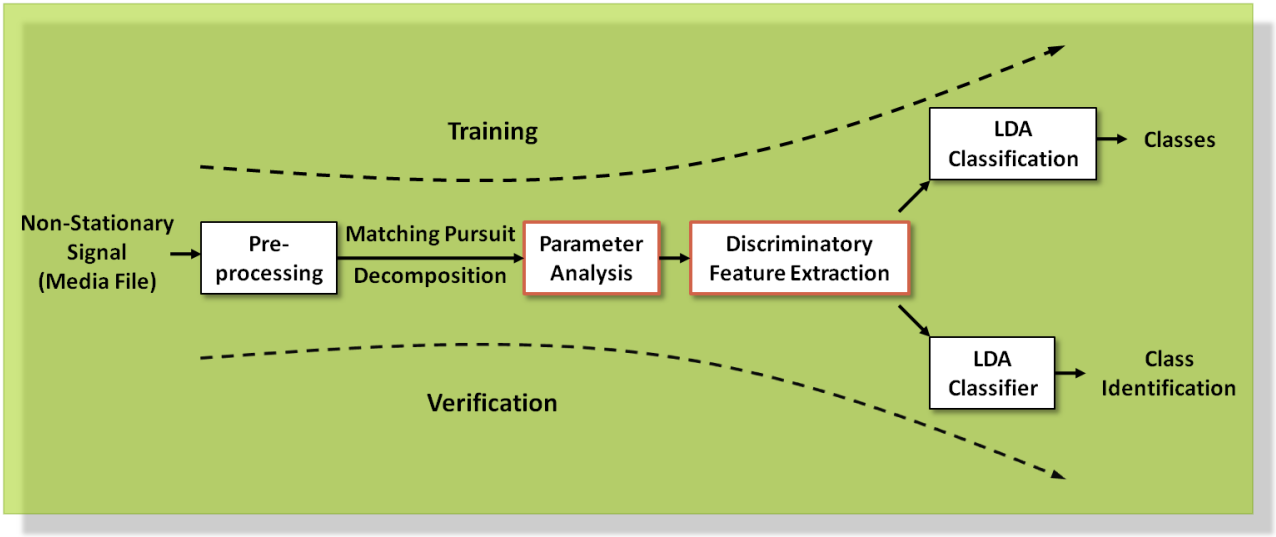


Figure 4.1: Miniature Schematic Diagram the Proposed Approach.

a more flexible and adaptive signal decomposition technique, MP with Gabor dictionary, is utilized to decompose signals and extract the features for classification. This study is the priori-step that will then leads us to next study in Chapter 6 - explore the extended version

of MP, MMP, to extract the features (fingerprints) of audio signals for copyright protection.

In this study, we propose a parametric analysis method to study the atoms obtained from the sparse decomposition and extract the discriminant features from the atom parameters to represent the signal. To the best of the author's knowledge, it is the first approach for classification by analyzing atoms of the sparse representation. Even though the approach in [55] for music classification applies the sparse representation technique as well, it directly utilizes the founded atoms as the features facilitated with the dimension reduction method. However, the size of the features still comparably larger than our approach. In the result section, we will compare the performances of these two approaches.

Figure 4.1 shows the schematic representation of the feature extraction, selection, and classification systems used in our work. Without any signal segmentations, MP decomposes the whole non-stationary signal into atoms, and the efficient classification feature sets are found by analyzing the atom parameters. In order to automatically group signals of same characteristics using the discriminatory features derived, pattern classification is carried out using LDA technique. The leave-one-out method is employed to estimate the correct classification rate with a least bias.

4.2 Signal Decomposition: Matching Pursuit with Gabor Dictionary

4.2.1 Atoms and Dictionary

MP decomposes signals into a linear expansion of atoms which are well localized both in time and frequency. Atoms are selected from a pre-defined over-complete dictionary, in this case Gabor dictionary, which includes functions with a wide range of time-frequency localization and suitable for general decomposition purposes.

The TF basis functions (atoms) in Gabor dictionary are generated by scaling, translating and modulating a single Gaussian function $g(t)$. For any scale $s > 0$, frequency modulation

ξ and translation τ, we denote $\gamma = (s, \tau, \xi)$ and define

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g(\frac{t-\tau}{s}) e^{i\xi t}. \tag{4.1}$$

Since a Gaussian function can be transformed into very different waveforms, the atoms in Gabor dictionary are very flexible and adaptive, and have good time-frequency localization. It makes it possible to approximate a non-stationary signal with an expansion of the atoms selected from Gabor dictionary. Compared to using orthogonal basis to decompose signal, the redundant property of Gabor dictionary utilized by MP performs the sparse decomposition with fewer bases (atoms).

Atoms are selected one by one from the dictionary, while optimizing the signal approximations (in terms of energy) at each step.

4.2.2 Iterative Algorithm

Recall Section 2.4.2 that the original signal x can be written in the form:

$$x = \sum_\lambda \alpha_\lambda u_\lambda \tag{4.2}$$

However, many cases only require to approximate most of the energy of the signal with a small subset of elements. Thus, x can be expressed by N elements and the residual (error) R_N:

$$x = \sum_{i=0}^{N-1} \alpha_\lambda u_\lambda + R_N \tag{4.3}$$

The MP algorithm aims to find a set of atoms and their corresponding coefficients to minimize the residual R_N. The algorithm is given as follows:

1. Initialization step: $R_0 = x$ and $i = 0$. Computation of each coefficient $\alpha_\lambda =< R_0, u_\lambda >$ for all the elements of D;

2. Find the maximum among all the coefficients: $\alpha_{\lambda_i} = \max|\alpha_\lambda|$;

3. New residual calculation: $R_{i+1} = R_i - \alpha_{\lambda_i} u_{\lambda_i}$;

4. Coefficients updating: $\alpha_\lambda =< R_{i+1}, u_\lambda >$

5. Stop criterion: stop if $\alpha_{\lambda_i} < \epsilon$ otherwise $i \Leftarrow i + 1$ and return to step 2.

In our case, the dictionary D will become the Gabor dictionary. Therefore, the elements of the dictionary u_λ are replaced by the elements g_{γ_λ}.

4.2.3 Faster Implementation of Matching Pursuit

The main disadvantage of MP is the high computational complexity required to repeatedly calculate all the inner products and search in the over-complete dictionary for the best atom. In order to lower the computational cost and accelerate the signal decomposition process, the iterative process can be stopped before the residual component will be decomposed completely, and the search for the atoms that best match the signal residue can be limited to a sub-dictionary.

There are two ways to stop the iterative process: one is to use a pre-specified limiting number M of the TF atoms, and the other is to check the energy of the residue $R_M f$. As long as the atoms extracted contain sufficient discriminant information to classify the sample into the pre-set categories, a smaller number of M is preferred. The number of iterations M is selected according to the size of samples and the complexity of classification. The signal decomposition is stopped after extracting the first M TF atoms. In this work, the number of iterations is relatively small, and thus the computational complexity is relatively low.

Considering the size of the samples and the complexity of classifications, a relatively large number of maxima is selected in this algorithm as the sub-dictionary, as long as the parameters obtained are accurate enough for classification.

In this study, MP signal decomposition is implemented using the LastWave signal processing software package [63]. Some explanations about the atom parameters are listed here:

- octave: the scale factor which controls the width of the window function.

- timeId: related to the discrete time samples where the atom is localized.

- freqId: related to the center frequency of the atom.
- chirpId: the chirp-rate of the atom. It is always "0" in this experiment.
- innerProdR: the real part of the inner-product between the signal and the atom.
- innerProdI: the imaginary part of the inner-product between the signal and the atom.
- phase: used for combining multiple atoms.
- g2Cos2: always "0" in this experiment.
- realGG: the real part of the inner-product between the complex atom and its conjugate. It is always "0" for most of the atoms in this experiment.
- imagGG: the imaginary part of the inner-product between the complex atom and its conjugate. It is always "0" for most of the atoms in this experiment.
- energy in atom: energy in atom. The first extracted atom contains the largest energy.
- coeff2 of atom: equals to energy in atom in this experiment.

4.3 Application in Music Classification

4.3.1 Music Sample Processing

For investigating the MP-based feature fingerprinting for music classification, two classes, rock-like music and classical-like music are chosen. The database is comprised of 112 pieces of music samples with 56 rock-like music and 56 classical-like music samples, and each sample has the duration of 10 seconds. All the samples fall into two categories, that is, rock-like music group (7 sub-groups with 8 pieces of 10-second clips in each subgroup), and classical-like music group (7 sub-groups with 8 pieces of 10-second clips in each subgroup) experiment. The number of iterations of the pursuit is increased to 3000 to get more detailed information for effective classifications. Thus, the book for each signal ends up with 3000 atoms in it,

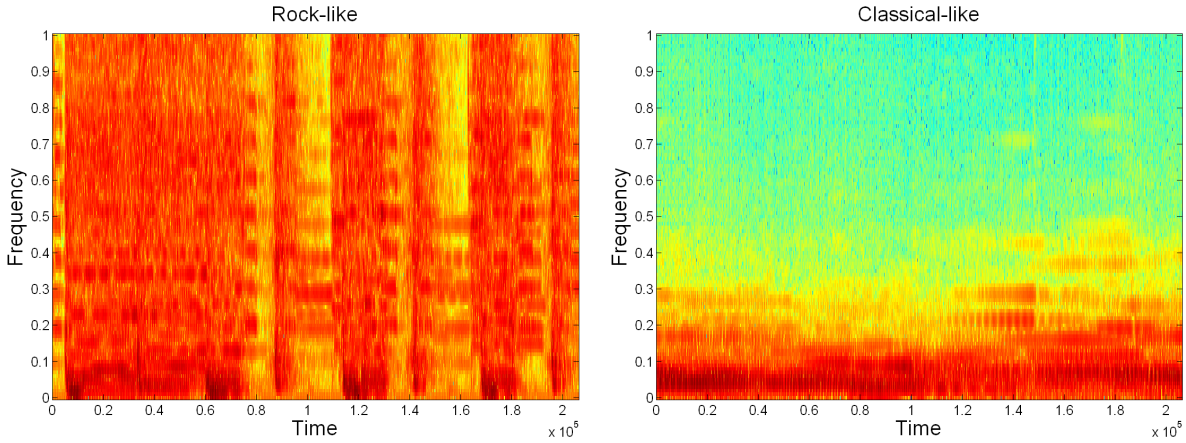
except if the pursuit stops before because the residue is zero, which has not happened in the experiment.

Since for classification purpose somewhat general characteristics of signals in a broad sense is sufficient, the fast implementation of MP is employed. The number of pursuit iterations is pre-set to control the decomposition process, and local maxima are used to limit the searching area. While ensuring that the atoms extracted from each music sample are sufficient for a satisfactory classification, we try to use fewer pursuit iterations and larger local maxima, to reduce the computational complexity as long as satisfying classification results can be obtained. In this experiment, a set of 300 maxima is selected for each iteration and the first 2000 atoms are analyzed to find the optimum classification feature set.

4.3.2 Parameter Analysis and Discriminatory Feature Extraction

In order to look more into the characteristics demonstrated by rock-like music samples and classical-like music samples, and define the discriminatory features for classification, the spectrograms of the samples are also studied. A spectrogram is the squared modulus of the STFT and is generally used to display the TF energy distribution over the TF plane. From the spectrogram plots, it is easy to observe that in general the energy distribution is different for rock-like and classical-like music samples. It was found that rock-like music samples usually contain higher energy components. In [72], Umapathy et al. studied the MP decomposition algorithm and observed that the octave distribution can reflect the spectral similarities for the same category of signals. Since rock-like music samples and classical-like music samples demonstrate different categorical characteristics with regard to the spectral energy distribution, it is expected that the octave parameter may carry distinguishing information to separate rock-like music samples from classical-like ones. Spectrograms of one rock-like and one classical-like music sample are randomly selected from the database and plotted in Figures 4.2(a) and 4.2(b), to show the visible differences of the spectral energy distribution between the two groups.

Knowing the distribution of octave may contain important discriminating information for



(a) Spectrogram of a rock-like music signal. X-axis: time samples. Y-axis: normalized frequency where maximum frequency corresponds to sampling frequency/2. Colors indicate different energy levels, with blue the lowest and red the highest.

(b) Spectrogram of a classical-like music signal. X-axis: time samples. Y-axis: normalized frequency where maximum frequency corresponds to sampling frequency/2. Colors indicate different energy levels, with blue the lowest and red the highest.

Figure 4.2: The spectrograms of rock-like music and classical-like music.

classification, this parameter, along with its derivative values such as the standard deviation of octaves in the first 2000 atoms, the mean of octaves in the first 2000 atoms, and the median of octaves in the first 2000 atoms, has been studied. The octave and/or its derivatives are selected into the test feature sets for music group classification. The optimum feature set, which brings up the best classification accuracy, is found to be: the standard deviation of octaves in the first 2000 atoms.

4.4 Classification Scheme

4.4.1 LDA

In this work, pattern classification is carried out using the LDA technique in Statistical Package for the Social Sciences (SPSS) [73]. To distinguish among the groups, a set of discriminating features are selected which measure characteristics in which the groups are expected to differ. LDA method tries to find one or more linear combinations of a set of discriminating features that best separate the groups of samples.

The procedure automatically chooses a first function that will separate the groups as

much as possible. It then chooses a second function that is both uncorrelated with the first function and provides as much further separation as possible. The procedure continues adding functions in this way until reaching the maximum number of functions.

4.4.2 Leave-One-Out Method

In this study, the classification accuracy is estimated using the leave-one-out method which is known to provide a least bias estimate. In the leave-one-out method, one sample is excluded from the dataset and the classifier is trained with all the remaining samples. Then the excluded sample is used as the test data and the classification accuracy is determined.

This operation is repeated for all samples in the dataset. The number of correctly classified cases is used to calculate the classification accuracy rate. Since each sample is excluded from the training set in turn, the independence between the test set and the training set is maintained. In a database with N examples, N experiments are performed. For each experiment, $N-1$ examples are used for training and the remaining example is used for testing. The number of correctly classified subjects is counted to estimate the classification accuracy rate. The true error is estimated as the average error rate on test examples:

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i. \tag{4.4}$$

4.5 Classification Results and Conclusion

4.5.1 Classification Results

The values of standard deviation of octaves in the first 2000 atoms are listed in Table 4.1. By observation, the threshold of 1.7 is assigned, which can completely separate the rock-like music samples from the classical-like music samples. When the standard deviation of octaves in the first 2000 atoms is smaller than 1.7, the music sample is classified into classical-like music group. When the standard deviation of octaves in the first 2000 atoms is larger than 1.7, the music sample is classified into rock-like music group. The classification accuracy is 100%.

Table 4.1: Standard deviation of octaves in the first 2,000 atoms of each music sample. The four numbers in each row correspond to the four music samples respectively.

Music Sample	Standard Deviation of Octaves			
Classical 1-4	1.2109	1.1631	1.2701	1.4257
Classical 5-8	1.5357	1.4144	1.0916	1.2308
Classical 9-12	1.0760	1.2239	1.4580	1.1023
Classical 13-16	1.2622	1.1759	1.4090	1.5346
Classical 17-20	1.4979	1.4900	1.4958	1.5222
Classical 21-24	1.4492	1.6053	1.4742	1.3996
Classical 25-28	1.3389	1.2897	1.2771	1.2380
Classical 29-32	1.2351	1.2903	1.3520	1.3613
Classical 33-36	1.3665	1.2858	1.2777	1.1167
Classical 37-40	1.3031	1.4725	1.2384	1.1055
Classical 41-44	1.1702	1.1286	1.1718	1.1266
Classical 45-48	1.3096	1.1946	1.4924	1.1853
Classical 49-52	1.2886	1.1800	1.2341	1.1556
Classical 53-56	1.1894	1.2725	1.3664	1.3428
Rock 1-4	2.1355	2.3155	2.1863	2.0359
Rock 5-8	2.0105	1.9743	2.0570	2.2351
Rock 9-12	2.5278	2.5570	2.3779	2.1647
Rock 13-16	2.2028	2.2540	2.1758	2.0557
Rock 17-20	1.9922	2.0358	2.0630	1.7830
Rock 21-24	2.0853	1.9753	2.0233	1.9941
Rock 25-28	2.0534	1.9518	1.9035	1.9630
Rock 29-32	2.0667	1.8370	1.8492	1.8096
Rock 33-36	2.1048	1.9141	1.8272	1.7141
Rock 37-40	2.0565	2.0237	1.9021	1.7591
Rock 41-44	2.7277	2.5827	2.3621	2.6165
Rock 45-48	2.5539	2.6482	2.6736	2.3581
Rock 49-52	2.4693	2.3978	2.2018	2.1915
Rock 53-56	2.2678	2.1218	2.0843	2.1882

4.5.2 Conclusion

Given the implementation comparison between the approach [55] and ours, it is necessary to compare the classification performances of these two methods. To be fair, the feature

dimension is set to be the same, i.e., 12. This is based on our scheme where only 12 atom parameters are utilized. According to this dimension of feature set, the best classification accuracy of the scheme [55] is below 70%, while our scheme is above 90%.

The experiments on the music databases verify again that MP, as an adaptive time-frequency tool, decomposes non-stationary signals into atoms whose parameters contain good discriminant information for classification. The study further proves that the octave has the discriminatory ability to classify audio signals.

The sparsity and the discriminant features of MP are key factors in designing a good fingerprinting scheme. The result above leads us to the next level, further exploring such possibility from its variant, which is explored in the next Chapter.

Chapter 5

MMP-based Audio Feature Fingerprinting

5.1 Introduction

Figure 5.1 demonstrates the procedure of the proposed feature fingerprinting approach.

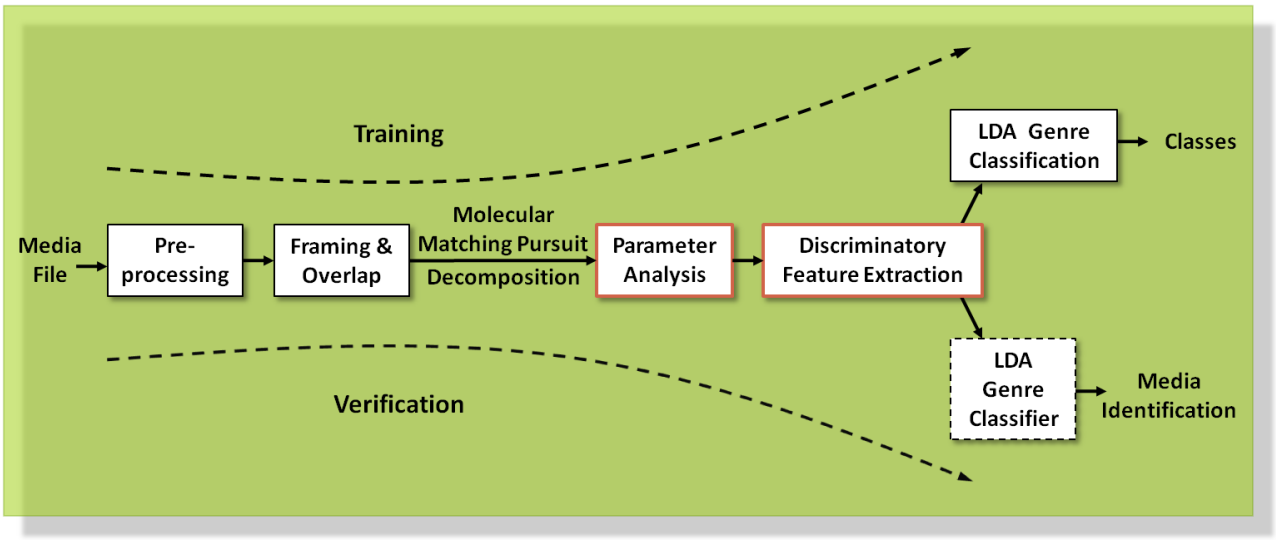


Figure 5.1: Miniature Schematic Diagram of the Proposed Approach.

This Chapter will concentrate on describing an audio feature fingerprinting (fingerprint extraction) scheme, which can be used for the legacy content identification. When a query of a media file is submitted to the identification system, the file will be first translated as a fingerprint using the proposed scheme. It will then be compared with the set of fingerprints in

the database. If a fingerprint in the database matches the target fingerprint with the highest similarity, the meta-data information associated to the found fingerprint can be utilized as the proof that the legacy of this media file has been registered and copyrighted. Recall that the watermarking technique can not be applied to identify the legacy content in that watermarks must be embedded in multimedia during the production, and therefore cannot be used to identify multimedia that are already in circulation.

Previous Chapter shows that MP provides a way of sparse representation of a signal and the parameter-analysis based on the MP atoms performs well in the classification of two music genres. Its computational complexity, however, makes it harder to be implemented for fingerprinting in a large scale. In this Chapter, an improved audio fingerprinting scheme is proposed, which uses one of the suboptimal sparse representation algorithms - MMP algorithm [57] to obtain the approximate representation atoms of the original signal and further extract the features out of them as the fingerprint of the signal. As introduced in Section 1.6.1, MMP was first proposed for audio decomposition by Daudet [57]; Parvaix et al. [58] then adopted the decomposed atoms by means of MMP, combined with the psychoacoustic model, as the watermarking platform. Our study, on the other hand, is the first trial that parametrically analyzes the MMP derived atoms and explores the most representable features (sparse features) in lower dimension to become qualify as signal fingerprint. Moreover, the robustness of the proposed scheme against various common attacks was considered in the process of searching the features. The robustness of the scheme was tested by distorting the original files using the common attacks. The cross-correlation approach is used to extract the fingerprint of the distorted file to identify the source of the file. The results showed that the proposed scheme can accurately identify the sources of the files with high probability. LDA is also used to improve the computational complexity of the matching algorithm.

The detailed descriptions of the principle of the MMP algorithm and the definition of the to-be-extracted features and the rationale are given in Section 5.2. Section 5.4 describes fingerprint matching methods. The Chapter is ended with evaluating the performance of the overall scheme for various distortions and the comparison with a classical audio feature

fingerprinting algorithm in Section 5.5.

5.2 Signal Decomposition: Molecular Matching Pursuit

Besides deriving a sparse representation of the original signal, MMP algorithm speeds up the matching pursuit by having an easier control of the base dictionary. The principle of MMP and the reason that MMP is suitable for audio feature fingerprinting [57] will be explained shortly.

In Chapter 5, it was found that the MP algorithm incurs high computational cost. In particular, in the case of a large dictionary, Steps 2 and 4 of MP can be very long because these two steps need to go through each element of D one by one. The MMP algorithm was proposed to reduce the computational cost. As indicated in Section 2.4.4, the MMP algorithm is to group atoms with similar time-frequency properties to form molecules so that several atoms can be subtracted from the residual at the same time for each iteration.

Moreover, for MP decompositions, the localization of the selected atoms in the time-frequency/time-scale planes is not uniform but reveals some of the intrinsic structure of the analyzed signal. In contrast, for MMP decompositions, the localization of the selected atoms in the same class(i.e., belonging to the same orthonormal basis) are within one area with neighboring time-frequency/time-scale parameters. This simple structure of atoms can offer a significant advantage in terms of coding cost from a signal compression perspective. On the other hand, the molecules also provide some relevant information about the structure of the signal. The suboptimality of MMP compared to MP is compensated by a better description of the structure of the signal and its simple and faster implementation.

The idea of selecting groups of atoms at each iteration was already developed in another extension of MP, called Harmonic Matching Pursuit algorithm [74], which looks for harmonic structures made of (quasi) harmonically related Gabor atoms. One of the main drawbacks of using this algorithm is that a large number of musical sounds are not harmonic (e.g., percussive sounds). For most audio signals, a given harmonic portion cannot simply be

described by a single Gabor atom, for it may have a sharp or a very slow attack transient; and therefore a given note may require a potentially large number of harmonic atoms. Daudet [57] showed that MMP provides better results than Harmonic Matching Pursuit. The MMP approach [57] claims that the local time-frequency/time-scale grouping is a stronger and more robust assumption about the structure of real audio signals than the harmonicity, because MMP considers dictionaries made by concatenation of a small number of orthonormal bases that are sufficiently incoherent. This is a major difference from the Harmonic Matching Pursuit, where the dictionary is made of a large number of very coherent atoms that make it more difficult to consider local time-frequency/time-scale structures.

In order to obtain most of the energy of the initial signal concentrated in a small number of elements, the choice of an appropriate dictionary is very important.

For obtaining better decomposition of audio signal, MMP utilizes the two elementary components, the tonal part (sum of sinusoids) and the transient part (sum of Diracs), to model the most of the audio signals. That is, the MDCT atoms are used to represent the tonal part, and the DWT atoms are used to represent the transient part. Therefore, MDCT with atoms u_λ and DWT with atoms v_λ are used to construct a 2-times redundant dictionary $\mathcal{D}$ for MMP, with $\mathcal{D} = \mathcal{C} \cup \mathcal{W}$, where $\mathcal{C}$ is an orthogonal basis of lapped cosines (also called an MDCT basis), and $\mathcal{W}$ is an orthogonal basis of discrete wavelets.

Let us review the procedure of the algorithm:

1. Initialization: $R_0 = x$, and $i = 0$. Compute each MDCT coefficient $c_\lambda =< R_0, u_\lambda >$ for all the elements of $\mathcal{C}$ and each DWT coefficient $w_\lambda =< R_0, v_\lambda >$ for all the elements of $\mathcal{W}$;

2. Compute the molecule index $\mathcal{T}$ that a set of u_λ defines and the molecule index $\mathcal{K}$ that a set of v_λ defines; find $K = \max\mathcal{K}$ and $T = \max\mathcal{T}$.

3. Identify the most significant structure. If $T \geq K$, then the most significant structure is of type "tonal molecule"; otherwise $K > T$, it is of type "transient molecule".

4. For a tonal molecule, identify atoms that define the most significant tonal molecule: $M_i = \sum_{\lambda=1...m_i} c_\lambda$. Update the residual: $R_{i+1} = R_i - \sum_{\lambda=1...m_i} c_\lambda u_\lambda$. Update the coefficients from the new residual: $c_\lambda =< R_{i+1}, u_\lambda >$.

   For a transient molecule, identify atoms that define the most significant transient molecule: $M_i = \sum_{\lambda=1...m_i} w_\lambda$. Update the residual: $R_{i+1} = R_i - \sum_{\lambda=1...m_i} w_\lambda v_\lambda$. Update the coefficients from the new residual: $w_\lambda =< R_{i+1}, v_\lambda >$.

5. Stop criterion: stop if $\max(K, T) < \epsilon$, otherwise $i \Leftarrow i + 1$ and return to step 2.

Notice that step (4) is to group atoms within a certain window around the significant molecule found.

The main advantages of this algorithm for feature fingerprinting are evidently the reduction of computational cost, and the capability of the extraction of very descriptive features so that the fingerprint fulfills the uniqueness requirement.

An example of a decomposition of an audio signal is shown in Figure 5.2, where we took a 5 sec-duration signal sampled at 44.1kHz. The reconstructed signal is obtained after 10 iterations and contains more than 18.5% of the energy of the initial signal.

5.3 Parameter Analysis and Discriminatory Feature (Fingerprint) Extraction

Here we extracted features from MDCT coefficients because they contained more information about the audio file than DWT coefficients. The MDCT function is given by:

$$X(k) = \sum_{m=0}^{n-1} x(m)\cos(\frac{\pi}{2n}(2m + 1 + \frac{n}{2})(2k + 1)),$$
$$for\ k = 0 \sim \frac{n}{2} - 1, \quad (5.1)$$

where $x(m)$, the overlap-segmented frame with length n derived from the original signal, is obtained by KBD windows. A time-frequency matrix A is then derived by the MDCT transform. Thus, for each iteration i, MMP provides an MDCT matrix A_i. An example

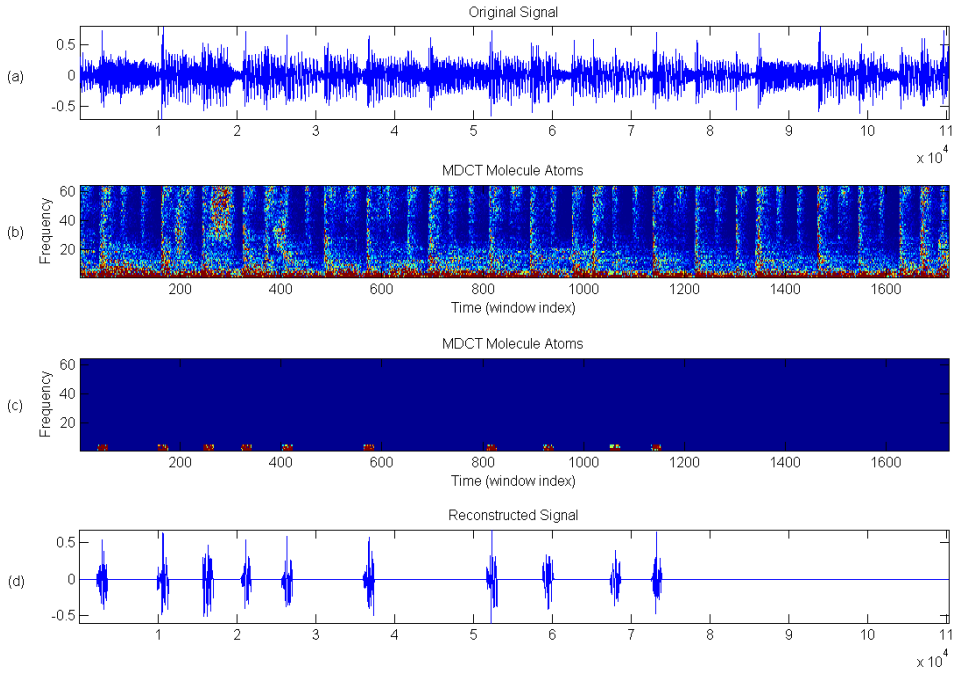


Figure 5.2: Example of decomposition with MMP algorithm, a) the original music signal, b) the MDCT coefficients of the signal, c) the molecule atoms after 10 iteration, and d) the reconstructed signal based on the molecule atoms in c).

of these MDCT matrices is given in Figure 5.3. The coordinates of each coefficient in the matrix gives information about time and frequency. As explained in Section 5.2, the matrix M contains the coefficients forming the molecule that contributes the most energy to the signal.

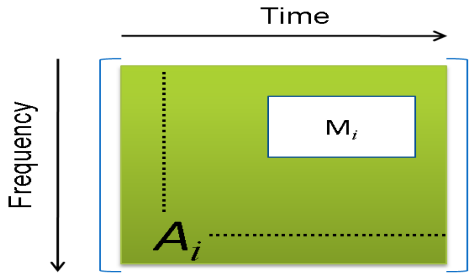


Figure 5.3: Example of decomposition with MMP algorithm

The higher the iterations process is, the more precise is the representation of the initial signal. The features we found do not require a very large number of iterations. Only a small

number of iterations, for example 10 iterations, were taken into account for simplicity's sake and because they contain the most relevant information about the file. Let N be the number of iterations and L be the number of extracted features. In the ith iteration, the feature vector $\mathcal{F}_i = [f_j]_{j=1...L}, i = 1...N$ is derived. Then the mean of these vectors after N iterations is defined as $\mathcal{F}_{avg} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{F}_i$. It is processed in the same manner for the variance to get $\mathcal{F}_{var} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{F}_i - \mathcal{F}_{avg})^2$. So at the end, a complete fingerprint $\mathcal{F} = \mathcal{F}_{avg} \cup \mathcal{F}_{var}$ forms a vector with $2L$ elements. In this experiment, for each iteration, 14 feature elements ($L = 14$) are generated. Therefore, the file fingerprint is a 28-dimensional vector. The small size makes it easy to compute, which meets the compactness requirement.

The fourteen features from the ith iteration were extracted as follows:

- maximum value of the MDCT coefficients and its logarithm as defined below:

$$f_1 = \max(M_i) \tag{5.2}$$

$$f_2 = \max(\log(M_i)) \tag{5.3}$$

- ratio between previous and current MDCT coefficients:

$$f_3 = \frac{\max(M_i)}{\max(M_{i-1})} \tag{5.4}$$

$$f_4 = \frac{\max(\log(M_i))}{\max(\log(M_{i-1}))} \tag{5.5}$$

- modeling the non-zero coefficient coordinates (h_{M_i}, v_{M_i}) of the ith molecule matrix M_i as the frequency and time parameters of a MDCT cosine function, so the feature is the mean of the summation of all the modeled coordinates:

$$\begin{aligned} f_5 &= \overline{\sum \cos(h_{M_i}, v_{M_i})} \\ &= \overline{\sum \cos(c^1(h_{M_i} + c^2)(v_{M_i} + c^3))} \end{aligned} \tag{5.6}$$

where c^1, c^2, and c^3 are constants;

- the dominant principal component, i.e., the eigenvector that corresponds to the highest eigenvalue of the frequency matrix windowed by $\mathcal{W}1_{4\times21}$: $E(\mathcal{W}1_{4\times21})$ with the four elements, where $E(\cdot)$ stands for the derived most principal eigenvector p of the covariance of the matrix $\mathcal{W}1_{4\times21}$; this window centers on the frequency that matches the highest MDCT coefficient. Given the equation:

$$\mathcal{W}1_{4\times21} \times \mathcal{W}1_{4\times21}^{T} \times p = \lambda_p \times p, \tag{5.7}$$

  where λ_p is the most principal eigenvalue, these features are depicted as:

$$\begin{aligned} f_{6\sim9} &= E(\mathcal{W}1_{4\times21}) \\ &= p; \end{aligned} \tag{5.8}$$

- the elements in the autocorrelation of the frequency matrix windowed by $\mathcal{W}2_{2\times21}$, which centers on the frequency that matches the highest MDCT coefficient - this feature is denoted by:

$$f_{10\sim13} = \mathcal{W}2_{2\times21} \times {\mathcal{W}2_{2\times21}}^{T}, \tag{5.9}$$

  which has four elements as well;

- the position of the most significant MDCT coefficient is presented as:

$$f_{14} = s(\max(M_i)) \tag{5.10}$$

  where $s(\cdot)$ represents deriving the segment index of the coefficient.

Therefore, the set of the features for the ith iteration forms the vector $[\max(M_i), \max(\log(M_i)), \frac{\max(M_i)}{\max(M_{i-1})}, \log\frac{\max(M_i)}{\max(M_{i-1})}, \overline{\sum\cos(h_{M_i}, v_{M_i})}, E(\mathcal{W}1_{4\times21}), \mathcal{W}2_{2\times21} \times {\mathcal{W}2_{2\times21}}^{T}, s(\max(M_i)]$.

The following points will further explain each item in the feature vector:

- the first two items in the vector are two types of computational methods of the feature - maximum value of the MDCT coefficient, and they aim for enhancing the feature strength;

- the third and fourth items actually describe the time relationship between two significant atoms in the previous iteration and the current iteration. Since the first number of iterations captures the significant atoms, the more stable these atoms are, the more robust these features will be. This feature, to some extent, also obey the statement by Haitsma et al. in [31] that the sign of energy differences is a property that is very robust to many kinds of processing.

- the fifth item maps the molecular coordinates to an average frequency value in order to measure the stability of the molecular coordinates;

- the sixth item presents the major shape of the matrix of the frequencies corresponding to the selected molecule by calculating its principal eigenvector;

- the seventh item emphasizes the dominant frequency values in each iteration by calculating their autocorrelation matrix;

- the last item in the feature vector precisely presents which segment of the signal the significant MDCT coefficient belongs to by extracting the column index of the coefficient, because the column index matches the segment of the signal.

Notice that we do not use the time information. Indeed, because of the uncertainty principle of Heisenberg explained in [60][61], it is impossible to obtain a high resolution for time and frequency at the same time. Actually, the frequency coordinate gives a precise information while the time coordinate provides only the number of the temporal windows used by the MDCT.

The experiment shows that, one particular molecule is very possibly selected in the ith iteration from the source signal but selected in the $(i \pm j)$th iteration from the attacked signal. Usually, j is a very small number. It implies that there are other molecules contain the similar contribution to the signal. Therefore, the features correspondingly have the same issue, i.e., the order of their values appearing over the number of iterations from the source signal and the attacked signal are not exactly the same. However, after the summation

and average operations, the values will approximately remain the same before and after the attacks.

5.4 Fingerprint Matching

In audio feature fingerprinting, the main problem for the matching part is to find a matching algorithm with low computational cost. Our proposed method uses the combination of cross-correlation (CC) for its efficiency and LDA algorithm for its speed.

The CC algorithm is well adapted to compare fingerprints as it gives a measure of similarity. Between two real signals x and y, the cross-correlation coefficient is computed with the following equation:

$$C_{xy}(n) = \sum_{j} x(j)y(n+j) \tag{5.11}$$

The more the two signals are correlated, the higher the final result is, so that we just need to find the strongest likelihood by comparing the new fingerprint with all the fingerprints of the database.

However, it can take a very long time to go through each element of the database with the CC algorithm, especially when this database is large. In order to reduce this computational cost, we choose LDA for genre(or inter-class) classification. The reason why LDA is chosen is under the assumption that the features are linearly separable. Our results prove this point as well.

Whereas, if the features of the fingerprints provided are not uncorrelated enough to distinguish the number of classes requested, the LDA classifier sometimes fails and gives a wrong file belonging to a wrong class. Thus, to fix this problem, an empirical confidence level threshold of the final match can be given so that if the LDA algorithm fails, the system returns to the step of the global matching until it finds a sufficient correlation.

In other words, LDA plays the role of a classifier for the fingerprints of the database. As we used three types of music (classical, pop and rock) in this scheme, the matching search

is made in two steps:

1. Find the class of the file by looking for which side of the boundary gives the degree of confidence greater than the confidence level threshold (the empirical number is 0.95), and go to (a) as the next step; otherwise go to (b) as the next step.

2. Find the matched file:

   (a) Within this class, search for the file by seeking the fingerprint which provides the highest likelihood and stop.

   (b) Among the classes, search for the file by seeking the fingerprint which provides the highest likelihood and stop.

An overview of the entire system is shown in Figure 5.4.

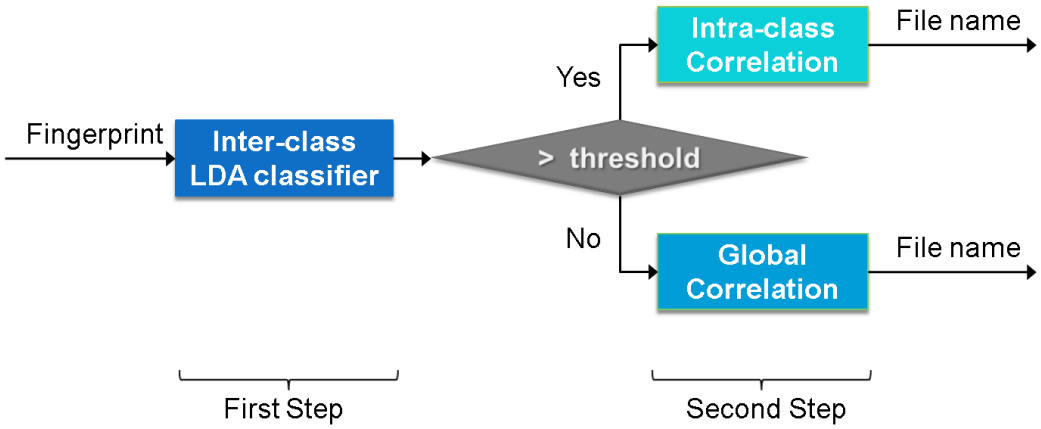


Figure 5.4: Fingerprint matching

The advantage of using LDA is not only that it classifies the musical genre, but also that the computational time is much less compared to the CC-based global matching search. More precisely, the search time can be reduced by a factor of n, where n is the number of classes.

5.5 Application in Music Identification

5.5.1 Music Sample Processing

At the beginning of the feature fingerprinting, there are a couple of preparation steps need to be done before the signal decomposition and fingerprint (feature) extraction as discussed in Section 1.5.2. Next paragraph introduces the preparations for this study.

All the audio data were downloaded from a benchmark database GTZAN genre collection [75]. Three types of audio, i.e., classic, pop and rock, and twenty files for each type were collected for this study. Each audio signal to be tested has a length of 5 seconds with 352 kbps sample rate, and each sample has 16 bits. Therefore, the total number of the signal sequence will be $352k \times 5/16 \approx 110000$. The sequence of the numbers will then be segmented by a length of 64. Thus, each segmentation is called M samples where $M = 64$. All these consecutive sets of M samples are ready to be processed by next stage-signal decomposition and fingerprint extraction. As a fingerprint is a 28-dimensional vector, the database becomes a 60-by-28 matrix. Even though the data set size is limited, the leave-one-out cross-validation technique has been used to make the accuracy result more generally reliable.

5.5.2 Attacks

In this section, we will study the robustness of the proposed scheme to the distortion. Three types of attacks were tested:

- Low Pass Filtering: a filter with an order 23 and a normalized cut-off frequency 0.3;
- Additive Noise: the signal-to-noise ratio is always set to approximately 15 dB;
- Compression: the original files with 352kbps are compressed to a bit rate of 16 kbps via MP3 processing (compression ratio$\approx 20:1$).
- Compression: the original files with 352kbps are compressed to a bit rate of 128 kbps via MP3 processing (compression ratio$\approx 3:1$).

Figures 5.5, 5.6 and 5.7 illustrate examples of comparisons between MDCT coefficients before and after attacks of the signal. Figure 5.8 shows the fingerprint robustness to the attacks on one music file.

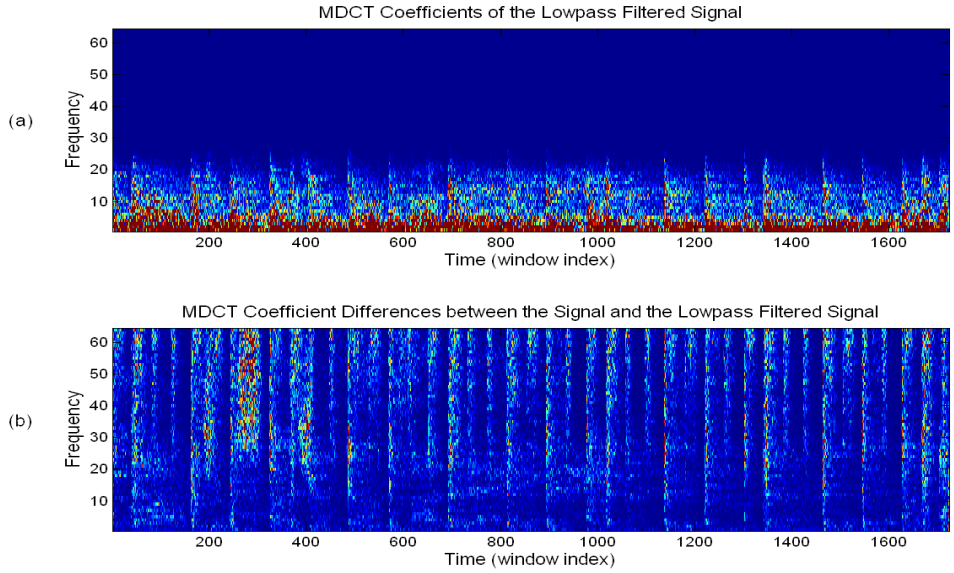


Figure 5.5: MDCT coefficients after low pass filter. (a) MDCT coefficients of the low pass filtered signal. (b) MDCT coefficient differences between the original signal and the low pass filtered signal.

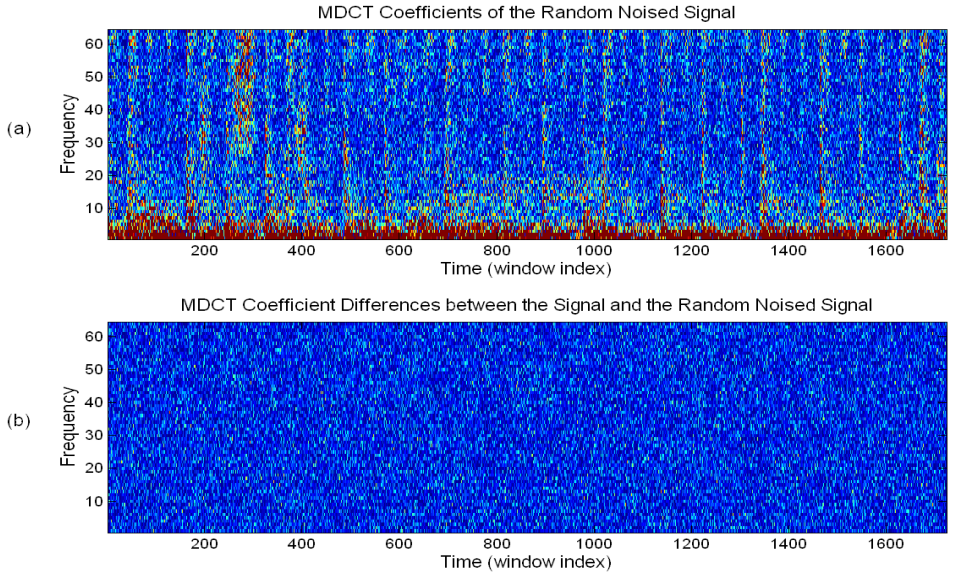


Figure 5.6: MDCT coefficients after random noise. (a) MDCT coefficients of the noised signal. (b) MDCT coefficient differences between the original signal and the noised signal.

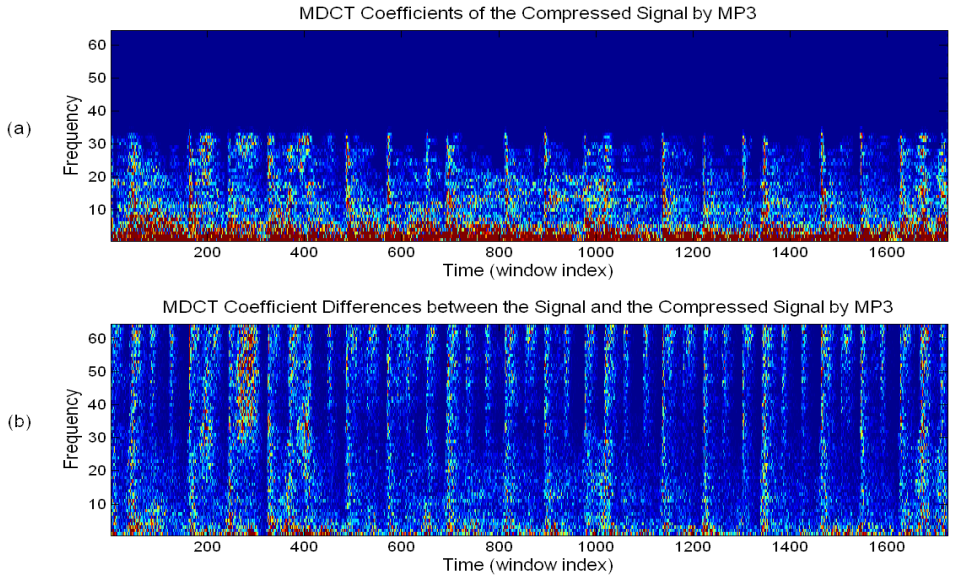


Figure 5.7: MDCT coefficients after MP3 compression. (a) MDCT coefficients of MP3 signal with bit rate 16kbps. (b) MDCT coefficient differences between the original signal and the MP3 signal.

5.5.3 Identification Results

The results are shown in Tables 5.1 and 5.2. Both tables indicate that the most problematic attack is MP3 compression with bit rate at 16 kbps. Compared to the original bit rate at 352 kbps, this compression ratio is extremely high, which is very unlikely to happen. Instead, we also test the MP3 compression at the popular bit rate of 128 kbps, and the identification accuracy is over 95%. These two results show the robustness of the approach even under high compression attack.

The results shown in Table 5.1 are obtained by utilizing the LDA classifier combined with correlation coefficients upon the features as mentioned in Section 5.4. To get the parameters of the classifier, the un-distorted features are used as training samples. The distorted features, as the testing samples, then can be classified to a class based on if its confidence level with this class is the highest and higher than the threshold 0.95. This threshold is chosen based on empirical evidence. The overall inter-class classification accuracy of our approach is 87.9% which is higher than the accuracy 75% in [55] where if the dimension of the features is equal to the dimension of features in our approach. The scheme in [55] is comparable to

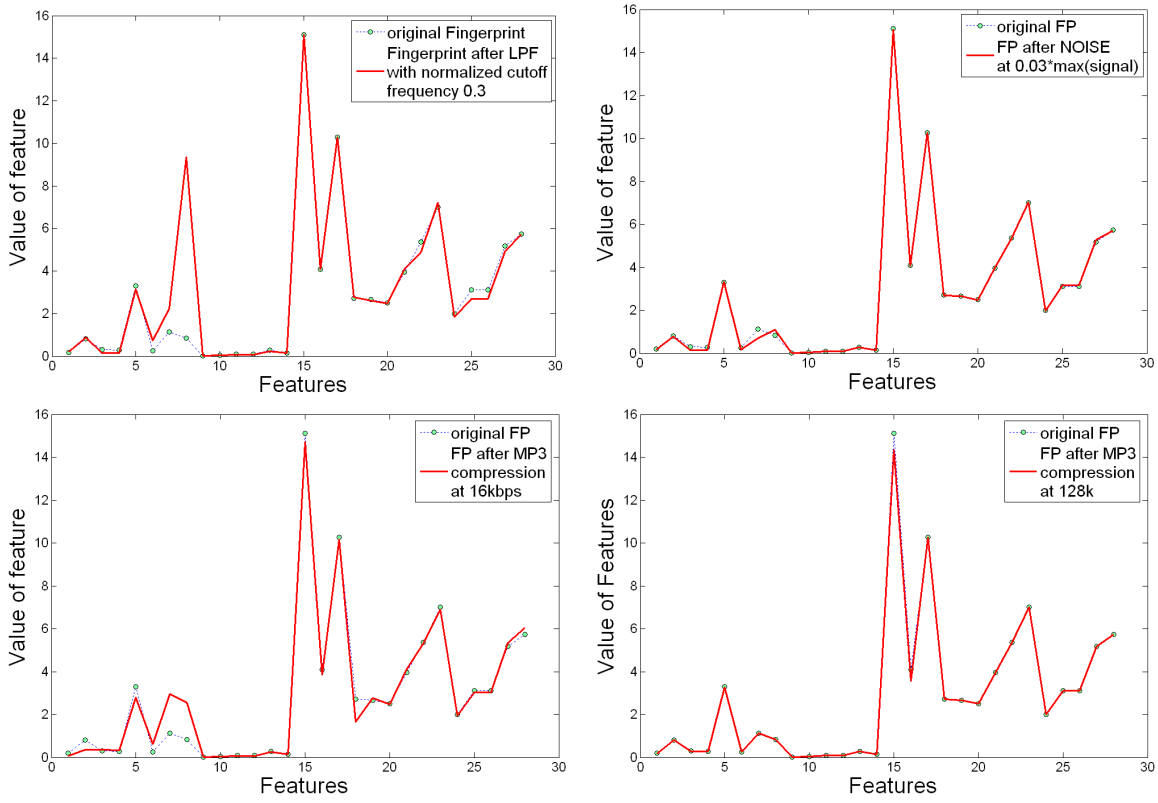


Figure 5.8: Fingerprint (features) comparison before and after attacks. a) low pass filter attack. b) random noise attack with amplitude range at 0.03 times the maximum value of the signal. c) MP3 attack at 16k bit rate. d) MP3 attack at 128k bit rate.

Table 5.1: LDA-CC(cross correlation)-based fingerprint identification accuracy on GTZAN after attacks - low pass filter, additive noise, and MP3 compression.

Attacks	Classical music	Pop music	Rock music	Average accuracy
Low pass filter [23 0.3]	75%	95%	90%	86.7%
Additive noise (SNR=15dB)	90%	95%	90%	91.6%
MP3 compression at 16kbps	80%	85%	70%	78.3%
MP3 compression at 128kbps	95%	95%	95%	95%

the proposed scheme because of the following two reasons: first, it uses the same database GTZAN; second, it also implements a sparse representation technique for inter-class classification. Even though our approach did not test all genre classes, the high accuracy of the music identification across genres shows its robustness. The leave-one-out cross-validation result described in the next paragraph also prove this conclusion.

The results shown in Table 5.2 are obtained without utilizing the LDA classifier, which means the identification is processed using correlation coefficients upon the features only. It is the same as the global correlation shown in Figure 5.4. The drawback is longer computational time but compensated with higher accuracy. The overall accuracy is 92.5%.

To make the result more generalized and comparable to other genre classification schemes, LDA classifier with the leave-one-out cross-validation method is applied on the distorted features. The classification accuracy becomes 78.3%. Note that the result is under distortion and still higher than the accuracy 75% without distortion in [55] at the same dimension of the features.

Another classical audio feature fingerprinting algorithm is worth to mention as well. The fingerprinting system developed by Philips Research [31], which is only referred to as "A Highly Robust Audio Fingerprinting System", is mentioned in many papers and often used as a comparison for newly developed systems [76][77]. In this algorithm, the power spectrum

Table 5.2: CC(cross correlation)-based fingerprint identification accuracy on GTZAN after attacks - low pass filter, additive noise, and MP3 compression.

Attacks	Classical music	Pop music	Rock music	Average accuracy	Leave-one-out accuracy
Low pass filter [23 0.3]	80%	100%	95%	91.6%	81%
Additive noise (SNR=15dB)	95%	100%	95%	96.7%	80.1%
MP3 compression at 16kbps	85%	90%	70%	81.6%	70%
MP3 compression at 128kbps	100%	100%	100%	100%	82%

density of Fourier transform of each audio frame is calculated and then the fingerprint bits for each frame is derivatively obtained between *Bark scale* (i.e., the approximately logarithmic bands that Human Auditory System (HAS) operates on). Given an audio that is 3s long, there will be 256 frames according to the segmentation rate 11.8ms/frame. As the definition, 32-bit sub-fingerprint derived for each frame, the magnitude of the fingerprint for this audio is 8192 bits. The false positive error is 3.6×10^{-20} given the threshold of bit error rate (BER) at 0.35. This is due to the redundancy of the fingerprint. For the same length of audio, if the size of the fingerprint is reduced to the number identical to the proposed fingerprint size, that is 134-bit, much shorter than the previous example, the false positive error turns to 12.3%. In fact, the identification error includes not only the false positive rate but also the false negative rate. The authors claim that: the smaller BER threshold, the smaller the false positive probability will be; on the other hand, the false negative probability will be negatively affected. However, the authors do not give a clear false negative probability with the same BER threshold 0.35. Based on the authors' claim, the probability will be higher than 12.3%. Thus, the identification error is higher than the proposed scheme.

5.5.4 Conclusion

As can be concluded from the testing result above, with small number of iterations or less, the extracted fingerprints with 28 dimensions of audio signals were still quite well matched among three music genres - classical, pop and rock. It makes the scheme very promising for music identification.

The comparison of two identification methods, with LDA and without LDA, tells that if the genre classification is involved, the identification results are less accuracy than the one not involves the genre classification. It implies that exploring strong features that can distinguish genres is needed to improve. This is very important in that it can extremely reduce the computational time.

The proposed system can be summarized as follows:

- **Compactness**: this is the first trial that the features are analyzed and extracted from the sparse representation, by means of MMP, of a signal.
- **Robustness**: the main kind of signal attacks, e.g. low pass filtering, random noises, MP3 compression are evaluated. The average identification rate is up to 92.5% which is higher than the classical algorithm proposed in [31].
- **Search speed and scalability**: by using a two-phase: classification and identification, the requested fingerprint matching can be achieved very faster than brute force searching.

The proposed algorithm can be used for the first step of the content registration process. When a content is submitted for the registration, its fingerprint is extracted and will be compared with the fingerprints in the database. The proposed algorithm can identify the candidate in the database that has the similar fingerprint. After that, experts may be required to judge if the submitted content was derived from the registered contents or it is an original. This part is outside of the scope of the scheme.

Chapter 6

Technical Analysis of the Studies

The previous Chapters demonstrated the schemes designed and studied for content copyright protection, and their performances. However, the techniques applied are worth of analyzing and summarizing such that the schemes can be viewed from different perspectives.

The three main techniques, PCA, DWT and DWT-MDCT-based Sparse Representation techniques MMP, are tightly connected by their similarities and also have their own features so that they are suitable for different applications. The discussion will be unfolded by comparing the similarities and differences of their characteristics.

6.1 The Commons of Techniques

Let's first discuss about their similarities summarized below:

1. They are all categorized as transform techniques.

2. They are all linear decomposition methods.

3. All can be used for signal compression.

It is known that PCA in transform field is called Karhunen-Loève (K-L) Transform. The main characteristics of PCA (K-L Transform) are: (a) the original signal X is the linear combination of bases (eigenvectors); (b) after transformed by a eigenvector matrix, the correlation of the original signal X can be completely removed; (c) during the compression,

the energy of the most principal truncated bases (eigenvectors) corresponds to the summation of the peer eigenvalues.

The linearity and compressibility of DWT and DWT-MDCT-based Sparse Representation techniques MMP have been elaborated in Sections 2.3.4 and 4.2.2. Their differences will be described in Sections 6.2.2 and 6.2.3.

6.2 The Characteristics of Techniques and Their Suitable Applications

6.2.1 Pre-defined bases vs. data driven approach

PCA (K-L Transform) is a data driven decomposition approach whose bases are either u_i or v_i which are derived from its own data; while DWT/MDCT use pre-defined external bases to decompose signal. Therefore, PCA's bases reflect the feature of data itself more than DWT and DCT without priori knowledge. Also, the orientations of the principal eigenvectors are unique to each other. Moreover, the orientations of the top principal eigenvectors are robust under distortion. The degree of their robustness relies on the magnitude of their corresponding eigenvalues. Notice that the orientation stands for the sign change not the value change, because the value of each element is very sensitive to original matrix change as demonstrated in Case 3 of Section 3.1.1. Thus, the orientation of the principal eigenvector leads to the fingerprint generation for the first contribution of this thesis (content-based watermark fingerprinting for P2P network), and used as part of the feature in the second contribution (MMP-based audio feature fingerprinting).

The weakness of the data driven bases is that these bases (especially each element of the eigenvector) are difficult to be interpreted intuitively by an existing technique. Nevertheless, DWT and DWT-MDCT-based sparse representation technique MMP apply pre-defined bases, such as wavelet function and modified discrete cosine bases. Due to the bases are fixed bases, the locations and amplitudes of the corresponding transform coefficients are interpretable parameters, the different manipulations of these parameters generate different features. The robustness of these features rely on the robustness of the bases involved.

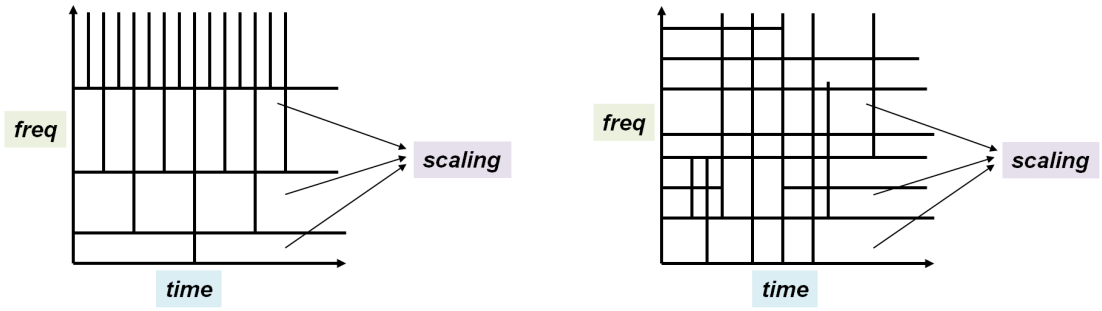
Because the bases chosen for the features are the dominant bases, the robustness can be sustained.

6.2.2 Signal Decomposition of DWT and MP (or MMP)

Besides STFT and Wavelet Transform, both are categorized as linear time-frequency analysis method, another one is categorized as none linear or bi-linear time-frequency analysis method, such as **Wigner-Ville Distribution (WVD)**. The form of the transform is defined as follow:

$$Wigner_x(t,f) = \int x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-j2\pi\tau f}d\tau \tag{6.1}$$

This transform uses signal itself to replace the window function or the wavelet function, so it prevent the conflict between time and frequency. Thus, if the signal is the Linear Frequency Modulation (LFM) signal, the WVD transform can precisely define the instant frequency. However, the drawback of the transform method is that the time-frequency map (or TF tiling) will have a by-product: the cross-term, if the signal consists of compound signals. It is inevitable that the cross-term will be mixed with the true frequencies. There is a better solution different from DWT and WVD, that is MP, defined in Chapter 5, which alter the TF tiling to any resolution to adaptively match the signal. In MP, the scaling parameter is independent of frequency as denoted in Figure 6.1



(a) DWT Time-Frequency Map (b) MP Time-Frequency Map

Figure 6.1: Time-Frequency Map Comparison Between Discrete Wavelet Transform and Matching Pursuit.

Notice that the wavelet function of DWT was also utilized by MMP to match those

percussive signals locally because the wavelet with smaller scale is more close to imitate the impulse signal. It prevents a long expression of sines and cosines and maintains the original intention of sparse representation technique. Since MMP sparsely represents the dominant components of the original signal from either DWT domain or MDCT domain, the features can be robustly and diversely extracted from these domains.

In next section, we will emphasize the other advantage of DWT technique because of its multi-resolution functionality which makes the watermark fingerprinting embedding domain supreme, smaller and easier to manipulate possible.

6.2.3 Compression

From compression perspective, DWT technique has another unique advantages among the three techniques we used in the two contributions.

The advantage stems from its ability to abstract the low resolution of the original data at different scale. As illustrated in Section 2.3.4, the original data have different expression based on the chose of the scale as shown below:

$$\begin{aligned} x(t) &= \mathbb{A}_1(t) + \mathbb{D}_1(t) \\ &= \mathbb{A}_2(t) + \mathbb{D}_2(t) + \mathbb{D}_1(t) \\ &= \mathbb{A}_3(t) + \mathbb{D}_3(t) + \mathbb{D}_2(t) + \mathbb{D}_1(t) \end{aligned} \tag{6.2}$$

where $\mathbb{D}_i(t)$ is called the detail at level i, and $\mathbb{A}_i(t)$ is called the approximation at level i. As i goes larger, the approximation $\mathbb{A}_i(t)$ of the original signal $x(t)$ becomes smoother (lower resolution of original signal) and its size is smaller.

In the watermark fingerprinting for P2P file sharing scheme, the DWT technique was implemented in the first step to obtain the approximation of data at level 5. The equation above implies that the approximation A_i is the supreme item which is the base for the other detail terms. Without this term, the details can not reconstruct a perceptible data.

On the other hand, the approximation at ith level compressed the data to $1/2^i$ from every dimension. The value of the scale parameter i is selected based on the application scenario.

In this study, the scale parameter was chosen so that the fingerprint embedding domain is small enough but also provides a certain space for the flexibility of fingerprint generation and embedding.

PCA (K-L Transform) and MDCT (or MDCT-based MMP) techniques can perform the compression, and these two techniques implement the compression by getting rid of the negligible coefficient terms. However, they do not have the ability to obtain the scaled data during the compression.

6.2.4 Computational Complexity

Since DWT coefficients can be found by applying the two-scale-equation-based filter bank, which is a computationally efficient algorithm, the computational complexity of DWT is $O(n)$. As for the MDCT-based MMP technique, the computational complexity derives from the MDCT. Since MDCT can make use of the Fast Fourier Transform (FFT) algorithm to transform the signal, the complexity of MDCT is $O(nlog_2 n)$.

The worst computational complexity among three techniques belongs to PCA (K-L Transform) technique. Because there is no fast algorithm currently available calculating the eigenvalues and eigenvectors, the complexity of PCA (K-L Transform) is very large. The main bottle neck of the calculation is to resolve the following equation:

$$|XX^T - \lambda I| = 0. \tag{6.3}$$

Hence, in the studies, the PCA (K-L Transform) is implemented only when the data is small enough for the calculation.

6.3 Summary

This small section summaries that why DWT-MDCT-based MMP techniques is suitable for feature fingerprinting while DWT & PCA techniques is for watermark fingerprinting.

In order to make the audio fingerprint sparse and robust, the sparse representation technique MMP was implemented to capture the dominant meaningful parameters of the signal.

On top of the dominant parameters, the unique features were further investigated and extracted.

The fingerprint embedding study, on the other hand, aims to use embedded fingerprint to enable the tracing of the copyright protected file in the Internet. The requirements, i.e., the embedding domain should be supreme and small, the fingerprint is robust, unique, imperceptible, and the fingerprinting procedure is simple, unified and repeatable, should be achieved in good balance. Therefore, due to its simplicity, DWT was employed to provide the supreme and small embedding domain; PCA was then utilized for fingerprint generation because of the robustness and uniqueness of the orientation of each eigenvector, and the imperceptibility by controlling the fingerprint perturbation on both eigenvalues and eigenvectors.

Chapter 7

Conclusions and Future Works

This thesis designed fingerprinting techniques to enhance the copyright protection from two perspectives. The first contribution is to propose a watermark fingerprinting scheme, by using DWT and PCA techniques, to trace the traitors in P2P file-sharing networks. The second contribution is to propose using MMP sparse decomposition to generate the feature fingerprint of an audio file. The feature fingerprint is used to identify if an audio file has been registered in the database.

Either the watermark fingerprinting or the feature fingerprinting technique needs to resolve multiple major problems, that is, the fingerprints, to be embedded or extracted, should be compact, while maintaining the robustness, discrimination, and ease of computation. Moreover, the imperceptibility is another criterion in watermark fingerprinting. The studies show that the sparse signal decomposition techniques provide the good results in solving the problems, such as PCA, DWT, MDCT, MP and MMP. According to the different requirements of the applications, the signals were correspondingly decomposed by the techniques in order to prepare for the next processes.

Unlike other watermark fingerprinting techniques which suffer from poor scalability mentioned in [78], this scheme is scalable through utilizing the DWT technique, not only because it reduces the burden of the media owner's server by only sending the small-size base file and making use of the P2P network infrastructure to support the majority of the file transfer process [78], but also because highly unique fingerprints can be generated. The PCA

technique, on the other hand, determines the orthogonal eigenvectors, which makes it possible to maximally distinguish the different fingerprints. Even though the magnitude of the fingerprint is not significant, it can be identified under distortion because the fingerprint matching does not rely on pixel by pixel match but the sign of the majority of elements in each column. This scheme has shown that the unique fingerprint has strong robustness against most common attacks such as Gaussian white noise, lossy compression, median filter, and geometric distortions. In addition, the PCA technique is calculated on a small size matrix, which causes low computation complexity. The sharable fingerprint, on the other hand, enhances the invulnerability to the collusion attack of the scheme to some extent.

The proposed P2P watermark fingerprinting scheme is the first design in the literature specific for P2P networks and will benefit those multimedia producers who want to share their big file, such as video file, utilizing the convenience of P2P networks. The design concept of the fingerprinting scheme, however, can be applied for regular media file sharing networks, for example, the youtube. The videos that involve content-complicated frames (except for medical applications where original contents are needed without any distortion) can apply this fingerprinting scheme.

There is still some development room for us to further improve the scheme in the future. In order to strongly withstand the common attacks, we need to adjust the scheme to enhance the robustness of the fingerprints without reducing their discrimination abilities and the perceptual quality of the fingerprinted file.

The study, MMP-based audio feature fingerprinting, utilized redundant dictionary based sparse decomposition techniques, to derive the most representable compact components to the signal. Thus, good features are reasonable to be found and are robust enough to perform the classification and identification. The features, chosen from as many aspects as possible, are evaluated for the popular attacks, such as additive noise, low pass filter, MP3 compression. The final results proofed the robustness of this algorithm against the popular attacks, particularly the addition of noise which is one of the most destructive attacks.

The stage's success encourages us to explore the potentials of the scheme in the future.

In the MMP algorithm, we used the atoms derived from the MDCT transform only as the atoms contain more information about the audio signal. Nevertheless, it should be interesting to optimize the MMP algorithm by adding wavelet elements in the dictionary without significantly affecting its computational time as it can provide more interesting features to have a better characterization of the original signal, because the signal which has a lot of transient parts needs wavelet elements to sparsely represent their main features.

The next focus will be the music identification among more music genres. To solve the problem, the classification scheme for inter-class separation plays a very important role, and the main part is still the strong discriminant features themselves, because they will be further used to distinguish each piece of music from the others. Therefore, more robust and discriminant features need to be found from the MMP atoms.

The application of the scheme is not limited to copyright protection. In fact, the idea can be borrowed to other applications, such as file browsing from internet or local computer, and even the applications in human-machine interaction.

Appendix A

Find Minimum Error by Using Lagrange Multipliers

$$\begin{aligned}
\varepsilon &= E[X - \hat{X}]^2 \\
&= E\left\{ \left[\sum_{i=m+1}^{N-1} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [y_i^{**}] \right]^2 \right\} \\
&= E\left\{ \left\langle \sum_{i=m+1}^{N-1} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} [y_i^{**}], \sum_{j=m+1}^{N-1} \begin{bmatrix} | \\ u_j \\ | \end{bmatrix} [y_j^{**}] \right\rangle \right\}
\end{aligned}$$

Assume that $\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}, \begin{bmatrix} | \\ u_j \\ | \end{bmatrix}$ $(i, j = 0, 1, \ldots, N-1)$ is orthogonal, i.e. $\left\langle \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}, \begin{bmatrix} | \\ u_j \\ | \end{bmatrix} \right\rangle = \delta_{ij}$, then

$$\begin{aligned}
\varepsilon &= E\left\{ \sum_{i=m+1}^{N-1} [y_i^{**}]^2 \right\} \\
&= E\left\{ \sum_{i=m+1}^{N-1} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}^T XX^T \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \right\} \\
&= \sum_{i=m+1}^{N-1} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}^T E\left\{XX^T\right\} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \qquad \text{(A.1)}
\end{aligned}$$

Since X is the mean-subtracted signal data, $E\left\{XX^T\right\} = Cov_X$ where Cov_X is the covariance

matrix of X, we get

$$\varepsilon = \sum_{i=m+1}^{N-1} \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}^T Cov_X \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}. \tag{A.2}$$

$$\tag{A.3}$$

Under the orthonomal restriction of $\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}$ and $\begin{bmatrix} | \\ u_i \\ | \end{bmatrix}$, using Lagrange multiplier leads to:

$$\frac{\partial}{\partial \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}} [\varepsilon - \lambda_i \left\langle \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}, \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} \right\rangle] = 0. \tag{A.4}$$

Therefore, the following is derived

$$(Cov_X - \lambda_i I) \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} = 0, \quad i = 0, 1, \ldots, N-1 \tag{A.5}$$

where λ_i is the Lagrange function, and I is $N \times N$ unitary matrix. Thus

$$Cov_X \begin{bmatrix} | \\ u_i \\ | \end{bmatrix} = \lambda_i \begin{bmatrix} | \\ u_i \\ | \end{bmatrix}, \quad i = 0, 1, \ldots, N-1. \tag{A.6}$$

Therefore,

$$\varepsilon_{min} = \sum_{i=m+1}^{N-1} \lambda_i \tag{A.7}$$

where λ_i becomes the eigenvalue of the covariance matrix Cov_X.

Appendix B

Proof for Equation in Wavelet Tutorial

Proof 1

$$< \phi_{j,n}(t), \phi_{j-1,\varkappa}(t) > \tag{B.1}$$

$$= \int_{-\infty}^{\infty} \sqrt{2^j}\phi(2^j t - n)\sqrt{2^{j-1}}\phi(2^{j-1}t - \varkappa)dt \tag{B.2}$$

$$= \int_{-\infty}^{\infty} \sqrt{2^{2j-1}}\phi(2^j t - n)\phi(2^{2j-1}t - \varkappa)dt \qquad \text{(Let } s = 2^{2j-1}t - \varkappa \text{ and substitute)} \tag{B.3}$$

$$= \int_{-\infty}^{\infty} \sqrt{2}\phi(2s + 2\varkappa - n)\phi(s)ds \qquad \text{(Use the two scale equation to replace } \phi(s)\text{)} \tag{B.4}$$

$$= \int_{-\infty}^{\infty} \sqrt{2}\phi(2s + 2\varkappa - n)\sum_m h_0(m)\sqrt{2}\phi(2s - m)ds \tag{B.5}$$

$$= \sum_m h_0(m)\int_{-\infty}^{\infty} \phi(2s + 2\varkappa - n)\phi(2s - m)2ds \qquad \text{(integral is 0 unless } m = n - 2\varkappa\text{)} \tag{B.6}$$

$$= h_0(n - 2\varkappa) \tag{B.7}$$

Proof 2

$$< \phi_{j,n}(t), w_{j-1,\varkappa}(t) > \tag{B.8}$$

$$= \int_{-\infty}^{\infty} \sqrt{2^j}\phi(2^j t - n)\sqrt{2^{j-1}}w(2^{j-1}t - \varkappa)dt \tag{B.9}$$

$$= \int_{-\infty}^{\infty} \sqrt{2^{2j-1}}w(2^j t - n)w(2^{2j-1}t - \varkappa)dt \qquad (\text{Let } s = 2^{2j-1}t - \varkappa \text{ and substitute}) \tag{B.10}$$

$$= \int_{-\infty}^{\infty} \sqrt{2}\phi(2s + 2\varkappa - n)w(s)ds \qquad (\text{Use the two scale equation to replace } w(s)) \tag{B.11}$$

$$= \int_{-\infty}^{\infty} \sqrt{2}\phi(2s + 2\varkappa - n)\sum_m h_1(m)\sqrt{2}\phi(2s - m)ds \tag{B.12}$$

$$= \sum_m h_1(m)\int_{-\infty}^{\infty} \phi(2s + 2\varkappa - n)\phi(2s - m)2ds \qquad (\text{integral is 0 unless } m = n - 2\varkappa) \tag{B.13}$$

$$= h_1(n - 2\varkappa) \tag{B.14}$$

Proof 3 Integrate both sides of the two-scale equation, we get:

$$\phi(t) = \sum_\varkappa h_0(\varkappa)\phi(2t - \varkappa) \tag{B.15}$$

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \phi(2t - \varkappa)dt \tag{B.16}$$

Since

$$\sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \phi(2t - \varkappa)dt \quad if \quad substitute \quad s = 2t \tag{B.17}$$

$$= \sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \frac{1}{2}\phi(s - \varkappa)ds \tag{B.18}$$

$$= \sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \frac{1}{2}\phi(t - \varkappa)dt \tag{B.19}$$

The Eq. B.16 can be rewritten as,

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \phi(2t - \varkappa)dt$$

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_\varkappa h_0(\varkappa)\int_{-\infty}^{\infty} \frac{1}{2}\phi(t - \varkappa)dt \tag{B.20}$$

Because of

$$\int_{-\infty}^{\infty} \phi(t)dt = \int_{-\infty}^{\infty} \phi(t-\varkappa)dt \tag{B.21}$$

the Eq. B.20 leads to the followings:

$$\begin{aligned}
\int_{-\infty}^{\infty} \phi(t)dt &= \sum_{\varkappa} h_0(\varkappa) \int_{-\infty}^{\infty} \frac{1}{2}\phi(t-\varkappa)dt \\
\int_{-\infty}^{\infty} \phi(t)dt &= \sum_{\varkappa} h_0(\varkappa) \int_{-\infty}^{\infty} \frac{1}{2}\phi(t)dt \qquad \text{(B.22)} \\
1 &= \sum_{\varkappa} h_0 \sqrt{2}\frac{1}{2} \\
\sum_{\varkappa} h_0 &= \sqrt{2}
\end{aligned}$$

Appendix C

Publications

- Xiaoli Li, Alexandra Randriamanohisoa, Ngok-Wah Ma, Sridhar Krishnan, "Audio Fingerprinting based on the Molecular Matching Pursuit Algorithm," to be submitted to IEEE Transactions on Information Forensics and Security.

- Sridhar Krishnan, Xiaoli Li, Yaqing Niu, Ngok-Wah Ma, and Qin Zhang, "Watermarking and Fingerprinting Techniques for Multimedia Protection," book chapter to appear in Multimedia Image and Video Processing (2nd edition), CRC PRESS.

- Xiaoli Li, Sridhar Krishnan, Ngok-Wah Ma, "A Wavelet-PCA-Based Fingerprinting Scheme for Peer-to-Peer Video File Sharing," *IEEE Transactions on Information Forensics and Security*, Volume 5, No. 3, pp. 365-373, 2010.

- Ying Shen, Xiaoli Li, Ngok-Wah Ma, and Sridhar Krishnan, "Parametric Time-Frequency Analysis and Its Applications in Music Classification," *Special Issue on Time-frequency Methods and Applications in Multimedia in EURASIP Journal on Advances in Signal Processing*, vol. 2010, 2010.

- Xiaoli Li, Sridhar Krishnan, Ngok-Wah Ma, "Application of grammar-based codes for lossless compression of digital mammograms," Journal of Electronic Imaging, Volume 15, 013021, 2006.

- Xiaoli Li, Sridhar Krishnan, Ngok-Wah Ma, "A novel way of lossless comparison of

digital mammograms using grammar codes," Canadian Conference on Electrical and Computer Engineering, Volume 4, pp. 2085 - 2088, 2004.

Bibliography

[1] S.G. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Transaction on Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[2] J. Tomàs-Buliart, M. Fernández, and M. Soriano, "Traitor Tracing over YouTube Video Service - Proof of Concept," *Telecommunication Systems 45 (1)*, pp. 47-60, 2010.

[3] Leandro de C.T. Gomes, P. Cano, E. Gómez, M. Bonnet, and E. Batlle, "Audio Watermarking and Fingerprinting: For Which Applications?," *Journal of New Music Research*, vol. 32, pp. 65-81, Mar. 2003.

[4] S. Craver, M. Wu, and B. Liu, "What Can We Reasonably Expect from Watermarks?" in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 223-226, New Paltz, NY, Oct. 2001.

[5] W. Luh and D. Kundur, "New Paradigms for Effective Multicasting and Fingerprinting of Entertainment Media," *IEEE Communication Magazine*, vol. 43, issue 6, pp. 77-84, June 2005.

[6] M. Wu, W. Trappe, Z.J. Wang, and K.J.R. Liu, "Collusion-Resistant Fingerprinting for Multimedia," *Signal Processing Magazine, IEEE*, vol. 21, issue 2, pp. 13-27, Mar 2004.

[7] F. Ergun, J. Kilian, and R. Kumar, "A Note on the Limits of Collusion-Resistant Watermarks," in *Proc. Eurocrypt'99*, pp. 140-149, 1999.

[8] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-Collusion Fingerprinting for Multimedia," *IEEE Trans. Signal Processing*, vol. 51, pp. 1069-1087, Apr. 2003.

[9] D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897-1905, Sept. 1998.

[10] A. Barg, G.R. Blakley, and G.A. Kabatiansky, "Digital Fingerprinting Codes: Problem Statements, Constructions, Identification of Traitors," *IEEE Transactions on Information Theory*, vol. 49, issue 4, pp. 852-865, 2003.

[11] H. Chu, L. Qiao, and K. Nahrstedt, "A Secure Multicast Protocol with Copyright Protection," *ACM (Association Computing Machinery) Special Interest Group on Data Communication (SIGCOMM) Computer Communications Review*, vol. 32, pp. 42-60, 2002.

[12] M. Fernandez and M. Soriano, "Fingerprinting Concatenated Codes with Efficient Identification," in *proceedings of the 5th international conference on information security (ISC'02)*, London, UK, pp. 459-470, Berlin: Springer, 2002.

[13] G. Tardos, "Optimal Probabilistic Fingerprint Codes," *Journal of the Association Computing Machinery (ACM)*, vol. 55, issue 2, no. 10, pages 24, 2008.

[14] S. He and M. Wu, "Joint Coding and Embedding Techniques for Multimedia Fingerprinting," *IEEE Transaction on Information Forensics and Security*, vol. 1, issue 2, pp. 231-247, 2006.

[15] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, *Peer-to-Peer Computing*, Hewlett-Packard's Company. [Online]. Available: [http://www.hpl.hp.com/techreports/2002/HPL-2002-57R1.pdf]. The Latest Accessed Date: Sept. 2011.

[16] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Communications Survey and Tutorial*, Mar. 2004. [Online]. Available: [http://www.cl.cam.ac.uk/teaching/2005/AdvSysTop/survey.pdf]. The Latest Accessed Date: Sept. 2011.

[17] D. Tsolis, S. Sioutas, and T. Papatheodorou, "Digital Watermarking in Peer to Peer Networks," *16th Internatial Conference on Digital Signal Processing*, pp. 1-5, Santorini-Hellas, Jul. 2009.

[18] A. Kaarna and P. Toivanen, "Digital Watermarking of Spectral Images in PCA/Wavelet-Transform Domain," *Geoscience and Remote Sensing Symposium. IGARSS '03. Proceedings. 2003 IEEE International*, vol.6, pp. 3564-3567, Jul. 2003.

[19] R. Liu and T. Tan, "An SVD-Based Watermarking Scheme for Protecting Rightful Ownership ," *Multimedia, IEEE Transactions on*, vol. 4, no.1, pp. 121-128, Mar 2002.

[20] T. D. Hien, Z. Nakao, K. Miyara, Y. Nagata, and Y. W. Chen, "A New Chromatic Color Image Watermarking and Its PCA-Based Implementation," *8th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2006*, subseries: Lecture Notes in Artificial Intelligence (LNAI) 4029, pp. 787-795, 2006.

[21] P. Indyk, G. Iyengar, and N. Shivakumar, "Finding Pirated Video Seuences on the Internet," Tech. Report, Stanford InfoLab, Stanford University, Feb. 1999.

[22] N. Shivakumar, "Detecting Digital Copyright Violations on the Internet," Ph.D. Dissertation, Stanford University, Aug. 1999.

[23] P. Indyk, "High-Dimensional Computational Geometry," Ph.D. Disseration, Stanford University, Sept. 2000.

[24] P. Cano, E. Batlle, T. Kalker and J. Haitsma, "A Review of Algorithm for Audio Fingerprinting," *IEEE Workshop on Multimedia Signal Processing*, pp. 169-173, Dec. 2002.

[25] P. Cano, E. Batlle, T. Kalker and J. Haitsma, "A Review of Audio Fingerprinting," *Journal of VLSI (Very Large Scale Integration) Signal Processing Systems*, vol. 41, issue 3, pp. 271-284, Nov. 2005.

[26] J. Lu, "Video Fingerprinting for Copy Identification: from Research to Industry Applications," in *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE) - Media Forensics and Security XI*, vol. 7254, pp. 725402-725402-15, Jan. 2009.

[27] J. Lourens, "Detection and Logging Advertisements Using its Sound," in *Communications and Signal Processing, 1990. COMSIG 90. Proceedings., IEEE 1990 South African Symposium on*, pp. 209-212, 1990.

[28] F. Kurth, A. Ribbrock, and M. Clausen, "Identification of Highly Distorted Audio Material for Querying Large Scale Databases," in *Proc. Audio Engineering Society 112th Int. Conv.*, Munich, Germany, May 2002.

[29] S. Subramanya, R. Simha, B. Narahari, and A. Youssef, "Transform-Based Indexing of Audio Data for Multimedia Database," in *Proc. of Int. Conf. on Multimedia Computing and Systems*, pp. 3-6, Ottawa, Canada, June 1997.

[30] J.S. Seo, M. Jin, S. Lee, D. Jing, S. Lee, and C.D. Yoo, "Audio Fingerprinting Based on Normalized Spectral Subband Moments," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 213-216, Nov. 2005.

[31] J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprinting System," *International Symposium on Music Information Retrieval (ISMIR 2002)*, pp. 107-115, Oct. 2002.

[32] A. Ramalingam and S. Krishnan, "Gaussian Mixture Modeling of Short-time Fourier Transform Features for Audio Fingerprinting," *IEEE Transaction on Information Forensics and Security*, vol. 1, issue 4, pp. 457-463, Dec. 2006.

[33] K. Sungwoong and C.D. Yoo, "Boosted Binary Audio Fingerprint Based on Spectral Subband Moments," *IEEE International Conference on Acoustics, Sppech, and Signal Processing*, pp. 241-244, July 2007.

[34] C.S. Lu, "Audio Fingerprinting Based on Analyzing Time-Frequency Localization of Signals," *IEEE Workshop on Multimedia Signal Processing*, pp. 174-177, Dec. 2002.

[35] M. Mihak and R. Venkatesan, "A Perceptual Audio Hashing Algorithm: A Tool for Robust Audio Identification and Information Hiding," *presented at the 4th Workshop on Information Hiding*, pp. 51-65, 2001.

[36] C. Burges, J. Platt, and S. Jana, "Distortion Discriminant Analysis for Audio Fingerprinting," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 165-174, May 2003.

[37] Y. Shen, X.L. Li, N.W. Ma, and S. Krishnan, "Parametric Time-Frequency Analysis and Its Applications in Music Classification," *Special Issue on Time-frequency Methods and Applications in Multimedia in EURASIP (European Association for Signal Processing Journal) on Advances in Signal Processing*, vol. 2010, pages 9, 2010.

[38] W. Heisenberg, "Physikalische Prinzipien der Quantentheorie, Leipzig: Hirzel English Translation The Physical Principles of Quantum Theory," Chicago: University of Chicago Press, 1930.

[39] S. Chu, S. Narayannan, and C.-C.J. Kuo, "Environmental Sound Recongition Using MP-based Features," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1-4, 2008.

[40] E. Allamanche, J. Herre, O. Helmuth, B. Fröba, T. Kasten, and M. Cremer, "Content-Based Identification fo Audio Material Using Mpeg-7 Low Level Description," in *Proc. of the Int. Symp. of Music Information Retrieval*, Indiana, USA, Oct. 2001.

[41] C. Papaodysseus, G. Roussopoulos, D. Fragoulis, T. Panagoppoulos, and C. Alexiou, "A New Approach to the Automatic Recognition of Musical Recordings," *J. Audio Eng. Soc.*, vol 49, no. 1/2, pp. 23-35, 2001.

[42] A. Kimura, K. Kashino, T.Kurozumi, and H. Murase, "Very Quick Audio Searching: Introducing Global Pruning to the Time-Series Active Search," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, vol.3, pp.1429-1432, Salt Lake City, Utah, USA, May 2001.

[43] S. Sukittanon and L. Atlas, "Modulation Frequency Features for Audio Fingerprinting," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, pp. 1773-1776, May 2002.

[44] T. Blum, D. Keislar, J. Wheaton, and E. Wold, "Method and Article of Manufacture for Content-Based Analysis, Sotrage, Retrieval and Segmentation of Audio Information," U.S. Patent 5,918,223, June 1999.

[45] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust Sound Modeling for Song Detection in Broadcast Audio," in *Proc. Audio Engineering Society 112th Int. Conv.*, pp. 1-7, Munich, Germany, May 2002.

[46] E. Batlle, J. Masip, and E. Guaus, "Automatic Song Identification in Noisy Broadcast Audio," in *Proc. of Signal and Image Processing (SIP)*, Aug. 2002.

[47] Etantrum (2002) [Online]. Available: [http://www.freshmeat.net/projects/songprint]. The Latest Accessed Date:

[48] D. Kirovski and H. Attias, "Beat-id: Identifying Music via Beat Analysis," in *5th IEEE Int. Workshop on Multimedia Signal Processing: Special session on Media Recognition*, pp. 190-193, US Virgin Islands, USA, Dec. 2002.

[49] K. Huang and S. Aviyente, "Sparse Representation for Signal Classification," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 609-616, 2006.

[50] Project-Team Metiss, "Modélisation et Expérimentation pour le Traitement des Informations et des Signaux Sonores," *Institut National De Recherche En Informatique Et En Automatique*, 2009.

[51] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *27th Annual Asilomar Conference on Signals, Systems, and Computers*, pp. 40-44, 1993.

[52] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic Decomposition by Basis Pursuit," *Society for Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing*, vol. 20, issue 1, pp. 33-61, Aug. 1998.

[53] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An Algorithm for Designing of Overcomplete Dictionaries for Sparse Representation," *Signal Processing, IEEE Transaction on*, vol. 54, no. 11, pp. 4311-4322, 2006.

[54] Y. Li, A. Cichocki, and S. Amari, "Analysis of Sparse Representation and Blind Source Separation," *Neural Computation*, vol. 16, no. 6, pp. 1193-1234, 2004.

[55] Y. Panagakis, C. Kotropoulos, and G.R. Arce, "Music Genre Classfication via Sparse Representations of Auditory Temporal Modulations," *17th European Signal Processing Conference (EUSIPCO 2009)*, Glasgow, Scotland, pp. 1-5, August 2009.

[56] J. Starck, M. Elad, and D. Donoho, "Image Decomposition via the Combination of Sparse Representation and A Variational Approach," *IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1570-1582, 2005.

[57] L. Daudet, "Sparse and Structured Decompositions of Signals With the Molecular Matching Pursuit," *IEEE Transaction on Audio, Speech and Language Processing*, vol. 14, issue 5, pp. 1808-1816, Sept. 2006.

[58] M. Parvaix, "An Audio Watermarking Method Based on Molecular Matching Pursuit," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008*, ISBN: 1-4020-7357-7, pp. 1721-1724, Las Vegas, Nevada, US, 2008.

[59] M. Bosi and R. E. Goldberg, *Introduction To Digital Audio Coding And Standards*, Kluwer Academic Publisher Norwell, MA, USA 2002.

[60] [Online]. Available: [http://www.techno-science.net/?onglet=glossaire&definition=8040]. The Latest Accessed Date: Sept. 2011.

[61] [Online]. Available: [http://plato.stanford.edu/entries/qt-uncertainty/]. The Latest Accessed Date: Sept. 2011.

[62] W. J. Phillips, *Wavelets and Filter Banks Course Notes*, [Online]. Available: [http://www.engmath.dal.ca/courses/engm6610/notes/]. The Latest Accessed Date: Sept. 2011.

[63] E. Bacry, "LastWave Documentation," [Online]. Availabe: [http://www.cmap.polytechnique.fr/ bacry/LastWave/download_doc.html], 2008. The Latest Accessed Date: Sept. 2011.

[64] L. Breiman, "Statistical Modeling: the Two Cultures," *Statistical Science*, vol. 16, no. 3, pp. 199-231, 2001.

[65] X. Li, S. Krishnan, N. W. Ma, "A Wavelet-PCA-Based Fingerprinting Scheme for Peer-to-Peer Video File Sharing," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 365-373, 2010.

[66] E. Chang, J. Wang, C. Li and G. Wiederhold, "RIME: A Replicated Image Detector for the World Wide Web," in *SPIE (Society of Photo-Optical Instrumentation Engineers) Multimedia Storage and Archiving Systems III*, pp. 58-67, Nov. 1998.

[67] J. M. Sanchez, X. Binefa, J. Vitria, and P. Radeva, "Local Color Analysis for Scene Break Detection Applied to TV Commercials Recognition," *Int. Conf. Vis. Inf. Syst.*, pp. 237-244, 1999.

[68] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* New York: Addison-Wesley, ISBN: 0201508036 9780201508031, 1992.

[69] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Towards Second Generation Watermarking Schemes," *Proc., of the IEEE Inter., Conf., on Image Processing, ICIP 99*, Kobe, Japan, vol. 1, pp. 320-323, 1999.

[70] M. Kutter, F. Jordan, and F. Bossen, "Digital Signature of Color Image Using Amplitude Modulation," in *Proc. SPIE (Society of Photo-Optical Instrumentation Engineers) Conference on Storage and Retrieval for Image and Video Databases*, vol. 3022, San Jose, CA, pp. 518-526, Feb. 1997.

[71] J. Dittmann, "Combining digital Watermarks and Collusion Secure Fingerprints for Customer Copy Monitoring," *Proc. IEEE Seminar Sec. Image & Image Auth.*, pp. 128-132, March 2000.

[72] K. Umapathy, S. Krishnan, and S. Jimaa, "Audio Signal Classification Using Time-Frequency Parameters," *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 249-252, August, 2002.

[73] SPSS Inc., "SPSS Advanced Statistics User's Guide," *User Manual*, SPSS (Statistical Package for the Social Sciences) Inc., Chicago, IL, 1990.

[74] R. Gribonval and E. Bacry, "Harmonic Decompostions of Audio Signals with Matching Pursuit," *IEEE Tran. Signal Processing.*, vol. 51, no. 1, pp. 101-111, Jan. 2003.

[75] Music analysis, retrieval and synthesis for audio signals. http://marsyas.info/download/data_sets.

[76] P. J. O. Doets and R. L. Lagendijk, "Theoretical Modeling of A Robust Audio Fingerprinting System," in *Proceedings of SPS 2004 (the 2004 IEEE Benelux Signal Processing Symposium)*, pp. 101-104, 2004.

[77] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer Vision for Music Identification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.

[78] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proc. 7th IEEE International Conference on Computer Vision, ICCV 1999*, pp. 1150-1157, 1999.