NEW DIGITAL AUDIO WATERMARKING ALGORITHMS FOR COPYRIGHT PROTECTION

Jian Wang, M.Sc.

Department of Computer Science

National University of Ireland, Maynooth, Co. Kildare, Ireland

NUI MAYNOOTH

Ollscoil na hÉireann, Má Nuad

A Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Supervisor: Dr. Joseph Timoney

September 2011

Declarations

I confirm this is my own work and the use of all material from other sources has been properly cited and fully acknowledged.

Signature: Jian Wang Date: September 2011

Acknowledgements

This research would not have been possible without the help and support from my supervisor, Dr. Joe Timoney. I am extremely grateful for all his time, effort, patience and guidance which he has invested and provided throughout the course of this PhD.

I would like to acknowledge and thank the Department of Computer Science for providing me with many opportunities to present my work. I would also like to extend my gratitude to the staff members in the Department of Computer Science. I do not have room to name everyone who has helped me out over the years, so I will say a special thanks to Adam Winstanley, Philip Maguire, Aidan Mooney, Stephen Brown, Susan Burgan, Patrick Marshall and Michael Monaghan. Special thanks are due to Ron for his help during this journey.

Finally, I would like to thank my family deeply for all their support and encouragement. I will never forget it.

Glossary

AAC	Advance Audio Coding
A/D	Analogue to Digital
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
bps	bits per second
COLA	Constant Overlap Add
CSPE	Complex Spectral Phase Evolution
CSVD	Compact Singular Value Decomposition
D/A	Digital to Analogue
dB	Decibel
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
GPA	Generalized Patchwork Algorithm
HAS	Human Auditory System
LSB	Least Significant Bit
MCLT	Modulated Complex Lapped Transform
MOS	Mean Opinion Score
MPA	Modified Patchwork Algorithm
MPM	Multiplicative Patchwork Method

NMR	Noise to Mask Ratio
ODG	Objective Difference Grade
PEAQ	Perceptual Evaluation of Audio Quality
PN	Pseudo Noise
PSD	Power Spectral Density
QIM	Quantization Index Modulation
RMS	Root Mean Square
SMR	Signal to Mask Ratio
SNR	Signal to Noise Ratio
SNR_{seg}	Segmental Signal to Noise Ratio
SQAM	Sound Quality Assessment Material
SS	Spread Spectrum
SVD	Singular Value Decomposition
TS	Time Spread

Abstract

This thesis investigates the development of digital audio watermarking in addressing issues such as copyright protection. Over the past two decades, many digital watermarking algorithms have been developed, each with its own advantages and disadvantages. The main aim of this thesis was to develop a new watermarking algorithm within an existing Fast Fourier Transform framework. This resulted in the development of a Complex Spectrum Phase Evolution based watermarking algorithm. In this new implementation, the embedding positions were generated dynamically thereby rendering it more difficult for an attacker to remove, and watermark information was embedded by manipulation of the spectral components in the time domain thereby reducing any audible distortion. Further improvements were attained when the embedding criteria was based on bin location comparison instead of magnitude, thereby rendering it more robust against those attacks that interfere with the spectral magnitudes.

However, it was discovered that this new audio watermarking algorithm has some disadvantages such as a relatively low capacity and a non-consistent robustness for different audio files. Therefore, a further aim of this thesis was to improve the algorithm from a different perspective.

Improvements were investigated using an Singular Value Decomposition framework wherein a novel observation was discovered. Furthermore, a psychoacoustic model was incorporated to suppress any audible distortion. This resulted in a watermarking algorithm which achieved a higher capacity and a more consistent robustness.

The overall result was that two new digital audio watermarking algorithms were developed which were complementary in their performance thereby opening more opportunities for further research.

Table of Contents

Declarations ii
Acknowledgements iii
Glossary iv
Abstract vi
List of Figures xiii
List of Tables xvii
List of Publications xviii
Chapter 1 General Introduction 1
1.1 Introduction 1
1.2 Brief history of watermarking 2
1.3 Illustration of watermarking algorithm 4
1.3.1 Watermarking definition 4
1.3.2 Watermarking characteristics 6
1.3.3 Watermarking trade-off 9
1.4 Applications of watermarking 10
1.4.1 Copyright protection 10
1.4.2 Authentication and Tamper proofing 10
1.4.3 Identification and Tracking of digital content 11
1.4.4 Copy prevention 12
1.5 Motivation 12
1.6 Contributions 13
1.7 Thesis Overview 14
Chapter 2 Literature Review 16
2.1 Introduction 16
2.2 Quantitative evaluation of the audio watermarking performance 17
2.2.1 Imperceptibility 17
2.2.2 Accuracy 22
2.2.3 Robustness 22
2.2.4 Capacity 23
2.2.5 Computational efficiency 23

2.3 Review of audio watermarking algorithms 24
2.3.1 Time domain based algorithms 24
2.3.1.1 LSB coding 24
2.3.1.2 Echo hiding 26
2.3.2 Transformation based algorithms 30
2.3.2.1 FFT based audio watermarking algorithms 30
2.3.2.2 DWT based audio watermarking algorithms 35
2.3.2.3 Spread Spectrum watermarking algorithms 40
2.3.3 Hybrid algorithms 44
2.3.3.1 Chirp Coding watermarking algorithms 44
2.3.3.2 Quantization based watermarking algorithms 46
2.3.3.3 Patchwork watermarking algorithms 47
2.3.3.4 Interpolation based watermarking algorithms 50
2.3.3.5 SVD based watermarking algorithms 54
2.4 Evaluation 57
2.5 Summary 62
Chapter 3 Complex Spectral Phase Evolution based audio watermarking algorithm 64
3.1 Introduction 64
3.2 The Complex Spectral Phase Evolution 64
3.3 The procedure of CSPE 65
3.4 The mathematical deduction of CSPE 66
3.5 Performance demonstration of CSPE 68
3.6 The performance comparison between CSPE and Quadratic Interpolation Estimation 69
3.7 Issues with CSPE 71
3.8 Improvement on CSPE 73
3.9 Using the improved CSPE in audio watermarking 79
3.9.1 Embedding 80
3.9.1.1 Dynamic selection of two candidate components 80
3.9.1.2 Embedding rules 81
3.9.1.3 Modification of one of the candidate components 83
3.9.1.4 Embedding process 84
3.9.2 Detection 86

3.9.3 Evaluation 86
3.10 Application to Real audio Signals 88
3.10.1 Issues with the proposed audio watermarking algorithm 89
3.10.2 The reason for the low *Precision* of the watermarking algorithm 91
3.10.3 Using bin location to define embedding rules 93
3.10.4 Modification of the selected components 94
3.10.5 Component verification process 96
3.10.6 Transparency and audible artifacts 96
3.10.6.1 The reason for 'Type I' click 96
3.10.6.2 The reason for 'Type II' click 98
3.10.6.3 The occurrence probability of both clicks 99
3.10.6.4 The solution to the audible artefact issue 100
3.10.7 Watermark detection procedure 102
3.10.8 Results 103
3.11 Summary 104
Chapter 4 robustness improvement of the Complex Spectral Phase Evolution based audio watermarking algorithm 105
4.1 Introduction 105
4.2 Investigation on the robustness of the previous algorithm 105
4.3 Embedding watermark information by manipulating the peaks 109
4.3.1 Using *r* to select the candidate peak 110
4.3.2 Embedding rules 110
4.3.3 Manipulation approach 111
4.4 Using thresholds to select and manipulate the candidate peak 112
4.4.1 Issue with only using Th_a 112
4.4.2 The first scenario that satisfies the assumption 112
4.4.3 The second scenario that satisfies the assumption 113
4.4.4 The solution to avoid the two scenarios 114
4.5 The embedding procedure 115
4.6 The detection procedure 117
4.7 Investigation on the impact of the parameters values on performance 118
4.7.1 Questions to be answered 118

4.7.2 Accuracy of the proposed algorithm 121
4.7.3 Robustness against MP3 attack 122
4.7.4 Imperceptibility 125
4.7.5 Brief summary 127
4.8 Using a peak sharpness measure to improve the robustness 128
4.8.1 The probability of one bin shift phenomenon 128
4.8.2 The reason for this one bin shift phenomenon 129
4.8.3 The solution to this one bin shift phenomenon 130
4.8.4 Experimental validation 131
4.9 Using an error-correction scheme to improve the robustness 132
4.9.1 Using Reed-Solomon coding to improve the robustness 132
4.9.2 Using Repetition to improve the robustness 136
4.10 Improving the capacity of the scheme 138
4.11 Validation on another test data set 139
4.11.1 Accuracy 140
4.11.2 Robustness against a variety of attacks 141
4.11.2.1 Robustness against MP3 attack 141
4.11.2.2 Robustness against AAC attack 143
4.11.2.3 Robustness against Additive White Gaussian Noise attack 145
4.11.2.4 Robustness against a lowpass filtering attack 148
4.11.2.5 Robustness against a highpass filtering attack 150
4.11.2.6 Robustness against Noise Removal attack 152
4.11.3 Imperceptibility 154
4.11.4 Computational efficiency 156
4.12 Comparison with other algorithms 157
4.13 Summary 158
Chapter 5 Singular Value Decomposition based audio watermarking Algorithm 159
5.1 Introduction 159
5.2 SVD 160
5.3 SVD based watermarking algorithm 161
5.3.1 Matrix organization 161
5.3.2 Using the second column of *U* to embed the watermark 161

5.3.3 Complete signal reconstruction using overlapping window frame series 167
5.3.4 The watermark embedding process 171
5.3.5 The watermark detection process 172
5.3.6 Audible distortion detection and suppression 173
5.3.7 Experimental validation 176
5.4 Evaluation 178
5.4.1 Imperceptibility 178
5.4.1.1 SNR 178
5.4.1.2 ODG 179
5.4.2 Robustness 181
5.4.2.1 Robustness against MP3 Compression 181
5.4.2.2 Robustness against other selected attacks 182
5.4.2.3 Using the repetition to increase the robustness 184
5.4.3 Capacity 185
5.5 Comparison with other algorithms 185
5.6 Summary 186
Chapter 6 Conclusion and Future Directions 187
6.1 Introduction 187
6.2 Conclusions 187
6.3 Future work 190
References 194

List of Figures

Figure 1.1 The flowchart of the embedding process 5
Figure 1.2 One type of detection processes 5
Figure 1.3 The other type of detection processes 6
Figure 1.4 Trade-off presents in watermarking system 9
Figure 1.5 Content identification using digital watermarking 12
Figure 2.1 The demonstration of how sample value is altered by different LSB algorithms [CS05] 26
Figure 2.2 (a) "one" kernel (b) "zero" kernel 27
Figure 2.3 The embedding process of the algorithm proposed in [FM09] 32
Figure 2.4 The detection process of the algorithm proposed in [FM09] 33
Figure 2.5 The DWT analysis of 16 samples data 36
Figure 2.6 (a) The embedding procedure (b) The detection procedure 43
Figure 2.7 The watermarking embedding process 47
Figure 2.8 The embedding process of the algorithm proposed in [DP09] 51
Figure 2.9 The detection process of the algorithm proposed in [DP09] 52
Figure 3.1 The procedure of CSPE 66
Figure 3.2 Frequency estimation of a multi-component signal by the CSPE algorithm (marked by arrows) 69
Figure 3.3 Accuracy comparison of frequency estimation between quadratic fit and CSPE 70
Figure 3.4 Frequency components that cannot be identified by CSPE 72
Figure 3.5 FFT spectrum of the signal 73
Figure 3.6 Magnitude response of three different window functions 75
Figure 3.7 Magnitude response of apodization function 77
Figure 3.8 Frequency estimation by improved CSPE 78
Figure 3.9 The demonstration of candidate component selection 81
Figure 3.10 The demonstration of embedding a bit '0' or '1' 82
Figure 3.11 The procedure of a watermark bit embedding 85
Figure 3.12 The *Precision* of each Signal 87
Figure 3.13 The Histogram distribution of *Precision* 88
Figure 3.14 The Precision distribution of 20 music files 90

Figure 3.15 CSPE frequency identification of a signal containing two components whose frequencies are very close 92
Figure 3.16 CSPE frequency identification of a signal containing one component 93
Figure 3.17 The demonstration of the embedding rule as shown in Equation (3.20) 94
Figure 3.18 The demonstration of bin location change by manipulation in order to satisfy Equation (3.20) 95
Figure 3.19 Spectrum of the 82th frame for original signal, signal after first change and final signal, illustrating Type 1 click. 97
Figure 3.20 Spectrum of the 313th frame for original signal, signal after first change and final signal, illustrating Type II click. 99
Figure 3.21 The embedding process of the improved algorithm 101
Figure 3.22 The detection process for each frame 102
Figure 3.23 The distribution of ODG scores for 25 sample watermarked files 104
Figure 4.1 The *Precision* distribution after 64 kbps MP3 compression 106
Figure 4.2 Magnitude spectrum using CSPE and FFT 108
Figure 4.3 The consistency of CSPE spectrum peaks before and after 64 kbps MP3 109
Figure 4.4 The demonstration of embedding rules 110
Figure 4.5 The demonstration of the peak being selected before MP3 compression 113
Figure 4.6 The demonstration of the peak being selected after MP3 compression 113
Figure 4.7 The demonstration of the peak being selected before MP3 compression 114
Figure 4.8 The demonstration of the peak being selected after MP3 compression 114
Figure 4.9 The embedding procedure 117
Figure 4.10 The *Precision* distribution without attack for 20 audio files with the parameters given by Group 1-20 from Table 4.1 121
Figure 4.11 (*a* to *d*): The *Precision* distribution after 64 kbps MP3 attack for 20 audio files with the parameters given by Group 1-40 from Table 4.1 124
Figure 4.12 The ODG score distribution without attack for 20 audio files with the parameters given by Group 1-20 from Table 4.1 127
Figure 4.13 The probability distribution of the bin shift phenomenon for 20 audio files 129
Figure 4.14 The demonstration of bin shift phenomenon 130
Figure 4.15 The *Precision* after using sharpness process 131
Figure 4.16 The *Precision* after using Reed-Solomon 134

Figure 4.17 The distribution of *Precision* after MP3 64 kbps when using different values for the repetition *d* 137
Figure 4.18 The distribution of $Precision_{mean}$ when using different values for the repetition *d* .. 138
Figure 4.19 The *Precision* distribution without any attack 140
Figure 4.20 The *Precision* distribution after 64 kbps MP3 attack without using repetition 142
Figure 4.21 The *Precision* distribution after 64 kbps MP3 attack, using repetition ($d = 5$) 143
Figure 4.22 The *Precision* distribution after AAC attack without using repetition 144
Figure 4.23 The *Precision* distribution after AAC attack using repetition ($d = 5$) 145
Figure 4.24 The ODG score distribution of each resulting signal file after adding a particular level of AWGN 146
Figure 4.25 The *Precision* distribution after a particular level of AWGN attack without using repetition 147
Figure 4.26 The *Precision* distribution after a particular level of AWGN attack using repetition ($d = 5$) 148
Figure 4.27 The *Precision* distribution after lowpass filtering attack without using repetition ... 149
Figure 4.28 The *Precision* distribution after lowpass filtering attack using repetition ($d = 5$) 150
Figure 4.29 The *Precision* distribution after highpass filtering attack without using repetition .. 151
Figure 4.30 The *Precision* distribution after highpass filtering attack using repetition ($d = 5$) ... 152
Figure 4.31 The *Precision* distribution after noise removal attack without using repetition 153
Figure 4.32 The *Precision* distribution after noise removal attack using repetition ($d = 5$) 154
Figure 4.33 (a) SNR distribution for all the 25 files (b) SNRseg distribution for all the 25 files 155
Figure 4.34 The ODG distribution for Group 1 and 2 156
Figure 5.1 The demonstration of the distortion introduced by manipulation different columns of the matrix *U*. 164
Figure 5.2 (a) Robustness of the created peaks (b) Robustness of the created peaks (c) Spectrum of the original signal and the modified signal 166
Figure 5.3 (a) $x_1(n)$ (b) $w_1(n)$: non-overlapping window frame series (c) $w_2(n)$: overlapping window frame series (d) Illustration of COLA principal 169
Figure 5.4 The flowchart of watermark embedding 172
Figure 5.5 The ODG distribution of 10 music files 173
Figure 5.6 The additional process of improving the imperceptibility 176

Figure 5.7 (a) ODG score before and after applying further suppression (b) Precision before and after applying further suppression 177
Figure 5.8 (a) SNR distribution for all the 25 music files (b) SNRseg distribution for all the 25 music files 179
Figure 5.9 ODG scores distribution for all the 25 music files 180
Figure 5.10 ODG scores distribution by manipulating the first column of *U* 180
Figure 5.11 (a) Robustness against MP3 attacks at 64 kbps, 96 kbps (b) Robustness against MP3 attacks at 128 kbps, 160 kbps 182
Figure 5.12 *Precision* achieved after four different attacks without using repetition 183
Figure 5.13 (a) *Precision* distribution of 25 music files without using repetition (without any attack) (b) *Precision* distribution of 25 music files after using repetition (after 64 kbps MP3) .. 185

List of Tables

Table 2.1 The MOS Score description 18
Table 2.2 The ODG Score description 21
Table 2.3 The mapping between MOS and ODG 58
Table 2.4 The four main characteristics of each typical audio watermarking algorithm 59
Table 2.5 The other characteristics of each typical audio watermarking algorithm 60
Table 3.1 Configuration of the coefficients α_1 and β_1 79
Table 4.1 Threshold and *r* setting of each different group 120
Table 4.2 Bit and Symbol errors distribution for each file 135
Table 4.3 Performance comparison when using different frame lengths 139
Table 4.4 Performance comparison of audio watermarking schemes 157
Table 5.1 Mean and standard deviation of *Precision* against different attacks 184
Table 5.2 Performance comparison of audio watermarking schemes 186

List of Publications

[1] J. Wang, R. Healy, and J. Timoney, "Digital Audio Watermarking by Magnitude Modification of Frequency Components Using the CSPE Algorithm," *China-Ireland International Conference on Information and Communications Technologies 2009*, NUI Maynooth, Ireland, 2009.

[2] J. Wang, J. Timoney, and M. Hodgkinson, "Using Apodization to improve the performance of the Complex Spectral Phase Estimation (CSPE) Algorithm," *China-Ireland International Conference on Information and Communications Technologies 2009*, NUI Maynooth, Ireland, 2009.

[3] M. Hodgkinson, J. Wang, J. Timoney, and V. Lazzarini, "Handling inharmonic series with median-adjustive trajectories," *The 12th International Conference on Digital Audio Effects (DAFX)*, Como, Italy, 2009.

[4] J. Wang, R. Healy, and J. Timoney, "Perceptually Transparent Audio Watermarking of Real Audio Signals Based On The CSPE Algorithm," *Signals and Systems Conference, IET Irish*, UCC, Ireland, 2010.

[5] J. Wang, R. Healy, and J. Timoney, "A novel audio watermarking algorithm based on reduced singular value decomposition," *The sixth International conference on intelligent information hiding and signal processing*, Darmstadt, Germany, 2010.

[6] J. Wang, R. Healy, and J. Timoney, "An improved audio watermarking scheme based on complex spectral phase evolution spectrum," *The 129th Convention,*

Audio Engineering Society, San Francisco, USA, 2010.

[7] J. Wang, R. Healy, and J. Timoney, "A robust audio watermarking scheme based on reduced singular value decomposition and distortion removal," *Elsevier*, *Signal Processing*, vol. 91 (8), pp. 1693-1708, August 2011.

[8] A. Mooney and J. Wang, "Watermarking", ISBN: 979-953-307-583-8.

[9] J. Wang, R. Healy, and J. Timoney, "Audio Watermarking," US Patent, Application No. 61/356,924, International Patent Application No. PCT/EP2011/059688 .

[10] J. Wang, R. Healy, J. Timoney, D. Reid, and G. Meade, "Voice mail Protection System," United Kingdom Patent Application No. 1118706.9.

[11] J. Wang, J. Timoney, "A new watermarking algorithm based on CSPE and SVD," paper to be submitted.

[12] J. Wang, J. Timoney, "A new fingerprinting algorithm based on CSPE," paper to be submitted.

[13] J. Wang, "A new fingerprinting algorithm based on SVD," paper to be submitted.

Chapter 1 General Introduction

1.1 Introduction

Technological advances in computing, communications, consumer electronics and their convergence have resulted in phenomenal increases in the amount of digital content that is being generated, stored, distributed, and consumed [SY06]. The term "content" broadly refers to any digital information, such as digital audio, video, graphics, animation, images, text, or any combinations of these types. This digital content can be easily accessed, perfectly copied, rapidly disseminated and massively shared without it losing quality [DD03], as opposed to the situation with earlier analogue media, such as audio cassettes and Video Home System (VHS) tapes. However, these advantages of digital media formats over analogue transform into disadvantages with respect to copyright management [Cve04], because the possibility of unlimited copying without a loss of fidelity has led to a considerable financial loss for copyright holders [CMB02, WL03, YKL01].

In terms of a solution to the financial losses incurred from unauthorised copying, content owners predominantly turn to cryptography, which is one of the most commonly used methods of protecting digital content. In the cryptography process, the content is encrypted prior to the delivery to the consumer, and then a decryption key is provided only to those who have purchased legitimate copies of the content. However, cryptography does not offer a robust solution to content piracy. For example, a pirate could purchase the encrypted content legitimately and then use the decryption key to

produce and distribute copies of the content illegally. In other words, once decrypted, the content has no further protection. Thus, there is a strong need for an alternative or complement to cryptography. In terms of the solution to the problems encountered with cryptography, watermarking has been proposed as it has potential to offer more robustness. It can protect the digital content during its normal usage because the copyright information is placed within the digital content in such a way that it cannot be removed. This unique feature of watermarking makes it one of the most promising techniques for digital content protection [Sin11], which has been the motivating factor behind much of the research in the last two decades.

This chapter is organized as follows: First, a brief history of watermarking is given followed by an overall illustration of how watermarking algorithms work. Then, typical applications of watermarking are introduced. The motivation and contributions of this thesis are explained subsequently. Finally, an overview of this thesis is provided.

1.2 Brief history of watermarking

The concept of watermarking has evolved from paper watermarking to digital watermarking. This concept has a long history, which can be traced back to 1282 when paper watermarks first appeared in Italy [KP00]. The watermarks were made by adding thin wire patterns to the paper moulds during the manufacturing process. The paper would be slightly thinner where the wire patterns were added and hence more transparent. The exact reason and purpose of the introduction of this watermark are uncertain. They

may have been used for practical functions such as identifying the mould on which sheets of papers were made, or as trademarks to identify the paper maker [KP00].

By the eighteenth century, watermarks on paper made in Europe and America had become more clearly utilitarian [CMB02]. They were used as trademarks to record the date the paper was manufactured, and to indicate the sizes of original sheets. It was also about this time that watermarks began to be used as anti-counterfeiting measures on money and other documents. The term watermark appears to have been coined near the end of the eighteenth century and may have been derived from the German term wassermarke [SW00].

In 1954, Emil Hembrooke of the Muzak Corporation filed a patent for "watermarking" musical works. An identification code was inserted in music by intermittently applying a narrow notch filter centred at 1 kHz. The absence of energy at this frequency indicated that the notch filter had been applied and the duration of the absence used to code either a dot or a dash. The identification signal used Morse code. The 1961 U.S. Patent describing this invention states [Hem61]:

"*The present invention makes possible the positive identification of the origin of a musical presentation and thereby constitutes an effective means of preventing such piracy*".

This system was used by Muzak until around 1984 [Wal01]. It is interesting to note that there was speculation at the time that this invention was delivering subliminal advertising messages to its listeners [CMB+07].

It is difficult to exactly determine when the concept of digital watermarking was first proposed. For example, in 1979, Szepanski [Sze79] described a machine-detectable pattern that could be placed on documents for anti-counterfeiting purposes. Nine years later, Holt et al. [HMW88] described a method for embedding an identification code in an audio signal. However, it was Komatsu and Tominaga [KT88], in 1988, who appeared to have first used the term digital watermark. It was not until the mid 1990's that interest in digital watermarking began to soar [CMB02], and this interest has continued until the present day.

1.3 Illustration of watermarking algorithm

In this section, the definition and characteristics of a watermarking algorithm will be illustrated.

1.3.1 Watermarking definition

Digital watermarking is the process by which a discrete data stream called a watermark is embedded within a digital content [KH01, GH99]. It is a special form of steganography, which is concerned with developing methods of writing hidden messages in such a way that no one, apart from the intended recipient, knows of the existence of the message [Bla10]. Generally, the process of watermarking can be divided into two parts: embedding and detection, the embedding process is depicted in Figure 1.1. As seen from Figure 1.1, the process of watermark embedding is very straightforward, that is, the watermark is generated and then is embedded into the original content by some means. A

variety of embedding algorithms have been developed which rely on the manipulation of some properties of the original content.

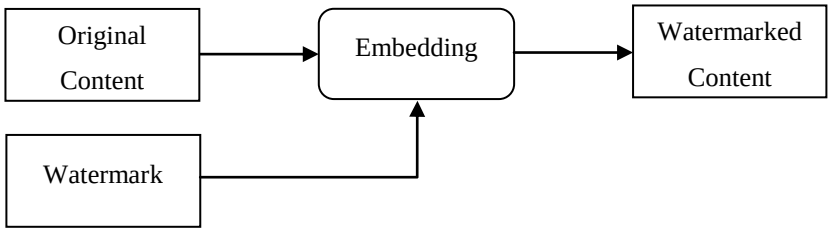


Figure 1.1 The flowchart of the embedding process

The detection process, as depicted in Figure 1.2, illustrates the embedded watermark information being recovered only by the use of the key.

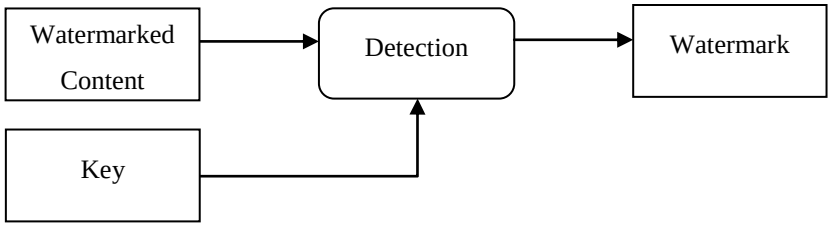


Figure 1.2 One type of detection processes

A more complex detection process, as depicted in Figure 1.3, illustrates the detection of the presence of watermark information, which requires not only the key, but also the original watermark and/or the original content. Compared with the detection process described in Figure 1.2, this needs extra storage capability for the original watermark information or original host information [CCL07]. However, the advantage of using the detection process described in Figure 1.3 is that the detection performance can be greatly improved when the original host or original watermark bit sequence is available [NH00].

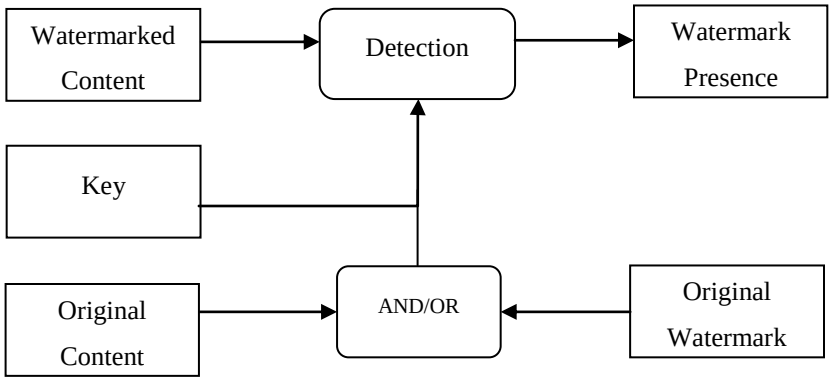


Figure 1.3 The other type of detection processes

1.3.2 Watermarking characteristics

A watermarking algorithm has the following characteristics [NC06, Cve07]:

1. ***Imperceptibility***: in general, the embedded watermark should not affect the human perception of the content. Namely, the watermark should be "invisible" in an image/video or "inaudible" in audio. However, in some special application scenarios, the watermark should be obtrusive to serve as a statement of ownership. Therefore, watermarking algorithms can be classified as 'perceptible' where the embedded watermark can be perceived and 'imperceptible' where the embedded watermark cannot be perceived.
2. ***Robustness***: any manipulation of the watermarked content is defined as an attack [LDSV05, SPR+01]. The embedded watermark should be robust against attacks. That is, the watermark recovery accuracy should not be decreased significantly after the watermarked content is attacked. Attacks on watermarks can be accidental or intentional. Accidental attacks are the result of standard signal processing that the

signal might undergo [JDJ99], such as Analogue to Digital (A/D) conversion, Digital to Analogue (D/A) conversion and lossy compression. Intentional attacks refer to those that deliberately distort or remove the embedded watermarks [JJ98, PAK98]. As far as audio is concerned, the main attacks, both intentional and accidental, can be divided into the following groups:

I. Filtering: such as highpass filtering, lowpass filtering and equalization. An equalizer only increases or decreases specified spectral regions.

II. Lossy compression: such as MP3 or Advance Audio Coding (AAC), which are used to reduce the amount of audio data.

III. Noise: such as noise adding or removal.

IV. Conversion: such as A/D, D/A or conversion of the sampling frequency (for example, from 32 kHz to 48 kHz).

V. Time stretch: increasing or decreasing the duration of an audio signal without changing its pitch.

VI. Pitch shift: change the pitch without changing the speed of the audio.

The task of designing a robust watermarking algorithm, which is able to withstand all or even a subset of possible attacks, appears to be quite difficult [Arn01]. Each algorithm currently proposed has its own weakness against certain types of attack. However, not every attack is possible with particular applications. Thus, identifying potential attacks that are associated with a specified application is essential [AH04].

The most powerful attacks are those that can remove or distort the watermark information without severely degrading the content quality. With these attacks, the watermark information cannot be recovered but the audio can be used normally [SPR+01]. Watermarking algorithms can be classified as ‘robust’, ‘fragile’ and ‘semi-fragile’ according to their robustness against attacks.

3. ***Capacity***: the watermarking algorithm should be capable of embedding a large amount of watermark information into the digital content.
4. ***Blindness***: in general, the watermarking algorithm should be blind, that is, the embedded watermark can be detected without requiring access to the original content or original watermark. However, some watermarking algorithms can only detect the presence of the original watermark. Therefore, watermarking algorithms can be classified as ‘blind’ and ‘informed’ in terms of how much information is required at the detection stage. The ‘blind’ watermarking algorithm only requires a ‘key’ to detect the watermark. While the ‘informed’ watermarking algorithm needs the original watermark or the original content to detect the presence of the watermark information.
5. ***Computational efficiency***: the efficiency of the watermarking algorithm will determine if it can be applied in time-critical applications.
6. ***Security***: Kerchhoff’s Principle states that a cryptosystem should be secure even if everything about the system, except the key, is publicly known [Sch96]. This principle was reformulated by Claude Shannon as “the enemy knows the system”

[BP10], which is embraced by cryptographers worldwide. As far as watermarking systems are concerned, the algorithm might be published or made public. However, an unauthorized user, who may even know the exact watermarking algorithm, should not be able to detect the embedded watermark unless the secret keys are disclosed.

7. ***Adjustability***: The algorithm should be tuneable to various degrees of robustness, imperceptibility and capacity to facilitate diverse applications [VKK09].

1.3.3 Watermarking trade-off

The imperceptibility, robustness and capacity are the three most important characteristics of a watermarking algorithm. However, there is a trade-off between these three characteristics [BSD10, OB08, DML+06]. This trade-off can be represented as shown in Figure 1.4.

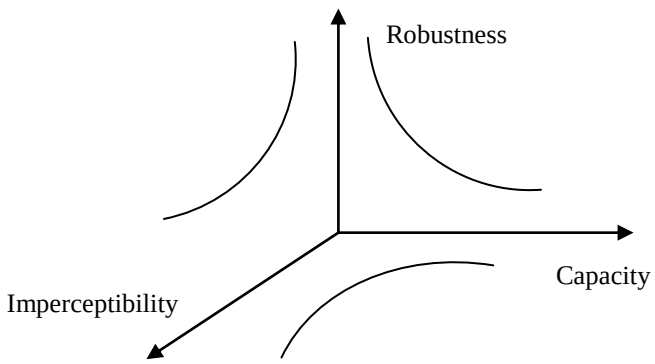


Figure 1.4 Trade-off presents in watermarking system

As seen from Figure 1.4, imperceptibility, robustness and capacity are conflicting characteristics of a watermarking system [EE05]. For example, a specific application may determine what the capacity is needed. After it is determined, there exists a trade-off between imperceptibility and robustness. If one then wants to make the watermark more

robust against attacks, a larger modification of the signal's properties to embed the watermark will be necessary. However, this will worsen the imperceptibility. Another scenario may be that with a predefined requirement for the imperceptibility, there will exist a trade-off between the capacity and robustness. For instance, the fewer the message bits that are embedded, the more redundant the watermark can be. Therefore, the watermarking will have a better error correction capability against attacks, that is, it is more robust [Lin00].

1.4 Applications of watermarking

Although copyright protection has been the major driving force behind research in the watermarking field, there are a number of other applications for which watermarking has been used or suggested [CMB01], for example, authentication and tamper proofing of digital content [SS11, Kun01]. This can be solved by watermarking [PPB10, QX11, KH99]. In the following sections, different applications of watermarking algorithms will be illustrated.

1.4.1 Copyright protection

For the protection of intellectual property, the content owner can embed a watermark representing copyright information in their content. This watermark can prove their ownership in court when someone has infringed on their copyright [LSL00].

1.4.2 Authentication and Tamper proofing

Due to the proliferation of freely available audio editing software, it has become easier to modify or forge digital content. There has been a recent trend toward addressing these

problems by using a digital watermarking algorithm [CHL06, JA08, LL08, Pha08, CW09]. One of the main advantages of using digital watermarking is that it does not require any additional memory. In addition, the algorithm can be designed to make the watermark sensitive to intentional attacks, but robust to accidental attacks such as lossy compression and channel noise, so that accidental modification will not affect the integrity or credibility of the digital content [RR11, Que01]. The term "credibility" refers to the fact that the content is authentic and has not been tampered with [KH99]. Furthermore, watermarking can be designed to identify which parts of the content have been tampered with [CH11, LC01, LL01, WM01, LLC06].

1.4.3 Identification and Tracking of digital content

The identification of publicly available digital content, whether on the Internet or traditional broadcast media, is important to a variety of users [TDE04]. It is important to end users who need accurate metadata when searching for desired music, important to rights organizations who distribute royalties, and important to content owners who wish to protect their intellectual property rights. Based on the identification of the content, the tracking of digital content can be implemented, which can serve a variety of purposes including [TDE04]:

1. ensuring the rights holder's royalties for material that is played on broadcast
2. building a statistical analysis of broadcast material for chart creation
3. verifying the scheduled transmission of advertisement spots

Identity information (ID) can be embedded into the digital content by a watermarking algorithm. Simply by detecting the embedded watermark, the content can then be identified. A block diagram of content identification based on watermarking is described as Figure 1.5. From Figure 1.5, it can be seen that the ID is embedded into the original content first, and then this ID can be detected at a later stage to identify the digital content precisely.

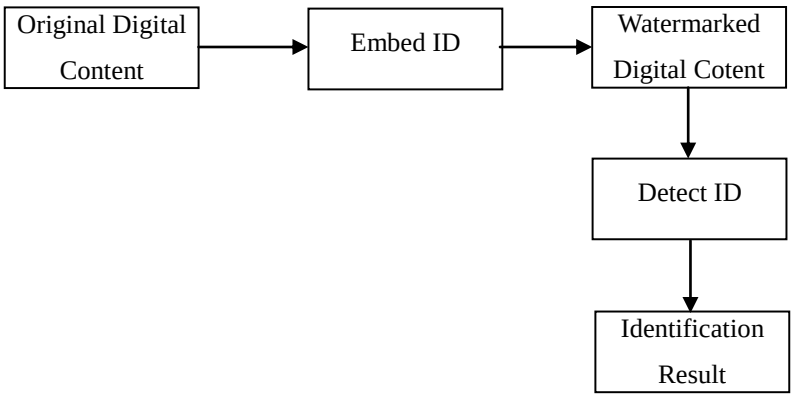


Figure 1.5 Content identification using digital watermarking

1.4.4 Copy prevention

The information stored as a watermark can be used directly to control digital recording devices for copy prevention purposes, where the watermark represents a copy-prohibit bit. The watermark detector in the recorder can use this bit to determine whether the content can be recorded or not [LSL00].

1.5 Motivation

The initial inspiration for the research undertaken in this thesis was the work of [Hea10], which motivated the development of new audio watermarking algorithms. The new

algorithm should include the following essential characteristics: imperceptible, robust and have a high capacity and computational efficiency so that it has the potential to be deployed commercially.

To satisfy these goals, two algorithms from distinct perspectives were developed. One was based on the Complex Spectral Phase Evolution (CSPE) and the other was based on the Singular Value Decomposition (SVD). These two developed watermarking algorithms have their own advantages and disadvantages and are complementary in their performance, which open possibilities for future fusion of these two algorithms.

1.6 Contributions

The overall original contributions made by this thesis are listed below:

1. The CSPE algorithm was improved by using apodization, so that more frequency components could be identified.
2. A novel approach of embedding watermarks by manipulating the frequency components in the time domain was devised, rendered feasible by using the CSPE algorithm.
3. A dynamic embedding position generation process was developed. This means that it is harder for an attacker to deduce where the watermark is embedded, because each embedding position is dependent on the signal properties of each specific frame.
4. A novel audio watermarking algorithm based on the CSPE algorithm was proposed and then subsequently improved. This algorithm achieved an acceptable performance, especially demonstrating a high imperceptibility and robustness.

5. A new observation regarding the application of SVD to watermarking was found. Based on this observation, a new watermarking algorithm was developed that introduced less distortion compared to other SVD based audio watermarking algorithms.
6. A novel way of removing the windowing effect when reconstructing the time domain watermarked signal was proposed.
7. An audio watermarking algorithm based on SVD was developed, which achieved an acceptable performance, especially in terms of its robustness and capacity.

1.7 Thesis Overview

This thesis focuses on audio digital watermarking algorithms and presents two new techniques.

Chapter 2 evaluates the existing audio watermarking algorithms. The advantages and disadvantages of each watermarking algorithm are assessed according to defined criteria. Based on my literature review, two different types of watermarking algorithms are found to be worthy of further investigation. One is a Fast Fourier Transform (FFT) based watermarking algorithm and the other is a SVD based watermarking algorithm. The merits and shortcomings of these two algorithms are assessed and the requirements for the thesis work are established.

Chapter 3 presents a CSPE based watermarking algorithm. The CSPE is a super-resolution spectral analysis tool that offers far greater accuracy than the standard FFT. This proposed watermarking algorithm is applied to real audio tracks and it is found that

the imperceptibility and accuracy are unacceptable. This algorithm is further improved by employing the component verification process and click removal process. Finally, the experimental results show that the improved algorithm could achieve a very good imperceptibility and accuracy.

Chapter 4 investigates the robustness of the improved algorithm proposed in Chapter 3 and demonstrates that it is not strong enough against certain attacks, motivating further improvement. A new observation concerning the CSPE spectrum is discovered, and based on this observation, a new CSPE based watermarking algorithm is developed. Its performance in terms of imperceptibility, robustness, capacity, and computational efficiency are examined and found to be acceptable.

Chapter 5 outlines the development of a SVD based audio watermarking algorithm. There are two reasons for this development. The first reason is that a high level of robustness can easily be achieved using SVD according to a literature review. The second reason is that the capacity of the algorithm proposed in Chapter 4 is not high enough for some specific applications. It is demonstrated that the addition of a psychoacoustic model improves the imperceptibility of the algorithm. The overall performance of this algorithm is compared to that of other existing SVD based audio watermarking algorithms, and is found to achieve a better performance.

Chapter 6 gives conclusions of the work presented in this thesis and offers suggestions for future work.

Chapter 2 Literature Review

2.1 Introduction

In order to acquire a better understanding of digital audio watermarking and then identify some of the unresolved issues within the current implementations, a thorough literature review is undertaken. The reviewed algorithms are diverse and so are divided into different categories such as time domain based algorithms, transformation based algorithms and hybrid algorithms according to the methodology each algorithm utilized. The advantages and disadvantages of the important algorithms in each category are examined according to the criteria defined below:

1. The performance in terms of imperceptibility, robustness, capacity, computational efficiency and the blindness of the algorithm.
2. The reliability of the results presented for each algorithm. This involves investigation of the test methodology employed.
3. Establishing whether the watermark bits embedded are easily removed.
4. Establishing whether the algorithms incorporate some additional processes to circumvent the trade-off between the imperceptibility and the robustness.
5. Establishing whether the algorithms carry out a comprehensive analysis on how each parameter affects the performance.

The reason for defining the above criteria is that they serve as important factors for evaluating a watermarking algorithm when applied in practice. In addition, these criteria

are useful for determining whether the algorithms under review are worthy of further research.

The organization of this chapter is as follows: Firstly, the quantitative evaluation approaches for the algorithms are introduced. Then, the algorithms belonging to each category are investigated with their performance assessed. All the performance results are derived from the reviewed papers. Finally, the research direction and the research methodologies to be used in this thesis are outlined based on the reviews.

2.2 Quantitative evaluation of the audio watermarking performance

Before reviewing each algorithm, the methodology of assessing the performance of an audio watermarking algorithm quantitatively is outlined. As mentioned in Chapter 1, imperceptibility, robustness, capacity and computational efficiency are the main characteristics that are frequently used to evaluate the performance of watermarking. The quantitative evaluations of these characteristics are described in detail as follows:

2.2.1 Imperceptibility

In general, there are three approaches to evaluate perceptual quality of audio [BSD10]:

1. Subjective evaluation by a human listening test
2. Objective evaluation by signal-oriented measures such as the Signal to Noise Ratio (SNR)
3. Objective evaluation that incorporates one of the Human Auditory System (HAS) models such as the Perceptual Evaluation of Audio Quality (PEAQ)

Subjective evaluation can be conducted in a number of ways. One approach is to use an ABX test [DP09]. Each test case is composed of the original audio file *A*, the watermarked audio file *B* and an undefined audio file *X* which could be either *A* or *B*. The listener is asked to identify whether *X* is *A* or *B*. A highly correct identification rate suggests that the watermark is perceptible, while an approximately 50% correct identification rate means that watermark is imperceptible because the identification is similar to a random guessing.

In addition, the Mean Opinion Score (MOS), specified by ITU-T recommendation P.800 [ITU96], can be used to grade the subjective listening quality of the watermarked content [YSZ11]. The five scales of the MOS score are shown in Table 2.1[YSZ11]:

Table 2.1 The MOS Score description

MOS	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

However, subjective evaluation based on human listening tests is time consuming and the results can be inconsistent across different listeners [AS02]. The reason for this inconsistency is that the hearing ability of different listeners varies depending on their age, lifetime exposure to loud noises and even personal preference in musical tastes, not

to mention that some listeners may be trained as expert listeners [Swe96]. Therefore, it is sometimes difficult to compare across different subjective evaluation results fairly, and it is desirable to have a more objective, signal-based measure available.

The SNR is widely utilized as an objective measure of audio quality. It is simple to interpret, straightforward to apply and is signal oriented [EB09]. According to IFPI [Ifp09], when the SNR is above 20 decibels (dB), audio watermarking should be imperceptible. The SNR can be formulated as follows:

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{\mathrm{n}} \mathrm{s}^2(\mathrm{n})}{\sum_{\mathrm{n}}[\mathrm{s}(\mathrm{n}) - \mathrm{s}'(\mathrm{n})]^2} \tag{2.1}$$

where $s(n)$ is the time domain original signal and $s'(n)$ is the time domain watermarked signal.

Since Equation (2.1) weights all time domain errors equally, not taking any time-varying energy and time-varying distortion into account, an improved measure can be obtained if the SNR is calculated over short frames and the results averaged. A frame-based measure called the 'Segmental Signal to Noise Ratio' ($\mathrm{SNR_{seg}}$) is defined as follows [DPH93]:

$$\mathrm{SNR_{seg}} = \frac{1}{M}\sum_{j=1}^{M} 10\,log_{10}\left[\sum_{n=N*(j-1)+1}^{N*j}\frac{s^2(n)}{[s(n)-s'(n)]^2}\right] \tag{2.2}$$

where *M* is the number of frames and *N* is the frame size.

Problems arise with the $\mathrm{SNR_{seg}}$ if frames of silence are included as they can lead to large negative $\mathrm{SNR_{seg}}$ values. This problem can be solved by setting a low threshold and replacing all frames with a $\mathrm{SNR_{seg}}$ below this threshold equal to the threshold instead

(a 0-dB threshold is reasonable) [DPH93]. At the other extreme, frames with a SNR_{seg} greater than 35 dB are not perceived by listeners as being significantly different but affect the resulting SNR_{seg}. An upper threshold (normally 35 dB) can be used to reset any unusually high SNR_{seg} values to this upper threshold [DPH93].

A low SNR or SNR_{seg} clearly indicates that the distortion introduced by watermarking is audible, but a high SNR or SNR_{seg} is not sufficient to indicate that the watermark is perceptually transparent because this measure does not consider any HAS model. From the many experiments carried out, the behaviour of HAS has been explored extensively [Moo91, ZF90]. The field of psychoacoustics [Hel72, ZF90, ZZ91] has made significant progress towards characterizing the HAS. Some terminologies such as "absolute hearing thresholds", "simultaneous masking" and "temporal masking" have been proposed [Pai00]. The "absolute hearing thresholds" characterizes the amount of energy needed in a pure tone such that it can be detected by a listener in a noiseless environment. "Masking" refers to a phenomenon where one sound is rendered inaudible because of the presence of another sound. This phenomenon can occur in the frequency domain which is termed as "simultaneous masking", or in the time domain, which is termed as "temporal masking". In order to reflect human perception more accurately, it is better to have an objective assessment that incorporates one of the HAS models.

The PEAQ is one of these objective assessment methods. It was defined as recommendation standard BS.1387 [Kab03] and has been used for more than ten years [CKL+97]. EAQUAL is a software implementation of PEAQ [MJM05]. The output of

the PEAQ is the Objective Difference Grade (ODG). This classifies the perceptual differences between the original and the watermarked audio signal. The ODG value is in a range of [-4, 0], as shown in Table 2.2, with 0 indicating that the two signals are perceptually identical and -4 indicating that the perceptual differences between the two are 'very annoying'. Therefore, the closer the ODG score is to zero, the more likely the signals are perceived as being identical.

Table 2.2 The ODG Score description

ODG	Description
0	Imperceptible
-1	Perceptible, but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

The correlation between the PEAQ and subjective listening test was investigated [CKL+97]. It was found that the correlation coefficient between both was 0.837 and 0.851 for the Basic version and the Advanced version of PEAQ respectively [CKL+97]. It is certain that the PEAQ cannot be a complete replacement for subjective listening test [MAK08], but it is a broadly accepted objective measure of the audio quality in industry [YK03] and has been used widely to evaluate the imperceptibility of watermarking algorithms [FM09, MRF10] .

2.2.2 Accuracy

The accuracy of a watermarking algorithm is defined as the watermark detection precision without undergoing any attack. It can be measured by the Bit Error Rate (BER) [BSD10], which is defined as follows:

$$\text{BER}\ (W_1, W_2\ ) = \frac{\sum_{i=1}^{N} W_1(i) \oplus W_2(i)}{N} \tag{2.3}$$

where W_1, W_2 denotes the original watermark bit sequence and the detected watermark bit sequence respectively, *N* denotes the number of bits and *i* denotes the bit number. In this thesis, '*Precision*' is used to evaluate the performance of the accuracy, as it is more straightforward. This is defined as follows:

$$Precision\ (W_1, W_2\ ) = \frac{N - \sum_{i=1}^{N} W_1(i) \oplus W_2(i)}{N} = 1 - BER\ (W_1, W_2\ ) \tag{2.4}$$

The meaning of each variable is the same as that defined in Equation (2.3). If *N* audio signals are used in an experiment, the mean precision, denoted as $Precision_{mean}$, is calculated as Equation (2.5), where *i* denotes the signal number.

$$Precision_{mean} = \frac{\sum_{i=1}^{N} Precision_i}{N} \tag{2.5}$$

2.2.3 Robustness

The robustness of a watermarking algorithm is defined as the watermark detection accuracy after undergoing attacks. It also can be measured by the BER [BSD10]. Likewise, *Precision* is used to evaluate the robustness for the same reason as stated in Section 2.2.2.

Generally, in order to improve the robustness, a 'repetition' process is incorporated into the scheme. This embeds the same watermark bit sequence repetitively. At detection side, the 'mode' operation, as defined in statistics [Gri08], is used to identify the watermark bit sequence. In statistics, the 'mode' operation is used to find the data that occurs most frequently in a defined data set [Gri08]. For example, in the dataset {0, 1, 0, 1, 1}, the mode is '1' as it occurs one more time than '0'. The procedure of incorporating the 'repetition' process into the watermarking scheme can be formalized as follows:

1. Generate a watermark bit sequence B_w for the signal to be watermarked.
2. At the embedding stage, embed B_w repeatedly into the signal, say d times.
3. At the detection stage, the detected bit sequence B_e is split into d groups: B_{e1}, B_{e2}, ..., B_{ed}. The i^{th} bit of the detected watermark bit sequence B'_w is determined as the mode of the bit set $\{ B_{e1,i}, B_{e2,i}, ..., B_{ed,i} \}$.

2.2.4 Capacity

The capacity of the watermark can be measured as the number of bits per second (bps). Suppose the length of the host audio is k seconds and the number of embedded watermark bits is n [BSD10], then the capacity is $\frac{n}{k}$ bps.

2.2.5 Computational efficiency

The computational efficiency can be assessed as the CPU time required for watermark embedding and detection. This is dependent on the implementation platform.

2.3 Review of audio watermarking algorithms

The variety of algorithms available can be divided into different categories according to the methodology they employ. The vast majority of audio watermarking algorithms fall into three categories:

1. Time domain based algorithms
2. Transformation based algorithms
3. Hybrid algorithms

The watermarking algorithms belonging to each of these three categories will be reviewed in detail in the following sections.

2.3.1 Time domain based algorithms

Time domain based algorithms, literally, embed the watermark in the time domain. They are simple to implement [OB08]. Many time domain based algorithms have been developed [XM06, KNS05, OSHY01, LS04, FLK06]. However, algorithms belonging to this category are less robust against attacks and statistical techniques are often utilized to improve the robustness [WNY09]. Two main algorithms belonging to this category will be reviewed, that is, a Least Significant Bit (LSB) based algorithm and an Echo hiding based algorithm.

2.3.1.1 LSB coding

LSB watermarking is one of the earliest techniques [YK99, CAM00] in audio watermarking as well as for other media types [FGD01, LC00, FGD02]. The standard approach is to embed the watermark bits by altering the values of particular samples in

the digital audio. The watermark bits are detected by comparing the altered values of samples with the original values of samples.

The main advantage of this algorithm is that it can achieve an extremely high capacity. The primary disadvantage is its extremely low robustness because random changes of the signal can destroy the watermark [Mob98]. It is very unlikely that the embedded watermark bits can survive D/A and subsequent A/D conversions [Mob98]. In addition, the alteration of the sample values introduces a low power Additive White Gaussian Noise (AWGN), which makes this algorithm less perceptually transparent because listeners are very sensitive to this noise [CS05].

A major improvement on the standard LSB algorithm was proposed in [CS05]. The basic idea is that after embedding the watermark bits by manipulating one bit of a 16-bit wav sample, all other 15 bits of a sample may be altered as well so that the difference between the original sample value and the manipulated sample value is minimized. As a result, this introduces less distortion. For example, if an original 16-bit sample value is ‘0000000000001000’ in binary, and the watermark bit to be embedded is a '0'. Suppose the watermark bit will be embedded into the 4^{th} last bit of the sample, instead of producing the value ‘0000000000000000’ in binary as per the standard algorithm, the improved algorithm also flips the first three bits of the sample producing the value ‘0000000000000111’ in binary. As a result, the difference between the original sample and the manipulated sample is only 1 in decimal, which is the closest to the original sample value. Thus, the distortion introduced is minimized. This example of

manipulating a sample value by the standard LSB and the improved LSB respectively is shown in Figure 2.1, where '①' represents the original sample value, '②' represents the sample value after manipulation by the standard LSB, '③' represents the sample value after manipulation by the improved LSB.

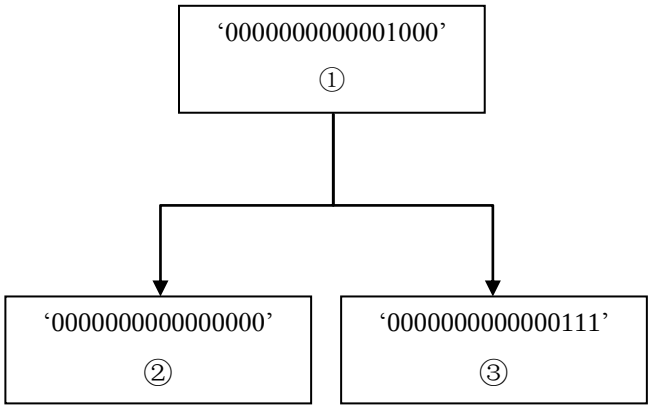


Figure 2.1 The demonstration of how sample value is altered by different LSB algorithms [CS05]

According to the experimental results [CS05], it was shown that the imperceptibility achieved a MOS score of about 5.0, that is, it is perceptually transparent. However, this improved version of the algorithm does not enhance the robustness to a significant extent.

2.3.1.2 Echo hiding

Echo hiding based watermarking embeds a watermark bit by introducing an 'echo'. An echo is a reflection of sound, arriving at the listener some time after the direct sound [GLB96]. Four parameters of the echo are used: initial amplitude, decay rate of the echo amplitude, 'one' offset (delay time to the original signal) and 'zero' offset. When the offset between the original and the echo decreases, the two signals blend. At a certain

point, the human ear does not hear an original signal and an echo, but a single blended signal. The point at which this happens is hard to determine exactly. It depends on the quality of the original recording, the type of sound being echoed, and the listener. The algorithm uses two different kernels, a ‘one’ kernel that is used to generate a ‘one’ offset echo to represent a binary ‘1’, and a ‘zero’ kernel that is used to generate a ‘zero’ offset echo to represent a binary ‘0’ [GLB96]. The two kernels are depicted in Figure 2.2.

(a) **(b)**

Figure 2.2 (a) “one” kernel (b) “zero” kernel

From Figure 2.2, it can be seen that the difference between the ‘one’ kernel and the ‘zero’ kernel is the length of their delay time: δ_1 and δ_0 respectively. Both delay times are below the threshold at which the human ear can discriminate between the echo and the host. Denoting $s(n)$ as the original signal, $k(n)$ as the echo kernel used to generate an echo, the echo system output $w(n)$ can be derived by convoluting $s(n)$ with $k(n)$.

At the embedding stage, the signal is split into different frames, then each frame is convoluted with a ‘one’ offset echo or a ‘zero’ offset echo depending on the embedded watermarking bit value. At the detection side, the echoed signal is analyzed by the cepstrum first, followed by the autocorrelation to get the power of the signal [GLB96]. The cepstrum is defined as the power spectrum of the logarithmic FFT power spectrum

[BHT63]. A ‘power spike’ will appear at each periodical δ_1 or δ_0 location, which can be detected as a watermark bit ‘1’ or ‘0’. Based on the experiments carried out [GLB96], it was found that the offset value is crucial for the robustness, and the decay rate is crucial for the imperceptibility. The experiment showed that this algorithm is robust against MP3 compression, A/D and D/A attacks. The disadvantages of [GLB96] are as follows:

1. Only three tracks were used as the test data, thus further investigation on larger data set is needed to verify the claims for its performance.
2. The imperceptibility and the bit rate of MP3 compression were not detailed.
3. The detection rules are very lenient because anyone can detect the watermark bits without requiring any key. Thus, it is very vulnerable to malicious tampering [KNS05, KP00].

To overcome these disadvantages, several different embedding processes based on echo hiding have been proposed. One of these is multi-echo embedding [XWSX99]. However, this algorithm has limitations in the allocation of the delay time of the echoes. The reason for these limitations is that preserving imperceptibility is difficult to realize in the case of multiple echoes.

In [KNS05], a Time Spread (TS) echo was proposed as an alternative to a single echo or a multi-echo. The basic idea had been previously proposed in [KNS02]. At the embedding stage, an echo is spread by a Pseudo Noise (PN) sequence, thus the amplitude of each echo becomes small and results in its power spectrum being nearly flat in the mean time sense. At the detection stage, all the processes are similar to the conventional

echo hiding approach, except for an additional process, that is, the cepstrum has to be despread by the original PN sequence in order to detect the echo. Thus, without the original PN sequence, the echo cannot be detected, which enhances the security of the algorithm proposed in [KNS05].

The performance of this algorithm was evaluated [KNS05]. The robustness against attacks is generally half of that in the case of no attack. Therefore, the proposed method is unable to achieve a strong robustness. Other spreading kernels such as chaotic sequences [Kub95] and time-stretching pulses [SAKS95] were suggested because they might improve the robustness. As far as the imperceptibility is concerned, an ABX test was carried out and it was found that this algorithm achieved a better imperceptibility than the single echo algorithm. A relatively detailed analysis on how each parameter affects the performance of the algorithm was provided in [KNS05]. However, the imperceptibility is very sensitive to the parameters such as the length of PN sequence, the amplitude of PN sequence and the amplitude of the echo. The test data is only composed of five different genres of music tracks so that the evaluation was limited.

Another TS echo hiding algorithm was proposed in [ES09]. The watermark bit was detected through the correlation between the cepstrum and the PN sequence. According to the experimental results presented [ES09], its robustness against signal processing attacks was poor, for example, a BER of 47% against a MP3 attack and a BER of 12.5% against a noise attack. The subjective listening test was acceptable with a MOS score of 4.7.

2.3.2 Transformation based algorithms

Transformation based algorithms generally embed watermark bits by exploiting the properties of the data in the representation following the transformations. Popular transformations are the Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) [EB09, WCC04, PW02, VTCL05, QZ04, WZ06]. Some techniques, such as Quantization Index Modulation (QIM), Singular Value Decomposition (SVD), and interpolation, are often utilized to manipulate the data to embed watermark bits in the representation following transformations. Many watermarking algorithms fall into this category because the watermark bits embedded are more robust against attacks.

A HAS model is generally incorporated to minimize the perceptual distortion introduced [OB08]. However, a tradeoff exists here because embedding watermark bits in perceptually significant components is more robust but less perceptually transparent. On the other hand, embedding watermark bits in less perceptually significant components is less robust but more perceptually transparent. In addition, incorporating a HAS model would increase the computation time [KCSH03], which restricts the use of these algorithms in time critical applications. In the following sections, typical algorithms in this category will be reviewed.

2.3.2.1 FFT based audio watermarking algorithms

FFT has been developed as a fast version of the Discrete Fourier Transform (DFT) [CT65]. The DFT is a well known and powerful computational tool for performing

frequency analysis of discrete time signals [Jay08]. It takes a discrete signal in the time domain and transforms this signal into the discrete frequency domain [Jay08].There has been a variety of watermarking algorithms proposed that are based on manipulating the components contained in the FFT spectrum. Most algorithms manipulate the magnitude of the FFT components [MJM03, MJM05, FM09, MRF10] and enhance the robustness against typical audio compression systems by incorporating a model of the HAS.

The scheme proposed in [MJM03] chooses a set of frequencies by comparing the FFT spectrum of the original signal with that of the corresponding compressed-decompressed signal. The watermark bits are embedded at those frequencies that have similar magnitudes in both spectrums. However, this choice disturbs the original signal at the most significant frequencies, which is not desirable as far as perceptual transparency is concerned.

The scheme proposed in [MJM05] introduces some randomness into the process of selecting the frequencies, which makes it possible to improve the transparency of the scheme at the price of losing some robustness. All these schemes are non-blind, that is, the spectrum of the original signal is needed to detect the embedded watermark bits.

The algorithm proposed in [FM09] embeds watermark bits based on the spline interpolation of the data derived from FFT transformation. The embedding process is depicted in Figure 2.3. As can be seen from the Figure, FFT analysis is applied to each frame (i.e. short segment) of the original signal to derive the magnitudes of the odd bins. Then the interpolated magnitudes of the even bins are derived by spline interpolation of

the magnitudes of the odd bins. The watermark bits are embedded by manipulating these spline-interpolated magnitudes of the even bins. Finally, the watermarked signal is reconstructed by inverse FFT.

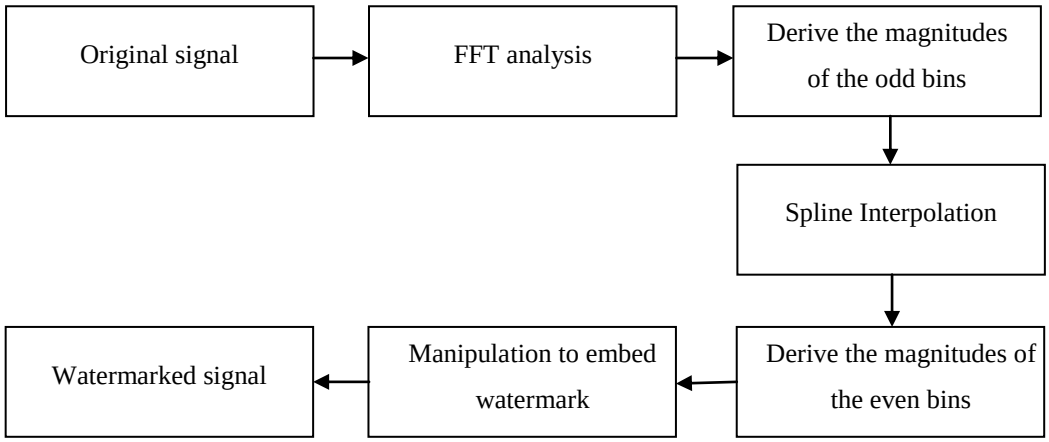


Figure 2.3 The embedding process of the algorithm proposed in [FM09]

The detection process can be depicted in Figure 2.4. As seen from the figure, FFT analysis is applied to the watermarked signal to derive the magnitudes of the odd bins and the even bins on a frame-by-frame basis. Then spline interpolation is used to derive the interpolated magnitudes of the even bins. These interpolated magnitudes of the even bins are compared with the FFT-derived magnitudes of the even bins to detect the watermark bits.

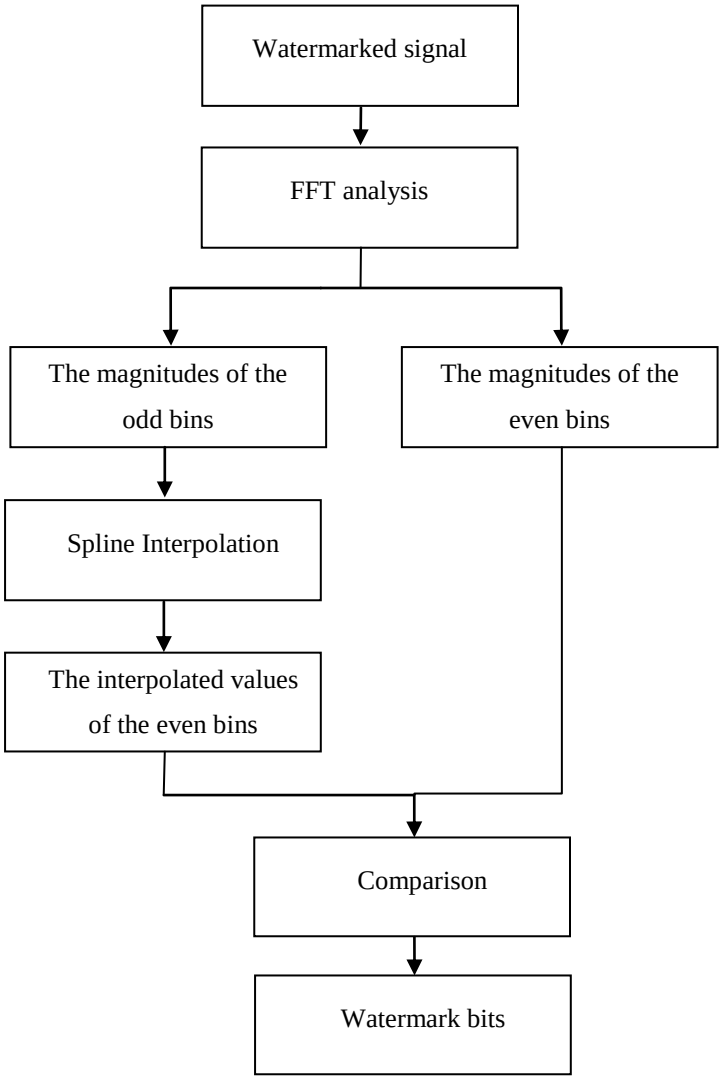


Figure 2.4 The detection process of the algorithm proposed in [FM09]

This algorithm achieved a high capacity about 3000 bps and it is robust against most attacks defined in the Strimark Benchmark for Audio v1.0 [XKH08]. The average ODG score achieved is -0.5, which is acceptable. The computational efficiency of this algorithm is high because only the interpolation, FFT, and inverse FFT processes are involved. The disadvantage of this algorithm is that the embedded watermark bits are

easily removed because the embedding position is known. Furthermore, because this algorithm is only based on a comparison of the magnitudes that are easily disturbed, thus the algorithm would be vulnerable against certain attacks [EB09]. Lastly, because the test was only based on five songs from the same album, the evaluation was limited.

In [DKK10], the original signal is segmented into non-overlapping frames. Watermarks are embedded by manipulating the magnitude of the highest prominent peak in the spectrum of each frame. The extraction process is an exact inverse of the watermark embedding process. The experiments showed that this algorithm was very robust and achieved a high SNR ranging from 20 dB to 28 dB [DKK10]. However, no perceptual evaluation was performed. In addition, the detection process is informed. The watermark information can be removed easily too because the embedding position is known.

In [MRF10], the watermark bits are embedded by disturbing the magnitudes of the spectrum of the original signal at some selected frequencies. The selected frequencies and the introduced disturbance are tuned carefully, aiming at achieving a good balance between the imperceptibility and the robustness.

Experiments were conducted to evaluate the performance of this algorithm [MRF10]. The robustness was not evaluated based on the conventional BER method, but on a particular method defined in [DML+06]. Thus, it is hard to know the robustness in terms of the conventional measure. However, based on the evaluation results given in [MRF10], it can be found that only few watermark bits have been correctly detected even

after using an error correction scheme [MRF10]. For example, only 60 out of 117 bits have been correctly detected after MP3 128 kbps, and only 44 out of 117 bits have been correctly detected after an additive noise attack. The imperceptibility achieved was an average ODG of -0.3 with the worst ODG score being -1.07 among 6 test music tracks. It is worth noting that the proposed algorithm included a relatively comprehensive analysis on how to tune the parameters to improve the performance, which would allow adaptation of this algorithm to different applications.

2.3.2.2 DWT based audio watermarking algorithms

The DWT decomposes a signal into an approximation signal and a detail signal by applying a lowpass filter and a highpass filter respectively. Both signals are then downsampled by a factor of two [Fil07]. The approximation signal can be subsequently divided into a new approximation and detail signals. This process is carried out iteratively producing a set of approximation signals at different detail levels (scales) and a final gross approximation of the signal. The maximum number of levels is determined by the length of the signal. More specifically, if the signal is composed of 2^D samples, the maximum levels would be *D*. By this recursive procedure, sets of coefficients are produced, known as the "approximation" coefficients and the "detail" coefficients [Fil07].

An exemplar DWT process is depicted in Figure 2.5, where the original input is a 16 samples signal and `o' means an actual system output [Edw92]. From the figure, it can be seen that this signal is decomposed into 4 different levels. For each level, a highpass

filtering output (“detail” coefficients) and a lowpass filtering output (“approximation” coefficients) are produced. The lowpass filtering output is processed again to generate the subsequent level of “approximation” and “detail” coefficients.

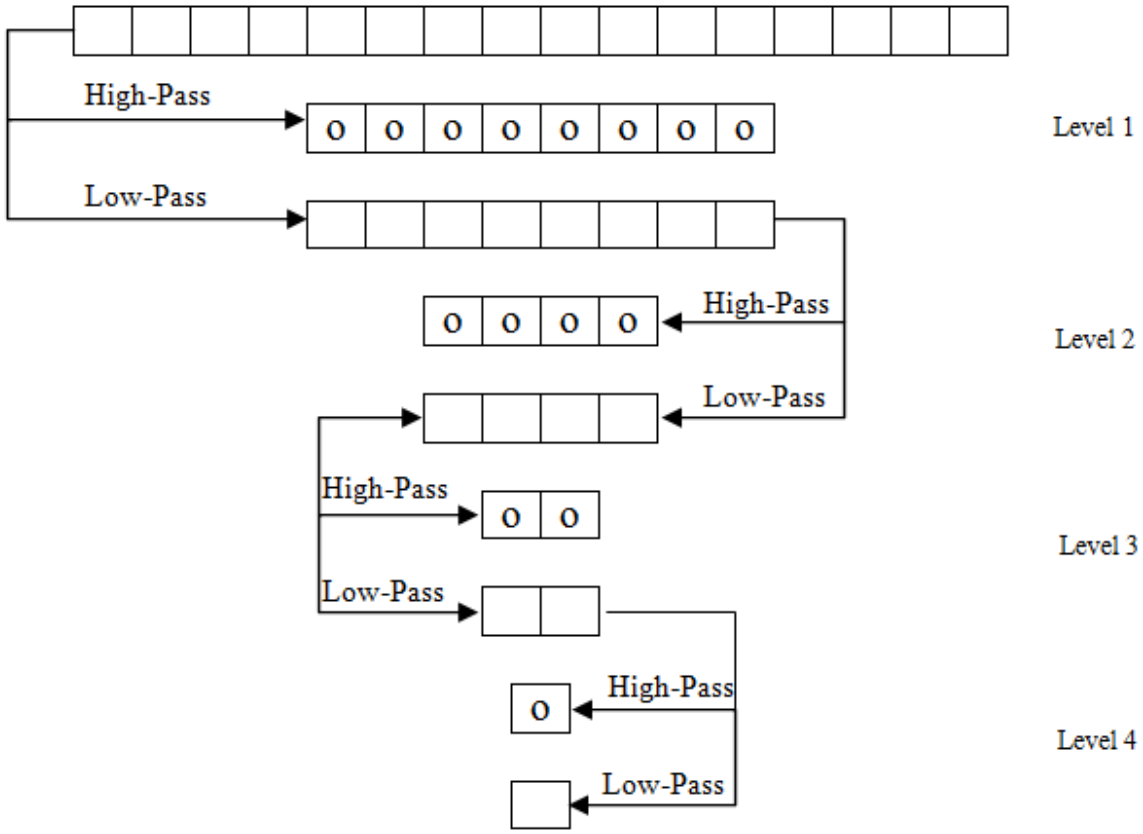


Figure 2.5 The DWT analysis of 16 samples data

Manipulation of the “approximation” coefficients to embed watermark bits is more robust. This is because these coefficients contain more significant components [KB11] that can survive different attacks. The “detail” coefficients are less favoured for embedding watermarks. The reason for this is that they contain less significant

components and are thus vulnerable to attacks. In addition, they have less energy so that perceptible distortion is more likely to be introduced.

There is a trade-off between the capacity and the robustness in determining which level of DWT coefficients should be manipulated. Manipulating a lower level of coefficients is conducive to achieving a higher capacity but a lower robustness. Manipulating a higher level of coefficients is conducive to achieving a higher robustness but a lower capacity [KB11]. In addition, a higher level DWT means a greater computational cost.

Some DWT based watermarking algorithms have been proposed in [FM10, HMB11, EB09, BSD10]. In [FM10], the main idea behind the proposed algorithm is to segment a long audio data sequence into many sections first, then embed watermark bits into the DWT low frequency coefficients [CW01].

Experiments were carried out on two music songs [FM10], and it was shown that the capacity is about 172 bps. The robustness against MP3 64 kbps was given a BER of 8.49% and against additive noise had a BER of 10%. The imperceptibility was only evaluated by the SNR, which gave a SNR of about 30 dB. Thus it is hard to know whether it is perceptually transparent or not. In addition, the test data was too small to give confidence in the result. Lastly, it was not suggested which level of DWT coefficients was used, which could have an impact on the robustness and the capacity as stated above.

In [HMB11], the watermark is embedded in the “detail” coefficients of the second level of the DWT to achieve an acceptable trade-off between the robustness and the imperceptibility. It is worth noting that, as the author mentioned, the two-level DWT gives a better result than a higher level DWT. The detection process is not blind. First, the second level “detail” coefficients of the watermarked audio are derived. Then, they are subtracted from the corresponding original coefficients to detect the watermark bits.

The performance of this algorithm was evaluated based on two different genres of music samples with a length of 11 seconds: pop and instrumental [HMB11]. The imperceptibility was assessed based on both the SNR and a subjective listening test. The SNR achieved was 28 dB and the subjective listening test gave a MOS score of 5.0. The robustness was assessed based on the correlation strength. The higher the correlation, the stronger the robustness. From the results, it was found that the correlation coefficient was high when different attacks were applied to the pop sample, the average value being about 0.90. However, the correlation coefficient was low when different attacks are applied to the instrumental sample. This was found to be extremely low for some attacks, such as the echo attack or zero cross attack [HMB11]. An evaluation against a MP3 attack at any bit rate was not given. The capacity was also not evaluated in this paper. Furthermore, the embedding position of this algorithm is known which makes the embedded watermark bits easily to be removed.

In [EB09], the watermark bits are embedded by manipulating the first level DWT approximation coefficients. The embedding procedure can be described as follows: a

pseudorandom sequence was employed to increase the security of the algorithm. If the watermark bit to be embedded is a ‘1’, the pseudorandom sequence is added to the first level approximation coefficients. If the watermark to be embedded is a ‘0’, it does not change anything, but generates a new pseudo random sequence with a new seed, which is to be used for the next watermark bit ‘1’. The detection procedure can be described as follows: the first level approximation coefficients of each frame are derived based on a DWT analysis. Then, they subtract the original first level approximation coefficients to produce a difference matrix. This matrix is correlated with all the generated pseudo random sequences. If all the correlation coefficients are lower than a predefined threshold, then the watermark bit is detected as a ‘0’. Otherwise, the watermark bit is detected as a ‘1’. Because the original coefficients and all pseudo random sequences are required at the detection stage, the watermarking algorithm is ‘informed’.

The performance of this algorithm was evaluated [EB09]. The SNR was approximately 30 dB for all the test samples. An informal subjective listening test was conducted and gave a MOS score of 5. These results suggested that this algorithm is extremely perceptually transparent. The BER varied depending on the type of attack: 0% for a filter attack, 15% for a re-quantization attack and 50% for a MP3 128 kbps attack. The robustness against MP3 is extremely weak because slight amplitude modifications introduced by the watermarking algorithm are distorted significantly following MP3 compression of the audio [EB09]. This algorithm is secure because of its dynamically

generated pseudo random sequences: without all the seeds, it is hard to detect the watermark.

In [BSD10], an adaptive audio watermarking algorithm based on the SVD and DWT was proposed. This is the first adaptive DWT SVD audio watermarking scheme but is influenced by one previously proposed for image watermarking [FGD01a]. The embedding process can be described as follows: the first level DWT approximation coefficients of each frame are derived and then organized as a two dimensional matrix. This matrix is then decomposed using the SVD and the coefficients in one of decomposed matrices are altered [DWW03], by which the watermark bits are embedded.

Experiments demonstrated that this algorithm achieved a good imperceptibility, with an SNR of about 24.37 dB and a MOS score of 4.46 [BSD10]. The proposed scheme is robust against attacks such as MP3 compression, lowpass filtering and additive noise [BSD10]. The capacity attained was 45.9 bps.

2.3.2.3 Spread Spectrum watermarking algorithms

Spread-Spectrum (SS) watermarking technology can trace its history back to SS communications, where a narrowband signal is modified so that its energy is spread over a much larger bandwidth. As a result, the signal energy present in any single frequency is almost undetectable. Likewise, in SS watermarking, the watermark energy is spread over many frequency bins so that the energy in any one bin is very small and is difficult to detect [CKL+97].

The general approach to a SS watermarking algorithm can be described as follows [KM03]: the data to be watermarked **x** represents a collection of samples from an appropriate invertible transformation on the original audio signal. It can be modelled as a random vector, where the elements are independent identically distributed Gaussian random variables with standard deviation δ_x, that is, $\mathbf{x} \sim N(0, \delta_x)$ [CKL+96, SZTB98, Sze79]. A watermark is defined as a direct SS sequence **w**, which is a pseudo-randomly generated vector ($\mathbf{w} \in \{\pm 1\}^N$) and is mutually independent with respect to **x**. Each element w_i is usually called a 'chip'. The watermarked data **y** is created as per $\mathbf{y} = \mathbf{x}+\delta\mathbf{w}$, where δ is the watermark amplitude. The watermark is detected by correlating **y** with **w** according to Equation (2.6):

$$C(\mathbf{y}, \mathbf{w}) = \mathbf{y}.\mathbf{w} = E[\mathbf{y}.\mathbf{w}] + N\left(0, {}^{\delta_x}\!/_{\sqrt{N}}\right) \tag{2.6}$$

The detector decides that a watermark is present if $C(\mathbf{y},\mathbf{w}) > Th$, where Th is the detection threshold.

The energy of watermark is very low as it is distributed across all the frequency bins. Thus, a strong interference might be introduced by an attack, which is the key deficiency of the SS watermarking algorithms. Some SS watermarking algorithms have been proposed in [CMB02, CKL+97, WPD99, SZTB98, WSK99, MF03, KM03]. In [CKL+97], it was recommended that the SS watermark be placed in the perceptually significant components of the signal spectrum, through which the robustness of the watermarking can be maximized. In addition, the perceptual masking phenomena of the

signal can be exploited when embedding watermarks, so that the energy of particular frequencies can be increased without degrading the imperceptibility but improving the robustness [CKL+97].

In [KM03], the vector **x** is composed of coefficients of a Modulated Complex Lapped Transform (MCLT) [Mal99]. The MCLT is an oversampled filter bank that provides a perfect reconstruction. Only the coefficients of the MCLT within 200 Hz – 2 kHz are watermarked. Sub-band selection aims at minimizing noise effects as well as its sensitivity to compression so that the robustness and imperceptibility can be improved. The embedding and detection process can be depicted as Figure 2.6.

The embedding procedure, as shown in Figure 2.6 (a), can be described as follows: a MCLT is firstly performed on each signal block to derive the MCLT coefficients, that is, **x**. Then, each watermark chip is spread across the audible sub-band in each block of **x**. Finally, a time domain watermarked signal is reconstructed by an inverse MCLT (iMCLT). The detection procedure, as shown in Figure 2.6 (b), can be described as follows: again a MCLT is firstly performed on each signal block to derive the MCLT coefficients, that is, **y**. Next, cepstrum filtering is applied on each block of **y** by which large fluctuations in **y** can be attenuated to reduce the fluctuation of the correlation. Finally, the presence of the watermark is detected based on the correlation.

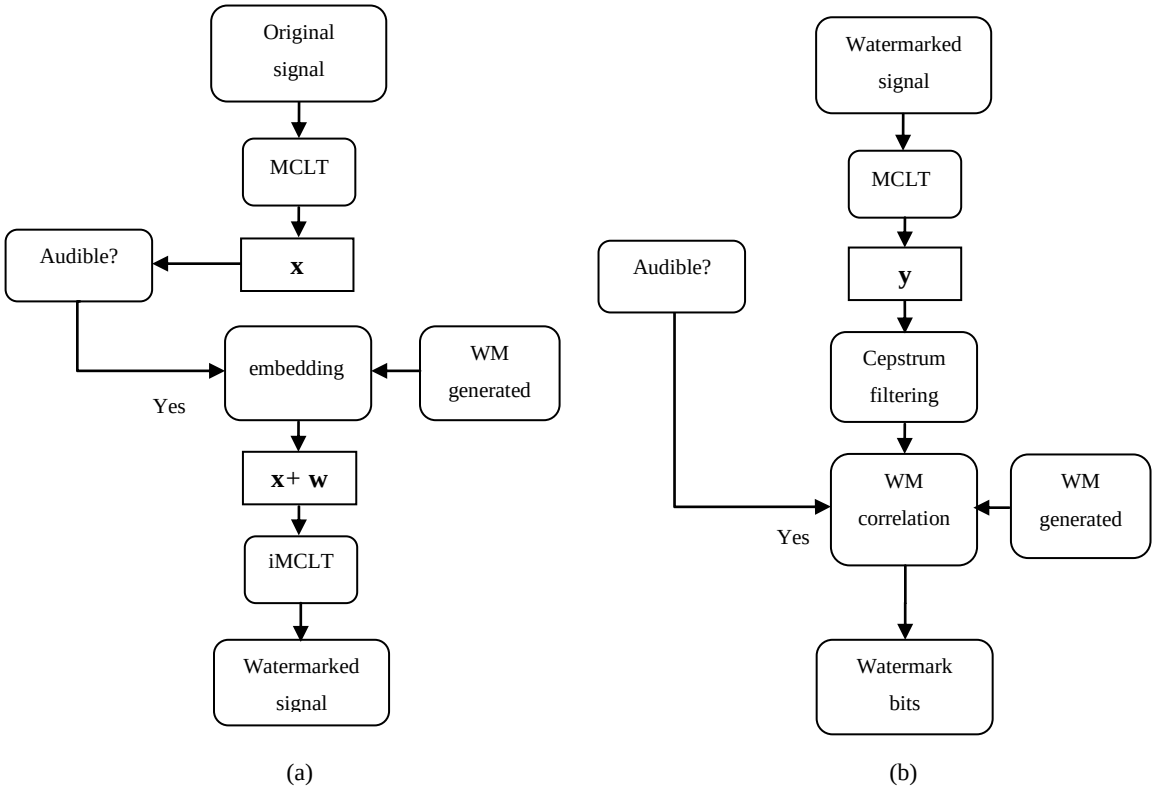


Figure 2.6 (a) The embedding procedure (b) The detection procedure

The robustness of this algorithm was evaluated by the attacks defined in Stirmark Audio [SLDP02], and all but one attack had a minimal effect on the correlation value. In other words, the algorithm is very robust. Imperceptibility and capacity have not been evaluated formally. Some additional processes were incorporated into the algorithm to improve the performance of the algorithm. These processes include block repetition coding to protect against de-synchronization attacks [AP98], cepstrum filtering, as mentioned already, to improve the robustness, and audible MCLT sub-band detection to improve the imperceptibility. By incorporating these additional processes, the trade-off between imperceptibility and robustness can be partly circumvented. However, the

algorithm proposed was computationally intensive because of adding these processes, as mentioned in [KM03].

A frequency selective based SS watermarking scheme was proposed in [MAK08]. This scheme uses only a fraction of the audible frequency range for embedding. It uses a secret key to select sub-bands for embedding. The sub-bands' frequency masking characteristics, based on a HAS model, are utilized to achieve a good balance between imperceptibility and robustness. The robustness of this algorithm was evaluated and it was shown that this algorithm was robust against lowpass filtering, MP3 128 kbps and additive noise [MAK08]. However, the capacity and imperceptibility were not evaluated. In addition, this algorithm was computationally intensive because many processes were incorporated, including the HAS model [LL11].

2.3.3 Hybrid algorithms

Hybrid algorithms refer to those novel algorithms, such as one based on Chirp Coding [Bla07], one termed "patchwork" [KAAA09], and one based on the SVD [XKH08], which cannot simply fall into either of the two categories mentioned above. A particular reason for defining this category is to highlight their novelty. These algorithms will be reviewed in detail in the following sections.

2.3.3.1 Chirp Coding watermarking algorithms

In [Bla07], a fragile watermarking algorithm was proposed. The embedding process can be described as follows: a 7 levels of wavelet decomposition of the signal is performed first to produce 7 levels of "detail" coefficients. In order to measure the global effect of

introducing the watermark into the signal, the “approximation” coefficients at the 7^{th} level are used as well. Thus, 8 decomposition vectors in total are generated. The reason of using the “detail” coefficients is because they are very sensitive to attacks such as lossy compression and audio slicing. The percentage of the energy of each vector occupying the total energy of the 8 vectors is then calculated. These percentages are rounded to the nearest integers and converted into a binary stream, which is to be used as a watermark bit sequence. Subsequently, a Chirp function is created. This function is then multiplied with a new signal, which is created based on the binary sequence, and scaled by a predefined scale factor to produce the chirp code. This chirp code is then added to the original signal to generate the watermarked signal. In order to make the watermark inaudible, the chirp code generated has a very low frequency and amplitude.

At the detection side, the same chirp function used at the embedding side is applied and correlated with the watermarked signal. By this means, the watermark bits can be recovered. Then, whether the signal has been tampered with or not can be validated by comparing the recovered watermark bit sequence with the possibly tampered binary stream, which can be generated from the watermarked signal directly.

As for this algorithm, the embedded watermarks are difficult to remove from the host since the initial and the final frequency of the chirp function are at the discretion of the user and its position in the data stream can be varied through application of an offset, all such parameters being combined to form a private key [Bla07]. The listening tests showed that there was no perceptual difference between the original signal and the

watermarked signal. The reconstruction of the chirp code is uniquely robust in the case of a very low SNR, thus it is easy to adapt this algorithm as a robust watermarking algorithm.

2.3.3.2 Quantization based watermarking algorithms

Among all of the blind watermarking schemes, the quantization based scheme is one of the simplest. In [CW01], a QIM function is defined as Equation (2.7):

$$s(\mathbf{x}, i) = q_{\Delta}(\mathbf{x} + d[i]) - d[i] \tag{2.7}$$

where q_{Δ} (.) is a uniform scalar quantizer with step size Δ, $\mathbf{x}$ is the vector to be quantized, i is the watermark bit number, and $d[i]$ is the dither vector defined as follows:

$$d[i,1] = \begin{cases} d[i,0] + \dfrac{\Delta}{2}, & d[i,0] < 0 \\ d[i,0] - \dfrac{\Delta}{2}, & d[i,0] \geq 0 \end{cases} \qquad i = 1,2,\ldots..N \tag{2.8}$$

where N is the number of watermark bits, and $d[i,1]$, $d[i,0]$ are the vectors for embedding a watermark bit '1' and '0' respectively. In general, quantization is applied on the coefficients derived from a signal transformation such as the FFT or DWT [KAK07, SGD11].

The watermark embedding process proposed in [KAK07] can be described as Figure 2.7. From Figure 2.7, it can be seen that the original signal is transformed by the DWT first. The step size Δ is then adaptively calculated based on the mean value of the DWT coefficients, followed by the dither vector calculation. The mean value of the DWT

coefficients is modulated based on Equation (2.7). Finally, the inverse DWT is applied to reconstruct the time domain watermarked signal.

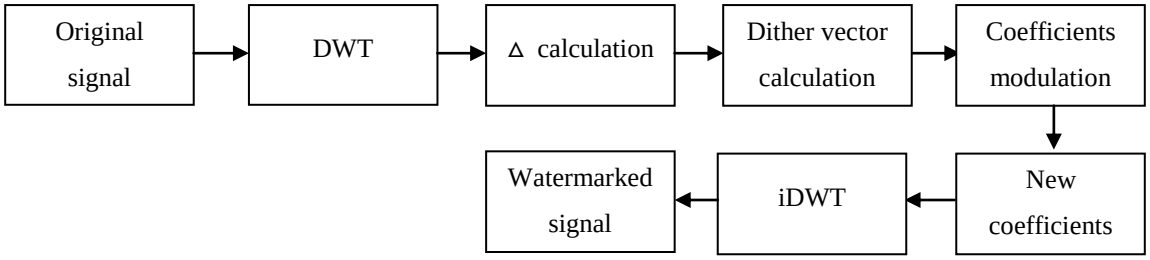


Figure 2.7 The watermarking embedding process

The performance of this algorithm was evaluated [KAK07]. The robustness against MP3 64 kbps was a BER of about 2%, against resampling gave a BER of about 0.3%, against echo had a BER of about 30% and against highpass filtering was a BER of about 21%. Thus, it can be seen that the algorithm cannot survive filtering and echo attacks. The capacity was about 86 bps and the imperceptibility was not evaluated.

2.3.3.3 Patchwork watermarking algorithms

The patchwork approach was first proposed by Bender for image watermarking [BGML96]. The idea is to select two patches (that is, data sets) randomly from the host signal and the mean values of these two patches are manipulated by a constant value, which defines the watermark strength. More specifically, two patches *A* and *B* are randomly selected, each element in *A* and *B* is altered according to Equation (2.9):

$$a_i^* = a_i + d, \quad b_i^* = b_i - d \tag{2.9}$$

where a_i^* and b_i^* are manipulated sample values in the patches *A* and *B* respectively, a_i and b_i are original sample values in the patches *A* and *B* respectively, and d is a constant value.

The detection process starts with the subtraction of the sample values between the two patches. The expected value of the differences of the two sample means $\overline{a^*}$ and $\overline{b^*}$, $E\left[\overline{a^*} - \overline{b^*}\right]$, is used to decide whether the patch contains watermark information or not. Since two patches are used rather than one, it can detect the embedded watermarks without the original host. Thus, it is a blind watermarking algorithm. Patchwork itself is a very good algorithm, but it has some inherent drawbacks. Note that,

$$E\left[\overline{a^*} - \overline{b^*}\right] = E\left[(\bar{a} + d) - (\bar{b} - d)\right] = E\left[\bar{a} - \bar{b}\right] + 2d \tag{2.10}$$

where $\bar{a}$, $\bar{b}$ are sample means of the original two patches. The patchwork assumes that $E\left[\overline{a^*} - \overline{b^*}\right] = 2d$ due to the prior assumption that random patch has the same expected values such that $E\left[\bar{a} - \bar{b}\right] = 0$. However, the actual difference of the original sample means is not always zero in practice [KAAA09, YK03].

An audio patchwork watermarking was first proposed in [Arn00]. Several attempts have been made to improve this algorithm. For example, the Modified Patchwork Algorithm (MPA) [YK03] and the Generalized Patchwork Algorithm (GPA) [YK03+]. As far as MPA was concerned, four patches are selected instead of the two patches originally. Two of these four patches are used for embedding a watermarking bit '1' and the other two are used for embedding a watermarking bit '0'. A watermarking bit

is embedded by manipulating the means of its associated patches in opposite directions. GPA was proposed to generalize MPA by employing both additive and multiplicative embedding rules. However, all these methods suffer from a prior assumption of the equality of the statistical behaviour of sets, that is, the conformance of the data in the patches to a particular statistical distribution.

In [KAAA09], a Multiplicative Patchwork Method (MPM) approach for audio watermarking was proposed. In order to embed the watermark information, two data sets of the host are chosen. A single bit of the watermark is embedded by multiplying or dividing the samples of one set and leaving the other set unchanged. The watermark bits are detected by comparing the energy of the two sets with a defined threshold. This method is implemented in the wavelet domain and sets are selected from the approximation coefficients. In order to control the inaudibility of watermark, the PEAQ algorithm is incorporated into the model to evaluate the perceptual quality of the watermarked audio following each iteration, based on which the watermark strength is adaptively tuned to get a better imperceptibility.

Simulation results showed that MPM outperformed all the previous patchwork methods [KAAA09]. The perceptual transparency achieved was an ODG score of -0.6. This value actually was used as the criterion for the watermark strength adjustment. The capacity achieved was 13 bps and was better than that of 10 bps in both [Arn00] and [YK03]. The robustness against attacks was evaluated and it was concluded that this algorithm was very robust against common attacks such as filtering, resampling and MP3

[KAAA09]. However, this algorithm was vulnerable to phase changes, for example, it was not robust to attacks such as pitch shifting. The computational efficiency was very low because the PEAQ had to be executed following each iteration. Furthermore, the test set only contains three music tracks with different genres, which was too small to achieve a reliable result.

2.3.3.4 Interpolation based watermarking algorithms

Interpolation is a technique for constructing new data points within the range of a set of discrete data [FM09, Wei79, FLK06]. Polynomial interpolation is one of the best-known interpolation approaches [FM09]. Its advantages consist of its simplicity of implementation and the good quality of the interpolant obtained from it. Spline interpolation is a particular type of polynomial interpolation [FM09]. In [DP09], a spline interpolation based watermarking scheme was proposed. The embedding process is shown as Figure 2.8.

From Figure 2.8, it can be seen that the time domain signal is divided into frames and each frame is further divided into four groups called ζ_1, φ_1, φ_0, ζ_0. To embed a watermark bit '0', the samples in φ_0 are replaced by the values interpolated from ζ_0, while the samples in φ_1 are unchanged. To embed a watermark bit '1', the samples in φ_1 are replaced by the values interpolated from ζ_1, while the samples in φ_0 are unchanged. Likewise, In either case, the samples in ζ_1 and ζ_0 are left unchanged. The watermarked signal is then reconstructed.

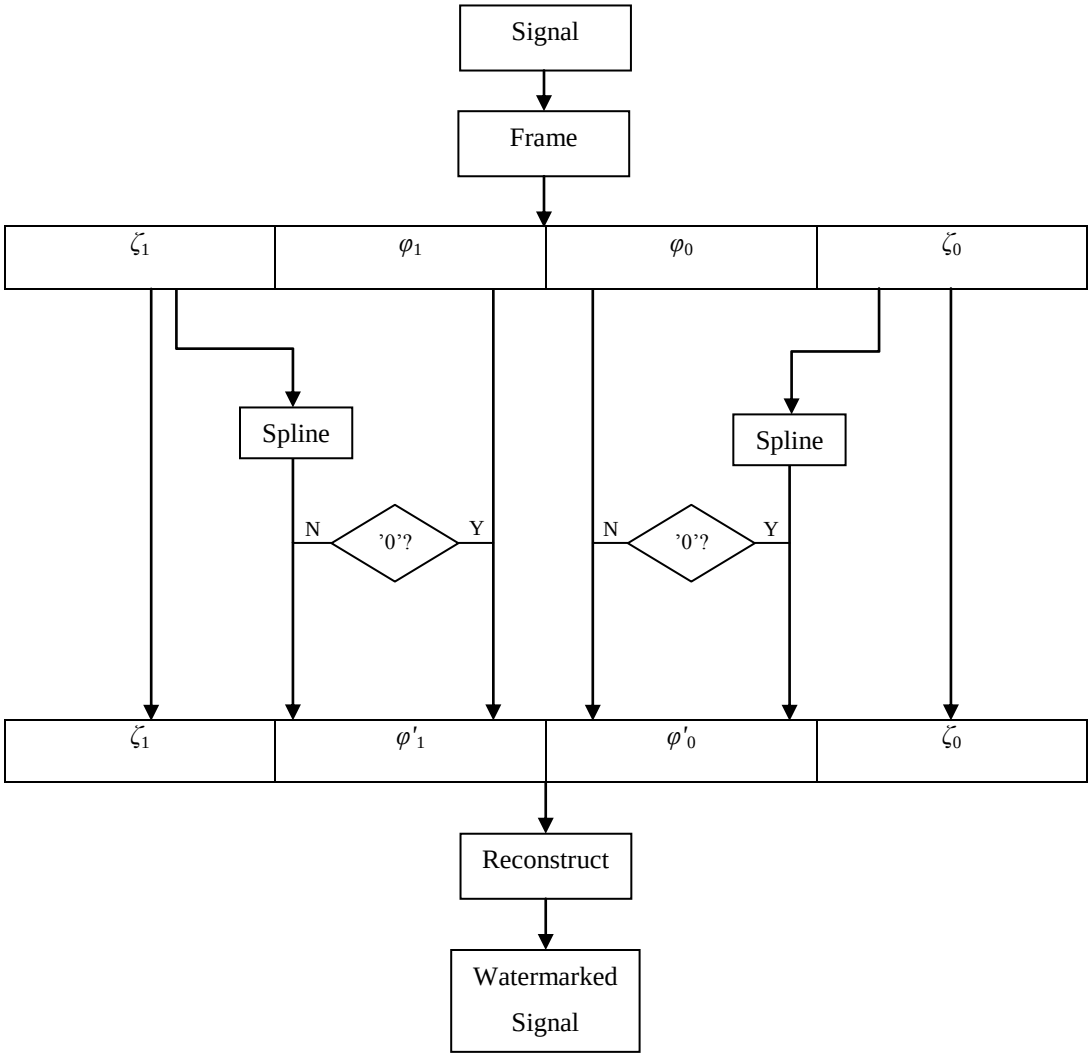


Figure 2.8 The embedding process of the algorithm proposed in [DP09]

The detection process is depicted as Figure 2.9. From Figure 2.9, it can be seen that each frame of the watermarked signal is divided into four groups called ζ_1, φ'_1, φ'_0, ζ_0. ρ^2_1 is calculated based on the mean squared error of the values from φ'_1 and the corresponding interpolated values from ζ_1. Similarly, ρ^2_0 is calculated based on the mean squared error of the values from φ'_0 and the corresponding interpolated values from ζ_0. If

$\rho^2_1 < \rho^2_0$, the watermark bit to be detected is a '1', otherwise the watermark bit to be detected is a '0'. The reason for this is that if the watermark bit embedded is a '1', ρ^2_1 should be equal to 0 and ρ^2_0 should be greater than 0 according to the embedding method.

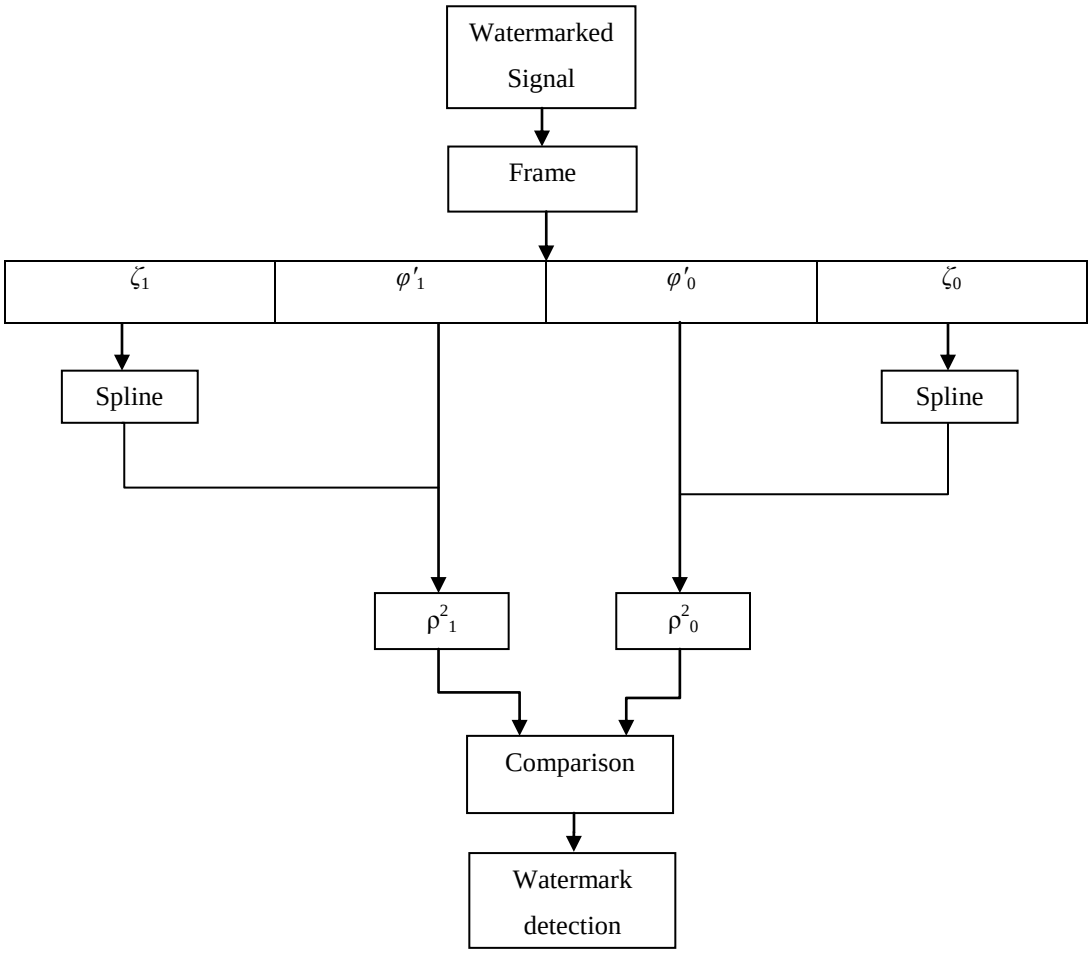


Figure 2.9 The detection process of the algorithm proposed in [DP09]

Earlier work [FIK06, MCL08] proposed the replacement of all the samples in set φ_1 and φ_0 with the interpolated values if the watermark bit is a '1'. On the other hand, if the watermark bit is a '0', the samples in both sets are not modified. At the detection stage, the mean squared error is calculated over the set φ_1 and φ_0. This process would

yield a zero mean squared error when the watermark bit is a ‘1’ and a finite non-zero error when the watermark bit is a ‘0’, in the absence of an attack. However, when attacks are applied to the signal, the detection result will be affected. This is because the distortion introduced by attacks can interfere with the calculation of the mean squared error. The algorithm proposed in [DP09] overcomes this drawback by proposing the substitution of values in one set from either φ_1 or φ_0 depending on the watermark bit. Because both sets are equally exposed to attacks, both mean square errors will be changed equally in a statistical sense.

The performance of this algorithm was evaluated [DP09]. An ABX subjective listening test was performed to assess the imperceptibility. It was found that as the frame size was increased, the perceptual distortion was increased. For example, when the frame size was 12 samples, it was imperceptible, while when the frame size was 18 samples, it was extremely perceptible. The robustness against different bit rates of MP3 compression was also evaluated. It was found that the robustness became stronger as the frame size was increased. For example, when the frame size was 14, the robustness against MP3 80 kbps was shown by a BER of 30%, when the frame size was 20, the robustness against MP3 80 kbps was shown by a BER of 20%. When repetition was used to enhance the robustness, the robustness increased dramatically. For example, when the number of repetitions was five, the robustness against MP3 80 kbps was shown by a BER of 15% when the frame size was 14. While when the number of repetitions was 11, the robustness against MP3 80 kbps gave a BER of 6% with the same frame size.

However, the proposed algorithm was not tested against other attacks such as filtering or noise. The capacity of this algorithm was high, around 3150 bps when the frame size was 14. This high capacity facilitates the improvement of robustness by utilizing the repetition approach. However, this proposed algorithm did not perform well if the signal contains many high frequency components [DP09]. Furthermore, only one audio clip selected from the Sound Quality Assessment Material (SQAM) database was used for the evaluation, which means its performance assessment was very limited.

2.3.3.5 SVD based watermarking algorithms

The SVD is a well-known numerical analysis tool used on matrices. If *A* is an arbitrary *m*-by-*n* matrix, with the full SVD, it can be decomposed as Equation (2.11):

$$A = USV^T \tag{2.11}$$

where *U* is a *m*-by-*m* unitary matrix and *V* is a *n*-by-*n* unitary matrix, called the left eigenvector and the right eigenvector respectively. *U* and *V* are both orthogonal. *S* is a *m*-by-*n* diagonal matrix. These elements in *S* are called singular values, which are the square root of the eigenvalues [LNK06]. The ordering of the eigenvector is determined by high-to-low sorting of singular values, with the highest singular value in the upper left index of the matrix *S*.

Recently, the SVD has been used extensively as an effective tool in digital watermarking [AA10, BSD10, CYH+07, FL08, CTL05, GZE03, Cha02, SSY02, OSM05, ZL05, LNK06, KBWN06, KBWN06a , HC07, TB97, MAS08, LT02, ZL05]. Most existing SVD based watermarking algorithms have been applied to images

[CYH+07, FL08, CTL05, GZE03, Cha02, SSY02, OSM05, ZL05, LNK06, KBWN06, KBWN06a, HC07, TB97, MAS08, LT02, ZL05]. SVD based audio watermarking algorithms also exist but there are only a limited number, some of which can be found in [AA10, BSD10]. All these SVD based watermarking algorithms can be grouped into two categories.

The first category of SVD based watermarking algorithms are 'informed' [MAS08, LT02, ZL05], requiring access to the original signal or the watermark in order to successfully detect the embedded watermark. The scheme proposed in [MAS08] can be seen as a typical example to illustrate the idea used for this group. At the embedding stage, a cover image A is decomposed into three matrices: U, S and V by the SVD. The watermark information W is then linearly added to S, resulting in a new matrix S'. Then an inverse SVD is applied to S', U and V to produce a watermarked image A_w. In the detection stage, U^T and V are multiplied with A_w to get S'. Then, the watermark W can be recovered by a linear subtraction between S and S'.

The second category of SVD based watermarking algorithms are 'blind' and the watermark information is embedded by manipulating the elements in the SVD decomposed matrices, such as U or S. These schemes are based on some findings proposed in [AA10, BSD10, CYH+07, FL08, CTL05].

Particularly, an image watermarking algorithm presented in [CYH+07] is based on two findings. Firstly, column-wise modification of the elements in the U matrix will cause less visible distortion than modifying these elements row-wise. Secondly, row-wise

modification of the elements in the V matrix will cause less visible distortion than modifying these elements column-wise.

Another SVD based digital image watermarking algorithm proposed in [FL08], was developed based on the observation that the elements in the first column of U and V can be modified for a robust SVD based watermarking.

In [CTL05], an image watermarking algorithm that was based on tuning the coefficients of U was proposed. In [AA10, BSD10], the proposed audio watermarking algorithm is based on the observation that changing S slightly does not affect the audio quality significantly and that the singular values in S are consistent after common signal processing. However, there are some drawbacks of manipulating S [AA10, BSD10]. First, the modification of the largest coefficients in S makes the embedded watermark easy to be removed, as the embedding position is noticeable [AP76]. Second, embedding watermark information based on the modification of the less significant coefficients in S risks that the modification is distorted after signal processing operations [AP76, RMK07]. Finally, only a small number of singular elements in S are available for manipulation, which makes it hard to achieve a high capacity [CTL05].

As far as the performance of the SVD based audio watermarking algorithm is concerned, a non-blind algorithm proposed in [OSM05] can be used as one typical example to illustrate. This algorithm embeds the watermark bits by manipulating the coefficients in the matrix S, to generate a new matrix S_w. Then, S_w is decomposed by the SVD as U_w, S'_w, V_w. U_w and V_w are retained for use in the detection stage. A MOS of 4.7

was achieved and the robustness against attacks was very strong. However, the robustness against MP3 was not given and the capacity was not evaluated [OSM05].

The non-blind audio watermarking algorithm proposed in [AA10] can be used as another example to illustrate the performance of the SVD based audio watermarking algorithm. The watermark information is embedded by manipulating the first element in the matrix *S*. The watermark information is detected by calculating the difference between the value of the first element in the original matrix *S* and that in the manipulated matrix *S*. A MOS of 5.0 was achieved for pop music signals, and a MOS of 4.3 was achieved for speech signals. This algorithm was strong against most attacks, but it was vulnerable for some attacks such as audio slicing or echo attack. However, the capacity was not evaluated, and only one music sample was used for test so that the evaluation was limited [AA10].

2.4 Evaluation

Up to now, a variety of popular algorithms developed for audio watermarking have been reviewed. In order to compare across all these algorithms, imperceptibility is assessed uniformly by the MOS score, and the ODG score can be mapped to the MOS score directly as shown in Table 2.3.

Table 2.3 The mapping between MOS and ODG

ODG	MOS
0	5
-1	4
-2	3
-3	2
-4	1

Table 2.4 and Table 2.5 are given to summarize the performance of the typical algorithms that were reviewed. Each algorithm listed has achieved a relatively better performance within its own group. The Chirp coding based audio watermarking algorithm is not listed in the tables [Bla07], as it was originally developed as a fragile watermarking. However, it has a great potential to be developed as a robust watermarking algorithm. In Table 2.4, four main characteristics of each watermarking algorithm are listed: imperceptibility assessed by MOS, robustness, capacity and computational efficiency. In Table 2.5, some other characteristics of the algorithms are listed:

1. Blindness
2. Additional processes: refers to those processes that can improve the performance of the algorithm, but can partially circumvent the trade-off as mentioned in Chapter 1. '✓' and '×' are used to indicate whether the additional processes were incorporated or not.

3. Parameters analysis: refers to the analysis on parameters that affect the performance. '✓', '×' are used to indicate whether this analysis was done or not.
4. Removability: refers to whether the watermark is easy to remove or not.
5. Reliability: refers to whether the performance given could be said to be reliable or not. This really depends on whether the test data was sufficiently large, and was taken across a variety of genres.

Table 2.4 The four main characteristics of each typical audio watermarking algorithm

Algorithm	Imperceptibility	Robustness	Capacity	Efficiency
LSB [CS05]	5.0	low	44100	high
Echo hiding [KNS05]	5.0	low	n/a	n/a
FFT [FM09]	4.5	high	3000	high
DWT [BSD10]	4.6	high	47	low
SS [KM03]	n/a	high	n/a	low
QIM [KAK07]	n/a	average	86	n/a
Patchwork [KAAA09]	4.4	high	13	low
Interpolation [DP09]	5.0	average	3000	n/a
SVD [OSM05]	4.7	high	n/a	n/a

Table 2.5 The other characteristics of each typical audio watermarking algorithm

Algorithm	Blindness	Additional Process	Parameters Analysis	Removability	Reliability
LSB [CS05]	✓	×	✓	easy	✓
Echo hiding [KNS05]	×	×	✓	hard	×
FFT [FM09]	✓	×	✓	easy	×
DWT [BSD10]	✓	×	×	hard	×
SS [KM03]	×	✓	✓	hard	✓
QIM [KAK07]	✓	×	✓	hard	×
Patchwork [KAAA09]	×	✓	✓	hard	×
Interpolation [DP09]	✓	×	✓	easy	×
SVD [OSM05]	×	×	×	hard	✓

From Table 2.4 and Table 2.5, it can be seen that different algorithms have different strengths and weaknesses. In general, the FFT based watermarking algorithm proposed in [FM09], the DWT based watermarking algorithm proposed in [BSD10], and the SVD based watermarking algorithm proposed in [OSM05] achieved a better overall performance compared to the others.

The computational efficiency of the algorithm proposed in [FM09] is very high which makes it very attractive for time-critical applications, and it has an extremely high capacity. However, the imperceptibility of [FM09] is not too satisfactory. The reason for

this could be that the coefficients estimation based on the FFT is not sufficiently precise, and thus the manipulation based on these estimated FFT coefficients introduces distortions. Therefore, a more precise coefficients estimation algorithm might help. Another issue with [FM09] is that its embedding rule is based on the comparison of the components' magnitudes, which makes it vulnerable to certain attacks that could distort the magnitudes. Therefore, a different embedding rule that is not based on the comparison of magnitudes would be desirable. Finally yet importantly, the embedding positions of the watermark in [FM09] are publicly known for the attacker, which makes the embedded watermark bits easier to remove. This should be improved.

As far as the DWT based audio watermarking algorithms are concerned, the output of the DWT is not as easily interpreted as that of FFT when applied to audio. Thus, it is difficult to have an intuition as to the impact of manipulation of DWT coefficients on the perceptual transparency, but it is easier to tailor the manipulation of the FFT coefficients to render it more perceptually transparent. Furthermore, there is no universal consensus on the type of wavelet or the level of decomposition in DWT analysis, and so many variants exist. Thus, developing new watermarking algorithms using the DWT is not pursued in this thesis.

A point worth noting is that the SVD is a powerful tool for audio watermarking. The reason for this is that the elements in the decomposed matrices of SVD are intrinsically robust, which makes the robustness of watermarking algorithms based on the SVD easier to achieve. Furthermore, the number of elements in the decomposed matrices

of the SVD is large, thus, there is a greater likelihood that SVD based audio watermarking algorithms can achieve a high capacity. As an example, it has been employed in [BSD10] and resulted in a very acceptable performance.

However, no SVD based audio watermarking scheme, which embeds watermark bits through manipulating the matrix *U*, has been proposed yet. However, it is worth investigating if it is possible to embed watermark bits based on manipulating the matrix *U*.

In addition, from Table 2.5, it can be seen that the criteria given in Section 2.1 have not been met sufficiently by these algorithms such as FFT based algorithm proposed in [FM09], DWT based algorithm proposed in [BSD10], and SVD based algorithm proposed in [OSM05]. This motivates further efforts to find ways of improving their performance.

2.5 Summary

In this chapter, a thorough literature review has been carried out. Based on all these reviews and combined with the problem set out in Chapter 1, it has been decided that the FFT based and the SVD based audio watermarking algorithms should be further investigated and improved. The work will be carried out in the following sequence:

1. A more precise frequency estimation approach should be designed or found first. This would help in reducing the perceptual distortion which is introduced when manipulating the frequency components to embed the watermark. In addition, this approach should be efficient.

2. A watermarking algorithm based on this more precise frequency estimation approach should be developed. This watermarking algorithm should meet the criteria set out in Section 2.1. Furthermore, an approach of generating embedding position dynamically should be designed, to make the embedded watermark bits less easily removed.
3. An investigation on the possibility of embedding watermark bits based on manipulating the matrix *U*, which is derived from SVD decomposition, should be carried out.
4. If the watermark bits can be embedded by manipulating the matrix *U*, then a new SVD based audio watermarking algorithm should also be developed and should satisfy the criteria set in Section 2.1.

Chapter 3 Complex Spectral Phase Evolution based audio watermarking algorithm

3.1 Introduction

In this chapter, an audio watermarking algorithm based on the CSPE, which is a super-resolution spectral analysis tool, is described. This watermarking algorithm is based on a frequency domain transformation of the signal. First, an improvement on the original CSPE is explained. Then a watermarking algorithm based on this improved CSPE is proposed and experiments to demonstrate the effectiveness of this watermarking algorithm on synthetic signals are performed. The proposed algorithm is then applied to real audio signals and problems associated with this are identified and resolved. The chapter closes with a perceptual transparency and robustness evaluation of the improved watermarking algorithm on real audio signals.

3.2 The Complex Spectral Phase Evolution

The estimation of the frequencies of a signal that is composed of sinusoidal components is often done in the frequency domain using peak-picking from the magnitude spectrum of the signal [KSS06], where the spectrum is almost always computed using the well-known FFT algorithm [SS10]. However, the accuracy of this algorithm is severely limited to cases where a component frequency is not a multiple of the windowed signal length divided by the sampling frequency [Cri89]. In essence, this means that only when a component frequency is aligned exactly with the analysis frequencies of the FFT can it

be measured accurately. Given the large number of frequency components that contribute to realistic sounds, this is a severe limitation when attempting to accurately analyze a signal. When a frequency component does not align exactly with the analysis frequencies of the FFT, a common solution that is used in sinusoidal modelling algorithms is to apply quadratic interpolation to the component spectral magnitudes immediately either side of the true frequency to find the correct frequency and magnitude values [AS04]**.** However, the performance of this method is highly dependent on the window function used and the length of the data for analysis [KM02]. The CSPE, as an alternative, will be detailed in following sections.

3.3 The procedure of CSPE

The CSPE was introduced by [GS06] as a method to accurately estimate the frequency of components that exist within a signal. It was also designed to be computationally efficient. It is actually related in some aspects to the cross-spectrogram technique of [Nel01] and the reassigned spectrum technique of [FH02].

The procedure of the CSPE can be depicted in Figure 3.1, where $x_0(n)$ is the original signal under analysis, and $x_1(n)$ is a one-sample shifted version of $x_0(n)$. As can be seen from Figure 3.1, the CSPE works as follows: an FFT analysis is performed twice. First on the signal $x_0(n)$ and a second time upon $x_1(n)$. Then, by multiplying the FFT spectrum of $x_0(n)$ with the complex conjugate FFT spectrum of $x_1(n)$, a frequency dependent function is formed from which the exact values of the frequency components it contains can be detected by extracting the angle information [GS06]. When graphed, this

frequency dependent function has a staircase-like appearance where the flat parts of the graph indicate the exact frequencies of the components.

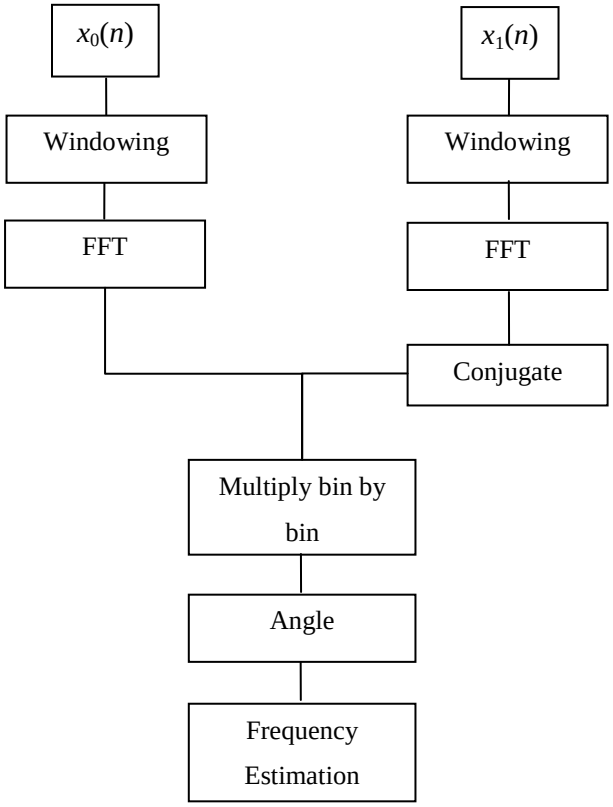


Figure 3.1 The procedure of CSPE

3.4 The mathematical deduction of CSPE

Mathematically, the algorithm can be described as follows: consider a real sinusoidal signal, with a frequency in Hz of $\beta = q + \delta$ where q is an integer and δ is a fractional number, with an amplitude of a and with an initial phase of b. w is the window function used in the FFT, N is the window length, F_{wx_0} is the windowed Fourier transform of $x_0(n)$, and F_{wx_1} is the windowed Fourier transform of $x_1(n)$. Then, by first writing

$$D = e^{\frac{j2\pi\beta}{N}} \tag{3.1}$$

The frequency dependent CSPE function can subsequently be written as [GS06]

$$CSPE_{w} = F_{wx_0} F^{*}_{wx_1} = (\frac{a}{2})^2 \begin{bmatrix} D^{*} \left\| F_w(D^n) \right\|^2 \\ + 2\,\mathrm{Re}\left\{ e^{j2b} DF_w(D^n) \odot F_w^{*}(D^{-n}) \right\} \\ + D \left\| F_w(D^{-n}) \right\|^2 \end{bmatrix} \tag{3.2}$$

where $\odot$ denotes the product on an element-by-element basis and * denotes the conjugation operation. The first and third terms in the square bracket are a positive frequency complex sinusoid and a negative frequency complex sinusoid respectively. The middle term represents the self-interaction terms from interference between the positive frequency component and the negative frequency component [GS06].

A windowed Fourier transform requires multiplication of the signal with the analysis window in time domain and thus the resulting transform is the convolution of the transform of the window function, *W*, with the transform of a complex sinusoid. Since the transform of a sinusoid is nothing but a pair of delta functions in the positive and negative frequency positions, the result of the convolution is merely a frequency-translated copy of *W* centred at $+\beta$ and $-\beta$. Consequently, with a standard windowing function, only when $k \approx \beta$ where k is the frequency bin number, the term $\| F_w(D^n) \|^2$ is considerable, and it decays rapidly when k is far from β. Therefore, the analysis window must be chosen carefully so that it decays rapidly to minimize any spectral leakage into adjacent bins. If this is so, it will render the interference terms, that is, the middle term, to be negligible in Equation (3.2). Thus, the frequency dependent CSPE for positive frequencies is:

$$CSPE_{w} \approx \frac{a^2}{4}\left\|F_w(D^n)\right\|^2 D^{-1} \tag{3.3}$$

Finding the angle information of Equation (3.3) leads to the frequency estimate

$$\begin{aligned} f &= \frac{-N\angle(CSPE_{w})}{2\pi} = \frac{-N\angle\left(\frac{a^2}{4}\left\|F_w(D^n)\right\|^2 D^{-1}\right)}{2\pi} \\ &= \frac{-N\angle(\frac{a^2}{4}\left\|F_w(D^n)\right\|^2 e^{-j\frac{2\pi}{N}\beta})}{2\pi} = \frac{-N(-\frac{2\pi}{N}\beta)}{2\pi} = \beta \end{aligned} \tag{3.4}$$

Thus, the non-integer frequency value f is obtained using Equation (3.4). The estimation of phase ϕ and magnitude m can be derived by Equation (3.5) and (3.6) respectively.

$$\phi = \angle \frac{2F_{wx_0}(k)}{W_{(k-f)}} \tag{3.5}$$

$$m = \left\|\frac{2F_{wx_0}(k)}{W_{(k-f)}}\right\| \tag{3.6}$$

The estimation of frequency, phase and magnitude by the CSPE, as shown above, can also be applied to signals containing more than one frequency component.

3.5 Performance demonstration of CSPE

An example of the output of the CSPE is shown in Figure 3.2, where the x-axis denotes the bin number and the y-axis denotes the estimated frequency value. Consider an exemplar signal $x_a(n)$ that contains components with frequency values of 17 Hz, 293.5 Hz, 313.9 Hz, 204.6 Hz, 153.7 Hz, 378 Hz and 423 Hz respectively. The sampling frequency is 1024 Hz. A frame of 1024 samples in length is windowed using a Blackman window and is padded using 1024 zeros. The frequency is computed as per Equation

(3.4). As shown in Figure 3.2, these frequencies can be identified accurately and are indicated by arrows in the graph in Figure 3.2. Their frequencies are found by reading along the y-axis. The largest error among all the estimates of the components frequencies is only approximately 0.15 Hz.

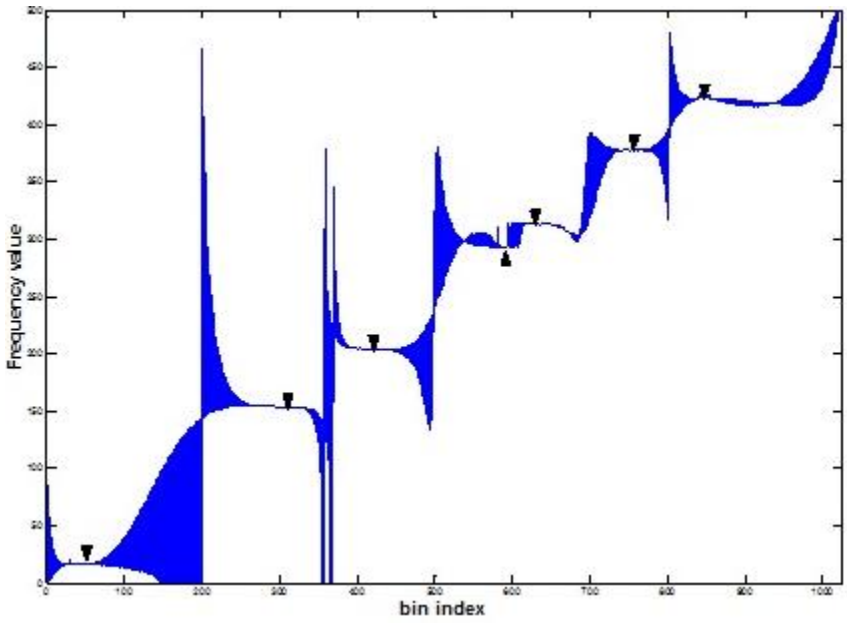


Figure 3.2 Frequency estimation of a multi-component signal by the CSPE algorithm (marked by arrows)

3.6 The performance comparison between CSPE and Quadratic Interpolation Estimation

An experiment was carried out to compare the accuracy of the Quadratic Interpolation Estimation Algorithm [RLV07] with the CSPE. The procedure of this experiment is described as follows:

1. First, twenty evenly spaced frequencies f_i ($0 < f_i < \frac{f_s}{2}$) are defined, where f_s is the sampling frequency.
2. For each initial frequency f_i, *m* random frequencies are generated, each of which has a small random fluctuation from f_i. Here, *m* = 1000.
3. *m* signals are created and each signal is composed of 20 components whose frequencies were generated in step 2.
4. The Root Mean Square (RMS) error of the frequency estimation by the CSPE and Quadratic Interpolation Estimation Algorithm for these *m* signals were calculated respectively for each f_i, which is shown in Figure 3.3. The x-axis denotes each f_i and the y-axis denotes the RMS error of each f_i estimation.

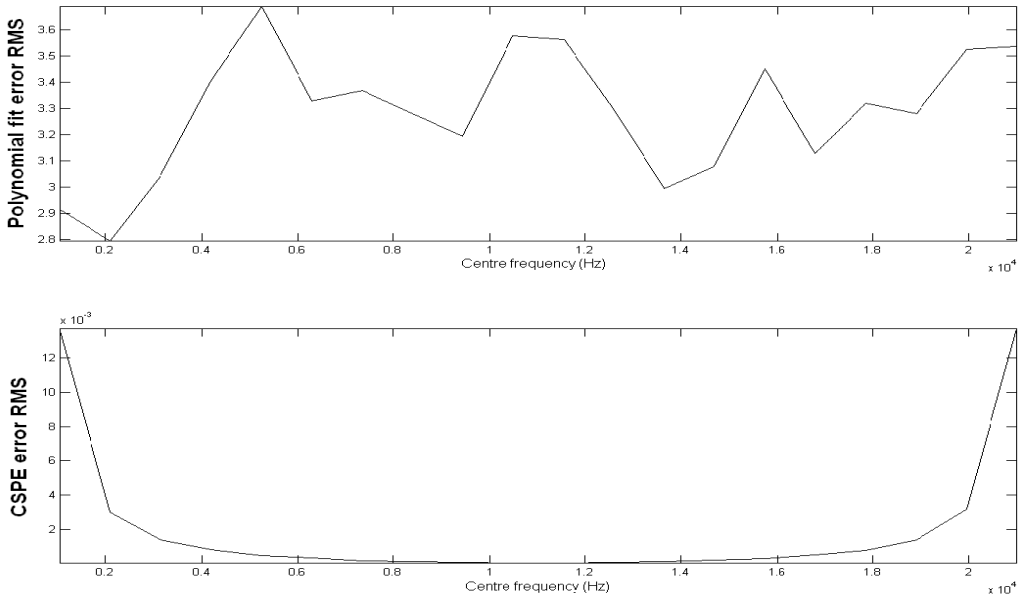


Figure 3.3 Accuracy comparison of frequency estimation between quadratic fit and CSPE

As shown in Figure 3.3, the RMS error of the CSPE estimate, which is of the order 10^{-3}, was found to be much smaller than that of the quadratic interpolation approach. Thus, the CSPE is more accurate in its frequency estimate than the quadratic interpolation estimation algorithm.

3.7 Issues with CSPE

The CSPE works properly when the components contained in the signal are constant and stable. However, there can be cases where some components will only appear for half or even less of the frame. Another experiment on an exemplar signal $x_b(n)$ was conducted. This signal has the same frequency components as $x_a(n)$ in Section 3.5, but each component of this signal appears for only a half or a quarter of the frame. The frame length was 1024. The resulting output of the CSPE was shown in Figure 3.4, where the x-axis denotes the bin number and the y-axis denotes the estimated frequency value. From Figure 3.4, it can be seen that there is no flat section in any part of the graph, meaning that none of frequency components were identified by the CSPE.

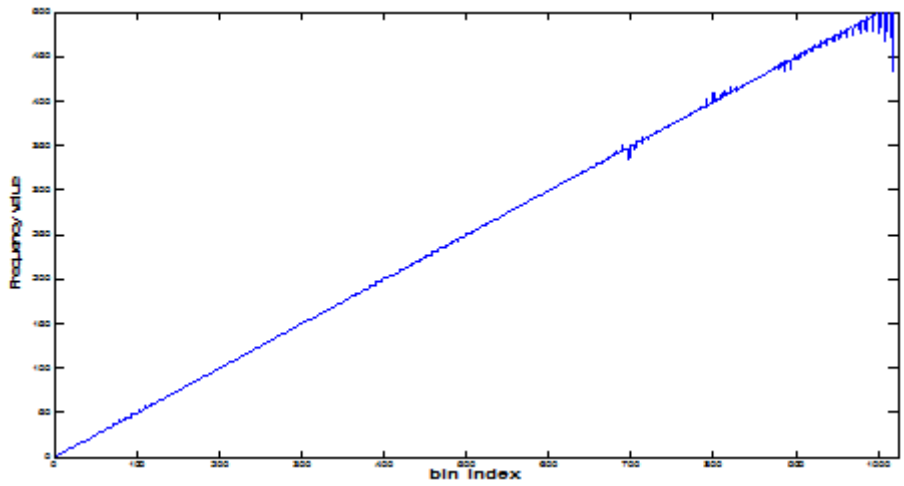


Figure 3.4 Frequency components that cannot be identified by CSPE

To compare, a straightforward FFT analysis of this signal was performed and its spectrum was shown in Figure 3.5, where the x-axis denotes the bin number and the y-axis denotes the magnitude value of each bin number. From Figure 3.5, it can be seen that there are peaks appearing in different bins, which suggests that the corresponding frequency components might exist in the signal. Therefore, if each component appears only in half or less of the frame, which can easily happen in real audio signals, the CSPE is unable to identify each of them. This is one of limitations of the CSPE as proposed in [GS06] and it needs to be overcome in order to apply this algorithm to the real signals.

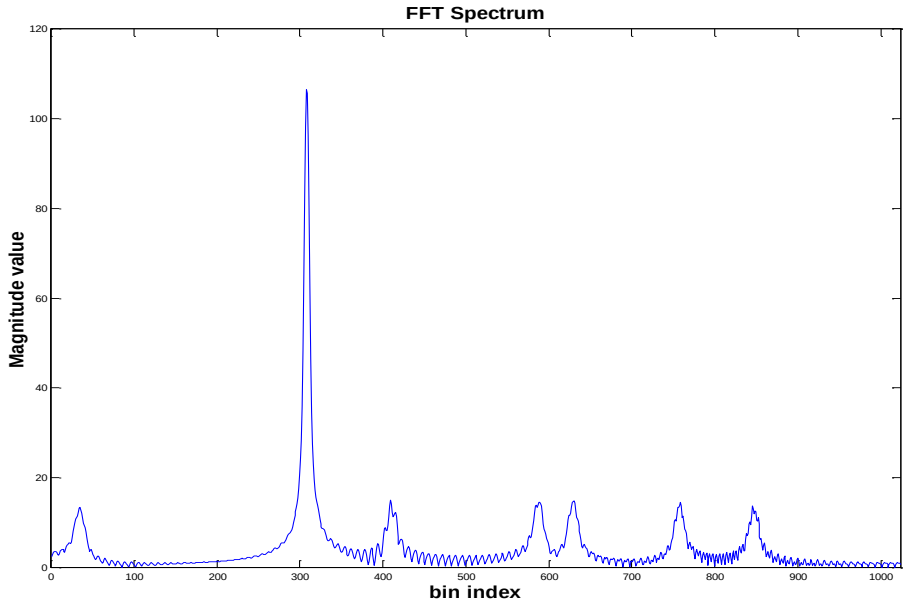


Figure 3.5 FFT spectrum of the signal

3.8 Improvement on CSPE

Suppose there are three signals: $x_2(n)$, $x_3(n)$, $x_4(n)$, each with length 1024 samples, sampling frequency 1024 Hz and comprising one frequency component at 123.5 Hz. The only difference between the three signals is that the component appears over the entire length of $x_2(n)$, while it appears only in a half and quarter length of signals $x_3(n)$ and $x_4(n)$ respectively. If a normal FFT analysis is performed, the single component with frequency value of 123.5 Hz will not be centred on any frequency bin. As a result, the FFT analysis produces a representation with significant peaks at the 124^{th} and 125^{th} bins. Denote $u_3(n)$ and $u_4(n)$ as two different unit step functions as defined in Equation (3.7) and (3.8), where *N* is the frame length.

$$u_3(n) = \begin{cases} 1 & 0 \le n \le N/2 \\ 0 & \frac{N}{2} + 1 \le n \le N \end{cases} \tag{3.7}$$

$$u_4(n) = \begin{cases} 1 & 0 \le n \le N/4 \\ 0 & \frac{N}{4} + 1 \le n \le N \end{cases} \tag{3.8}$$

It is possible to rewrite $x_3(n)$ and $x_4(n)$ as the product of $x_2(n)$ with $u_3(n)$ and $u_4(n)$ respectively, as defined in Equation (3.9) and (3.10).

$$x_3(n) = x_2(n)u_3(n) \tag{3.9}$$

$$x_4(n) = x_2(n)u_4(n) \tag{3.10}$$

Denoting F_{x_2} and F_w as the FFT transform of $x_2(n)$ and a window function $w(n)$ respectively, such as a Blackman window, the windowed FFT transform of the signal $x_2(n)$ can be written as:

$$F_{x_{2w}} = F_{x_2} * F_w \tag{3.11}$$

where * denotes the convolution. Similarly, the windowed FFT transform of the signals $x_3(n)$ and $x_4(n)$ can be written as:

$$F_{x_{3w}} = F_{x_3} * F_w = F_{x_2} * F_{u_3} * F_w \tag{3.12}$$

and

$$F_{x_{4w}} = F_{x_4} * F_w = F_{x_2} * F_{u_4} * F_w \tag{3.13}$$

Examining Equation (3.12) and (3.13) it is possible to interpret the terms $F_{u_3} * F_w$ and $F_{u_4} * F_w$ as the actual windowing operation that is applied to the signal $x_2(n)$ in the frequency domain. If the FFT transform of the original window function and the other

two window functions applied to $x_2(n)$ are compared, that is, F_w, $F_{u_3} * F_w$ and $F_{u_4} * F_w$, simplifying the notation as F_w, F_{w3} and F_{w4}, it can be seen that there is an important difference in the width of their main-lobe and the depth of their side-lobe, as shown in Figure 3.6. The x-axis denotes the bin number and the y-axis denotes the magnitude in dB. More specifically, when the signal contains components that do not appear over the entire frame, the side-lobe of its actual window function spectrum are not suppressed significantly, as F_{w3} and F_{w4} show in Figure 3.6. This affects the CSPE in such a way that the self-interaction term outlined in Equation (3.2) is not sufficiently suppressed, which results in the CSPE being incapable of finding the exact signal frequency of $x_3(n)$ and $x_4(n)$.

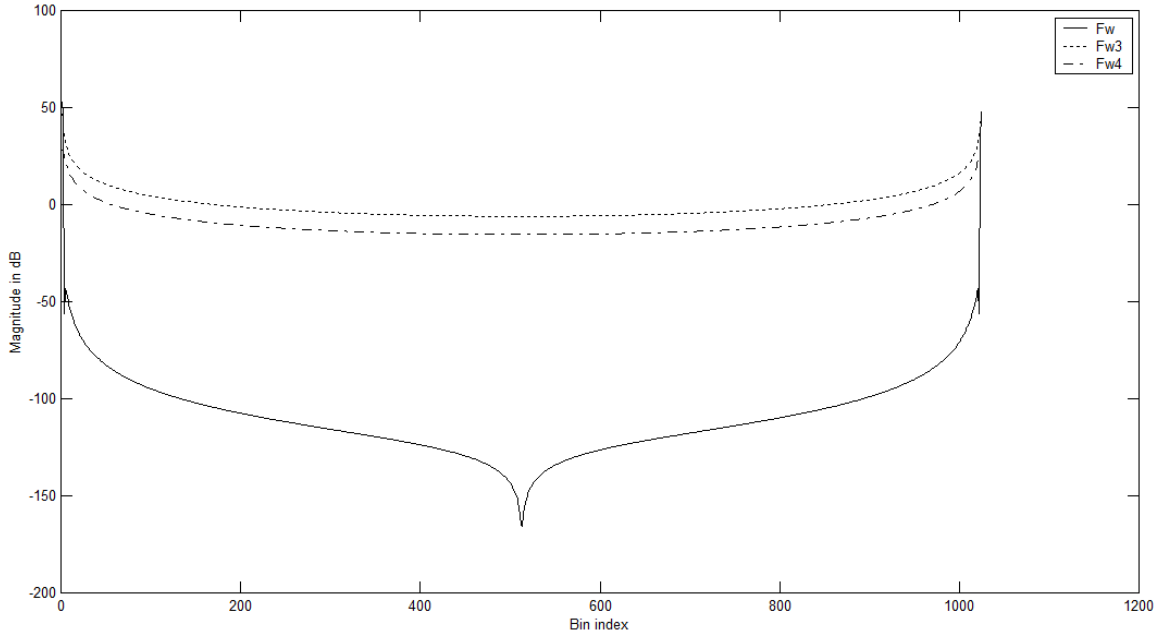


Figure 3.6 Magnitude response of three different window functions

Inspired by the reason of introducing a window function in the FFT analysis, a second window function was introduced to suppress the side-lobe caused by the convolution effect of the window spectrum with the step function. This is known as apodization [SDF95]. It is more commonly used in image processing. Normally, an apodization function is used to suppress the effect of side-lobe at the expense of lowering the spectral resolution. The experimental results in [TFJ00] have shown that the Kaiser Window function is better for apodization than other window functions such as the Poisson, Gaussian or Tukey. The apodization function can be defined as Equation (3.14):

$$w_A(n) = (w_{ks}(n))^{\alpha_1} \tag{3.14}$$

where $w_{ks}(n)$ is defined as Equation (3.15):

$$w_{ks}(n) = 1 - kaiser\ (N, \beta_1) \tag{3.15}$$

where $kaiser\ (N, \beta_1)$ is the Kaiser window function, and β_1 is a parameter to control the side-lobe levels. The side-lobe suppression effect is dependent on the coefficients α_1 and β_1. The impact of different values of α_1 and β_1 on the suppression effect was evaluated by experiment. An example of the effect of suppression of the side-lobe by the apodization function is depicted in Figure 3.7. It can be found that when $\alpha_1 = 3$ and $\beta_1 = 0.01$, it has a side-lobe suppression level greater than 300 dB. In general, when α_1 is bigger and β_1 is smaller, the side-lobe suppression is better.

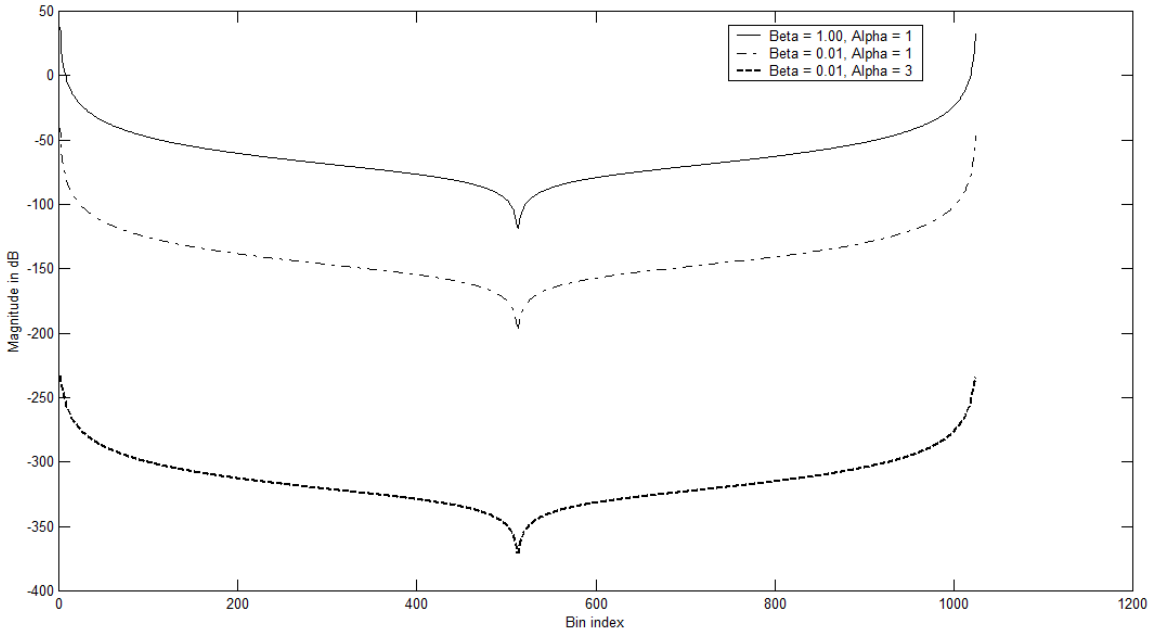


Figure 3.7 Magnitude response of apodization function

Next, the signal $x_b(n)$ was analyzed with the improved CSPE, which incorporated the apodization function as defined in Equation (3.14) ($\alpha_1 = 3$, $\beta_1 = 0.01$). The CSPE frequency identification result is shown in Figure 3.8 where the arrows indicate the identified frequency components. It can be seen from Figure 3.8 that the improved CSPE is capable of identifying all the frequency components.

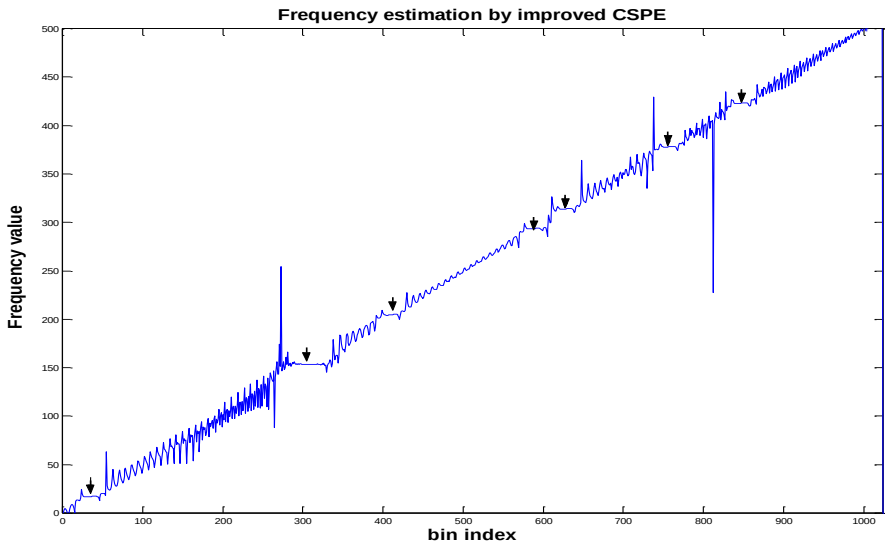


Figure 3.8 Frequency estimation by improved CSPE

However, when frequency components appear over a different proportion of a frame, the value of α_1 and β_1 has to be adapted to achieve a satisfactory identification result. An experiment was carried out to give a reference to the user on which values of α_1 and β_1 should be used when frequency components appear over a different proportion of a frame. The experimental result is shown in Table 3.1, where 'Y' means the frequency component can be identified and 'N' means the frequency component cannot be identified. It can be seen that when α_1 = 18 and β_1 =0.01, it can identify frequency components which only appear over 1/16 of a frame. However, care has to be taken when selecting the value of α_1, as there is a trade-off between the level of side-lobe suppression and main-lobe width [SAH05].

In summary, the improvement on the CSPE [GS06] by incorporating the apodization function is important for the analysis of real audio signals, where frequency components contained can appear over any proportion of a frame. In the next section, an audio watermarking algorithm based on this improved CSPE is developed.

Table 3.1 Configuration of the coefficients α_1 and β_1

	proportion of a frame			
	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$\alpha_1 = 1$, $\beta_1 = 0.01$	Y	N	N	N
$\alpha_1 = 3$, $\beta_1 = 0.01$	Y	Y	N	N
$\alpha_1 = 10$, $\beta_1 = 0.01$	Y	Y	Y	N
$\alpha_1 = 18$, $\beta_1 = 0.01$	Y	Y	Y	Y

3.9 Using the improved CSPE in audio watermarking

In this section, a watermarking algorithm based on the improved CSPE is developed and is first applied to synthetic signals. The framework for this is the FFT-based watermarking approach. As mentioned in Chapter 2, the algorithm proposed in [FM09] is simple, computationally efficient and can possibly achieve an acceptable performance. However, its embedding position is known. This should be improved, as it makes the embedded watermark easy to remove. In addition, its imperceptibility can be improved by using the super-resolution CSPE tool, as the frequency component estimation by the FFT is not accurate enough so that the spectral manipulation used in the algorithm can

introduce some audible distortions. The proposed algorithm will be detailed in the following sections.

3.9.1 Embedding

The basic embedding procedure of this algorithm can be explained as follows: the host signal is first segmented into frames of uniform length and each frame is then analyzed using the improved CSPE to identify the magnitude and phase of the components it contains. Then two components are chosen based on a user defined reference value, followed by manipulation on these two components according to a defined rule in order to embed a watermark bit. The process of selection and modification of the two candidate components will be detailed in the following sections.

3.9.1.1 Dynamic selection of two candidate components

It was decided to make the process of choosing the candidate components as flexible as possible by defining this as a dynamically chosen pair of values. It depends on a user-defined reference bin value *r* but is also dependent on the signal under consideration and the ability of the CSPE to identify the components of this signal. The candidate components were defined as being the nearest frequency bins above and below the user-defined reference bin value *r* by more than a predefined threshold *Th*. The purpose of using this threshold is to avoid selecting two frequency bins that are too close to each other.

Denote *lc* as the highest CSPE-detected frequency bin that is lower than *r*, by more than a threshold *Th*. Denote *rc* as the lowest CSPE-detected frequency bin that is

above *r* by more than *Th*. A simple line graph, as shown in Figure 3.9, can be used to show the selection of the candidate components. As seen from Figure 3.9, there are 7 frequency bins identified by the CSPE, b_1, b_2, b_3, b_4, b_5, b_6, b_7. The *lc* would be the b_3, as it is the first bin that below *r* by more than *Th*. The *rc* would be b_5, as it is the first bin that above *r* by more than *Th*.

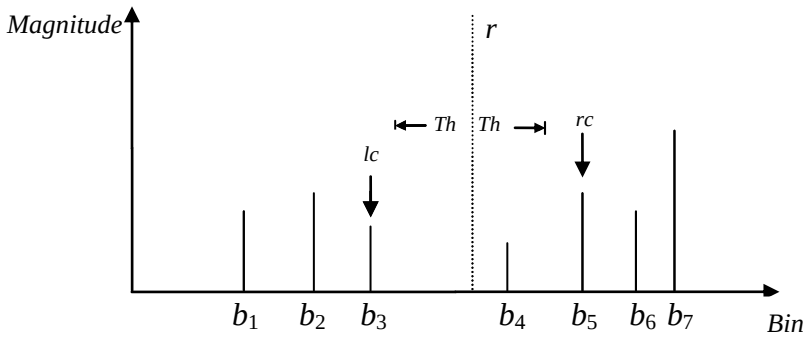


Figure 3.9 The demonstration of candidate component selection

The reference bin value *r* is, in effect, a private key and this value is needed in order to detect the watermark. This adds to the security of the algorithm. In addition, as the embedding position is dynamically selected dependent on each frame, thus it is difficult for the attacker to remove the watermark.

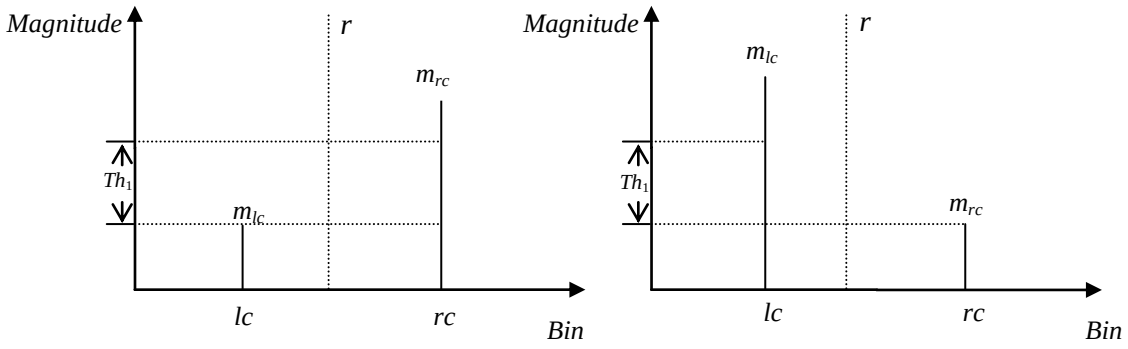
3.9.1.2 Embedding rules

Embedding rules are defined as Equation (3.16).

$$\text{If } bit = 0 \text{ let } m_{rc} > m_{lc} + Th_1 \tag{3.16}$$

$$\text{If } bit = 1 \text{ let } m_{lc} > m_{rc} + Th_1$$

where m_{lc} and m_{rc} denote the magnitude of component *lc* and *rc* respectively, and Th_1 denotes the threshold of the difference between m_{lc} and m_{rc}. This can be demonstrated in Figure 3.10.



(a) embed a '0' (b)embed a '1'

Figure 3.10 The demonstration of embedding a bit '0' or '1'

Assuming that the enemy knows the original signal and the watermarked signal, a simple subtraction of one from the other will provide the difference (i.e. the watermark). However, there is no way of knowing if each difference actually represents a bit '0' or '1', as embedding either bit of '0' or '1' would result in the difference. This consideration makes this method more favourable than [HT09] and many others.

At the embedding stage, according to Equation (3.16), the algorithm would first compare the magnitude of both candidate components in any given frame before deciding if any modification is required. If they are already in the correct relationship, no modification is needed. Otherwise, one of them should be modified. The way of modifying a component is explained in the next section.

3.9.1.3 Modification of one of the candidate components

As mentioned previously, the CSPE can be used to identify a component extremely accurately within a signal, and then can be used to calculate its phase and magnitude. Let us assume that the magnitude of *lc* is lower than that of *rc*, in a frame in which it needs to be of a higher magnitude to represent a '1' bit according to Equation (3.16). Because the component's phase and frequency have been estimated accurately, the modification can be applied in time domain so that it introduces less distortion. More specifically, by adding a component with the same frequency and phase but of different magnitude, the magnitude of the candidate component can be modified directly to satisfy Equation (3.16). There are two ways of modification, one of which is to increase the magnitude of the component *lc*, as illustrated by Equation (3.17):

$$x'(n)=x(n)+(m_{rc}-m_{lc}+Th_1)cos(2\pi f_{lc}\,n+\emptyset_{lc}) \tag{3.17}$$

where $x(n)$ and $x'(n)$ represents the original signal and the manipulated signal, f_{lc} and $\emptyset_{lc}$ represent the frequency and phase of the component *lc*. By Equation (3.17), the magnitude of the component *lc* will be increased to $m_{rc} + Th_1$, so that the rule of embedding a bit '1' is met.

Alternatively, the magnitude of the component *rc* can be decreased to meet the rule of embedding a '1' bit, as shown in Equation (3.18):

$$x'(n)=x(n)+(-m_{rc}+m_{lc}-Th_1)cos(2\pi f_{rc}\,n+\emptyset_{rc}) \tag{3.18}$$

where f_{rc} and $\emptyset_{rc}$ represent the frequency and phase of component *rc*. By Equation (3.18), the magnitude of component *rc* will be decreased to $m_{lc} - Th_1$, so that the rule of embedding a bit '1' is met.

Based on experiments, it was found that reducing one component rather than increasing another component to satisfy Equation (3.16) introduces less audible distortion. Likewise, if the magnitude of *rc* is lower than that of *lc*, in a frame in which it needs to be of a higher magnitude to represent a '0' bit according to the embedding rule, the Equation (3.19) can be used to modify the component *lc* so that the magnitude of component *lc* will be decreased to $m_{rc} - Th_1$.

$$x'(n)=x(n)+(-m_{lc} + m_{rc} - Th_1)cos(2\pi f_{lc}\, n+\emptyset_{lc}) \tag{3.19}$$

3.9.1.4 Embedding process

To summarize, the procedure to embed a watermark bit, either '0' or '1', is depicted in the block diagram shown in Figure 3.11. As can be seen from Figure 3.11, if the watermark bit to be embedded is a '1' and $m_{lc} > m_{rc} + Th_1$, or if the watermark bit to be embedded is a '0' and $m_{rc} > m_{lc} + Th_1$, nothing needs to be changed. Otherwise, the manipulation will be applied according to Equation (3.18) or (3.19).

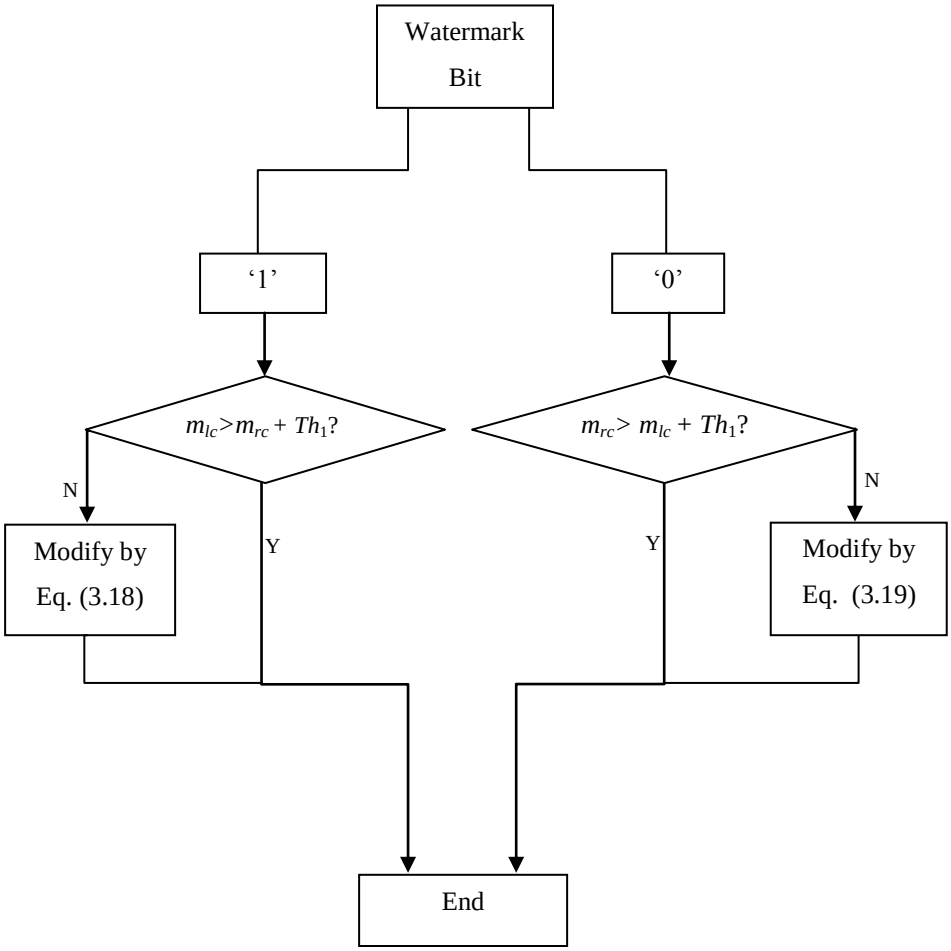


Figure 3.11 The procedure of a watermark bit embedding

In addition, the predefined reference bin value *r* could result in a failure to find two appropriate candidate components. For example, if *r* is 2000, then two components with frequency bin lower and higher than 2000 respectively need to be selected from the signal. However, it could happen that the signal may not contain any component whose frequency bin is below 2000, or any component whose frequency bin is above 2000. Even though this might be a rare occurrence, especially for real signals, it would make

the component selection fail. In order to solve this problem, an iterative search process can be incorporated to find the first reference bin value, after which two appropriate candidates for modification can be selected. This reference bin value will be used as a private key. As an added-value of this process, the iteration can continue to find more suitable reference bin values, which could be used as a form of redundancy to enhance the robustness, or as a form of obfuscation to enhance the security of the proposed algorithm. Furthermore, by this additional process, the reference value will be dynamically determined based on each frame of the signal, which also enhance the security of the algorithm.

3.9.2 Detection

In order to detect the watermark, the system must be provided with the reference value *r*, along with other parameters such as the frame length, the window type, and α_1 and β_1 used in Equation (3.14). The watermarked audio signal is then segmented into frames using the same frame size as is used for embedding. Two candidate components are selected by the improved CSPE. The system calculates the magnitude of the candidate components, and performs a comparison. From this comparison, the watermark bit sequence will be detected.

3.9.3 Evaluation

An experiment was carried out on 500 synthetic signals to evaluate the performance of this watermarking scheme. Each signal contains many equally spaced frequency components with a variety of frequencies across the human hearing range of 20 Hz to

20,000 Hz, and is sampled at 44100 Hz. The number of components in each generated signal is not constant. For each signal, a unique step constant which defines the distance in Hz between two neighbouring frequency components is randomly generated. Therefore, to ensure no possibility of two signals being identical, 500 step constants are created that range from 169 Hz to 668 Hz. For each synthetic signal, a randomly generated 150 watermark bits are embedded and then detected. The result of this experiment is depicted in Figure 3.12, where the x-axis denotes the signal number and the y-axis denotes the *Precision*, which is defined in Chapter 2.

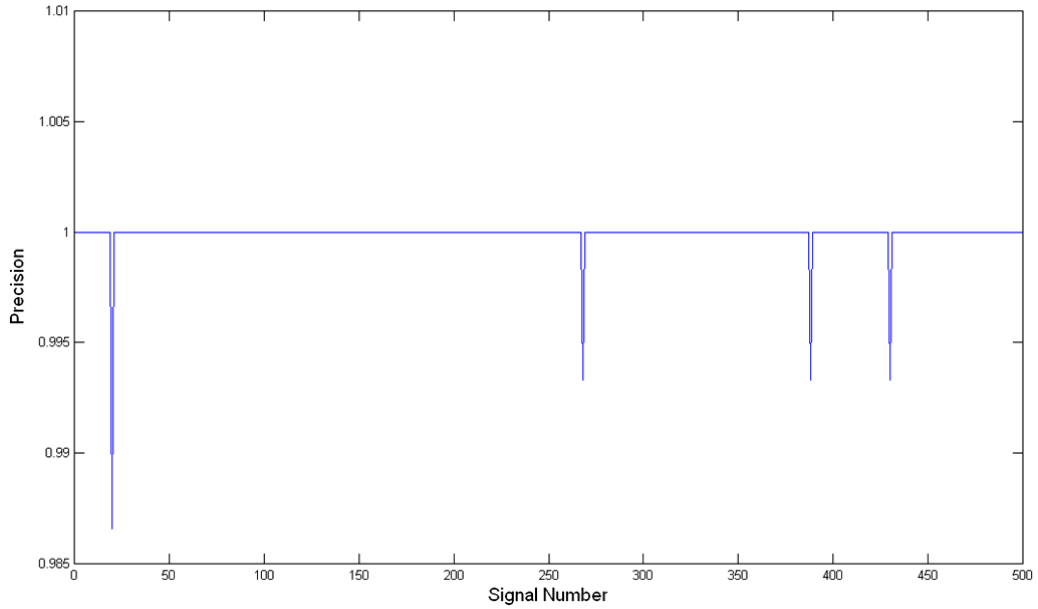


Figure 3.12 The *Precision* of each Signal

The histogram distribution of *Precision* is shown in Figure 3.13.

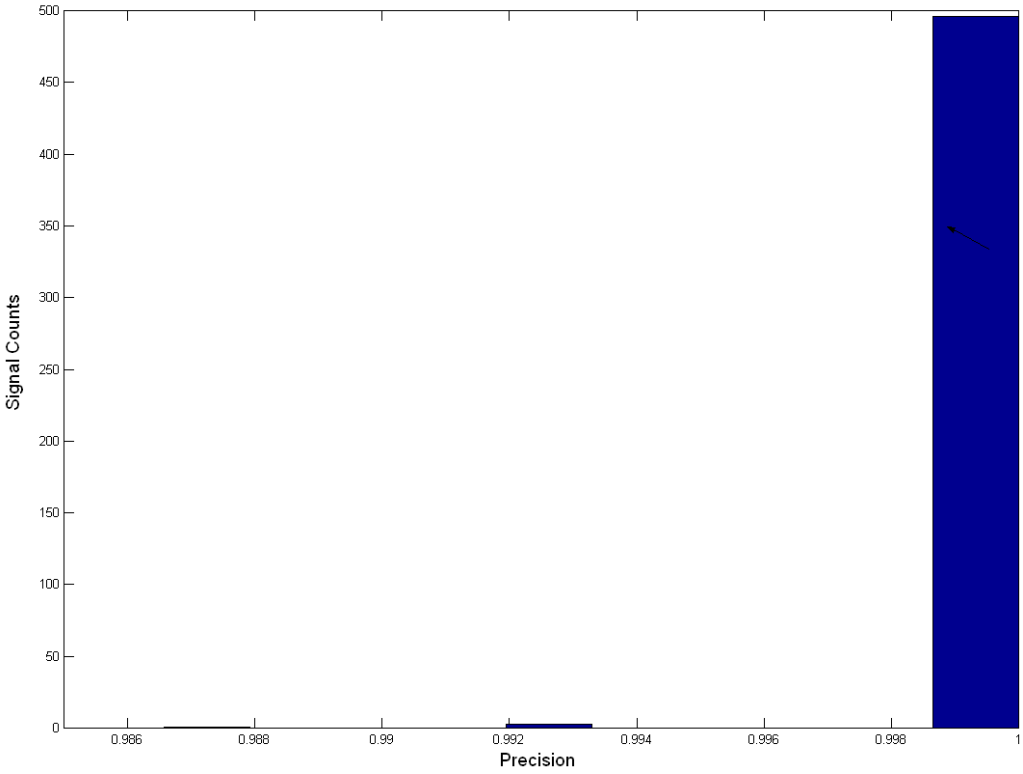


Figure 3.13 The Histogram distribution of *Precision*

From the Figure 3.12 and Figure 3.13, it can be seen that the accuracy is extremely high. 99.2% of signals produce a *Precision* of 100%. This means that for 99.2% of the 500 randomly generated signals, the watermark was detected perfectly. Of those not perfectly detected, the bit sequence recovery rate was above 98.66%. In the next section, the proposed watermarking algorithm will be applied to real audio signals.

3.10 Application to Real audio Signals

Real audio signals contain a more complicated structure of frequency components and each frequency component can change over time, which increases the difficulty of

identifying them correctly. However, this obstacle has to be overcome as applications such as copyright protection and ownership proof are based on real audio signals.

If the presence of an embedded message can be detected, then attempts can be made to attack it. However, if the presence of an embedded message cannot be detected, then no such intentional attacks would be applied [Bla10]. Therefore, in developing an audio watermarking algorithm based on real audio signals, the imperceptibility and accuracy will be considered primarily, and then followed by the robustness against common or expected attacks.

3.10.1 Issues with the proposed audio watermarking algorithm

An experiment was conducted to verify the accuracy of the proposed watermarking algorithm on real audio signals. 20 music files were randomly selected from a collection of more than 300 music files with a variety of genres. The 20 watermark bit sequences were randomly generated. Each sequence contains 150 watermark bits. The *Precision* distribution is depicted in Figure 3.14. As can be seen from the Figure 3.14, the *Precision* achieved without any attack is quite low, ranging from 56.25% to 81.25%. The $Precision_{mean}$ achieved was 68.85%.

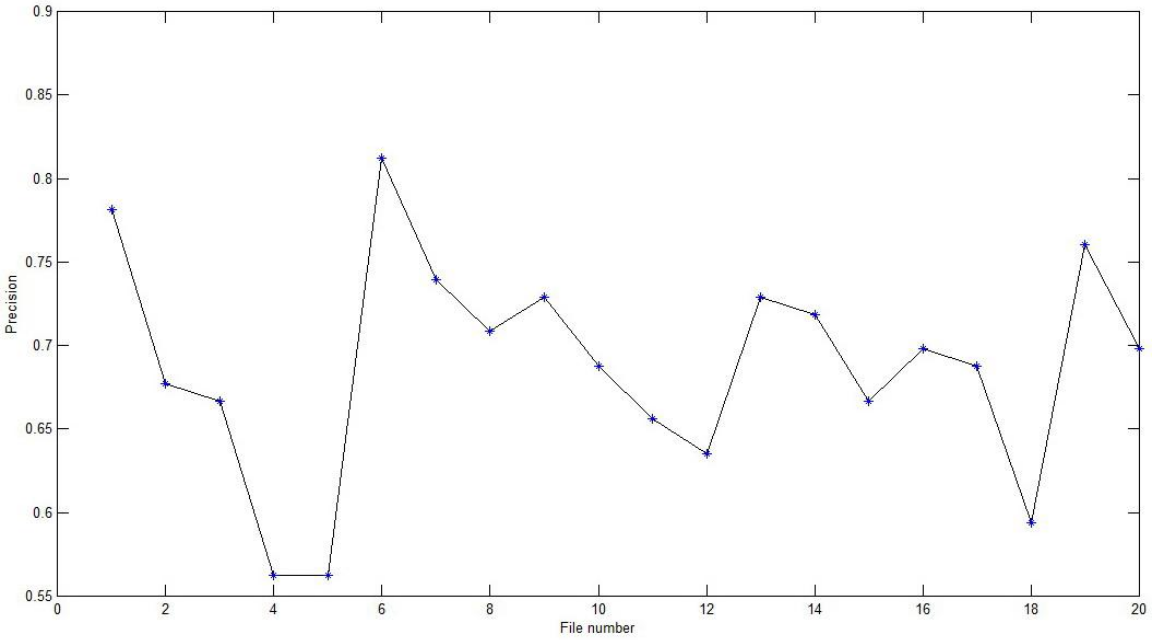


Figure 3.14 The Precision distribution of 20 music files

By analyzing the watermarked audio signals, it was found that, on occasion, components selected by the CSPE as candidates for modification at the embedding stage were not subsequently selected at the detection stage. Since only the magnitudes of selected components were modified, it seemed that this modification was the reason for the inability of CSPE to identify the component after watermarking. However, the CSPE method does not rely on a component's magnitude to identify it in a signal so it is likely that some other indirect result of the modification has affected the identification of components in the watermarked signal by the CSPE. This will be explored in the next section.

3.10.2 The reason for the low *Precision* of the watermarking algorithm

When two frequency components are too close to each other, the self-interaction term outlined in Equation (3.2), which results from interference between two closed components, could obscure one of the components with a much lower magnitude [CH00, GS06]. Therefore, it can be hypothesized that the components, which had been identified by the CSPE before modification, may have become impossible to identify because of their magnitude being reduced and then obscured by a spectrally close component.

An experiment was set up to test this hypothesis. A signal $x_5(n)$ containing two components *a* and *b* was created. These two components have a frequency value of 105 Hz and 107 Hz respectively, and have initial amplitude of 1.0 and 0.3 respectively. The window length used was 2048 samples and the sampling frequency was 512 Hz. Before any modification, both components *a* and *b* can be identified by the CSPE, which can be seen from the flat sections in the upper panel of Figure 3.15. The x-axis denotes the bin number and the y-axis denotes the frequency value in Hz identified by CSPE. The flat section, which means the identified frequency component, is marked by the arrow. After the amplitude of component *b* is reduced to 0.015, CSPE can only identify component *a*, as can be seen that there is only one flat section in the lower panel of Figure 3.15.

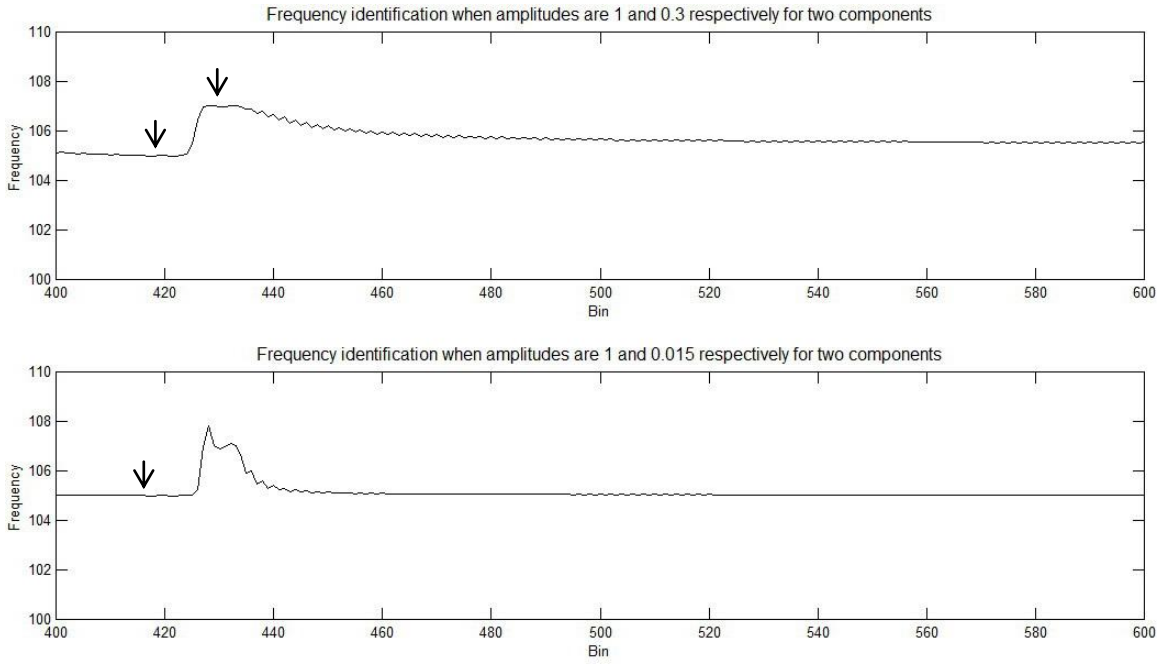


Figure 3.15 CSPE frequency identification of a signal containing two components whose frequencies are very close

As a comparison, a signal $x_6(n)$ only contains the component *b* with a frequency of 107 Hz. The window length used was 2048 samples and the sampling frequency was 512 Hz. It can be identified by the CSPE when its amplitude is 0.3, as shown in the upper panel of Figure 3.16. The x-axis denotes the bin number and the y-axis denotes the frequency value in Hz identified by CSPE. The flat section is marked by the arrow. When its amplitude is reduced to 0.015, it still can be identified by CSPE, as shown in the lower panel of Figure 3.16.

Therefore, the reduction of the magnitude of one component can render it obscured by its adjacent components so that it cannot be identified by CSPE. This phenomenon was

the source of the problems for the low *Precision* of watermark detection. Therefore, some measures have to be taken to avoid this problem.

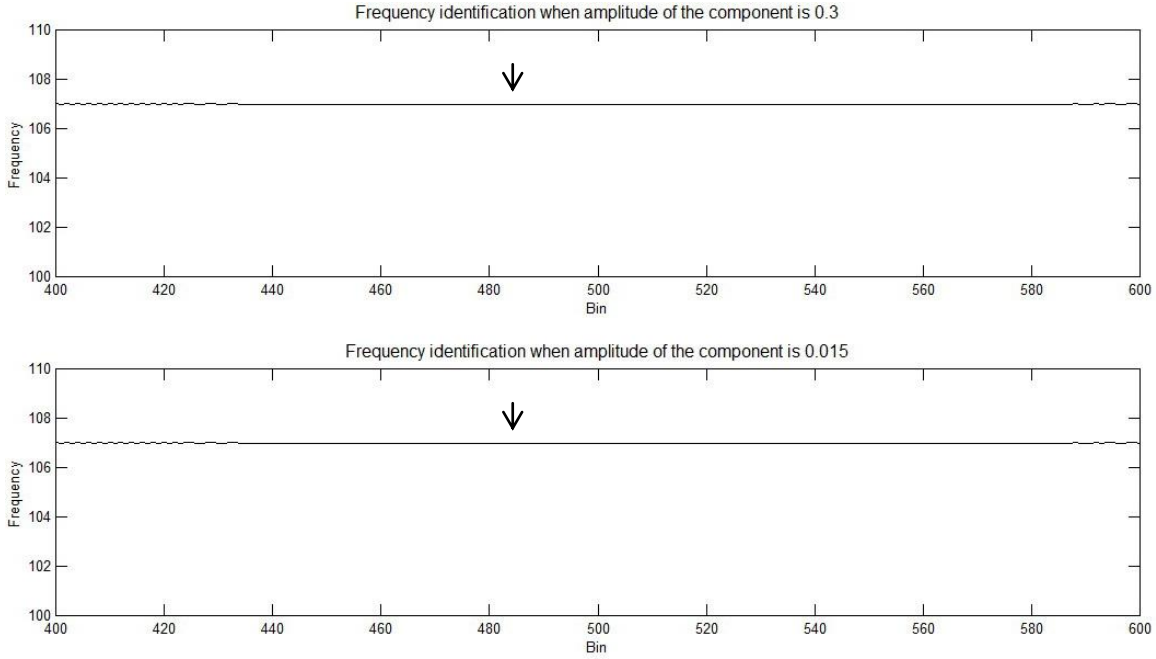


Figure 3.16 CSPE frequency identification of a signal containing one component

3.10.3 Using bin location to define embedding rules

One scenario with the method in Section 3.9.1.2 is: a bit to embed is a '1', $m_{lc}<m_{rc}$ but m_{rc} is very low. In order to satisfy Equation (3.16), m_{lc} has to be reduced to almost zero according to Equation (3.18). This component might subsequently be impossible to detect at the detection stage because of the phenomenon mentioned above.

By using the frequency bin value, instead of the magnitude value, as the rule for modification, this issue can be addressed. A new embedding rule can be defined as follows:

$$\begin{aligned} &\text{If } bit=0 \text{ let } |lbin\%2 - rbin\%2| = 0 \\ &\text{If } bit=1 \text{ let } |lbin\%2 - rbin\%2| = 1 \end{aligned} \tag{3.20}$$

where '*%*' denotes the modulus operation, *lbin* represents the bin location of the component *lc*, *rbin* represents the bin location of the component *rc*. The candidate components selection method is same as that defined in Section 3.9.1.2.

By way of illustration, if the watermark bit is a '0', it requires that the bin location of both components should be either both odd or both even (in which case |*lbin*%2 – *rbin*%2| returns 0). If the watermark bit is a '1', then one bin location should be odd and the other bin location should be even (in which case |*lbin*%2 – *rbin*%2| returns 1). This can be explained by Figure 3.17. If the watermark bit to embed is a '1' and if the bin value of *lc* and *rc* is 23 and 37 respectively, that is, *lbin* = 23 and *rbin* = 37, then Equation (3.20) is not satisfied as |23%2 – 37%2| returns 0.

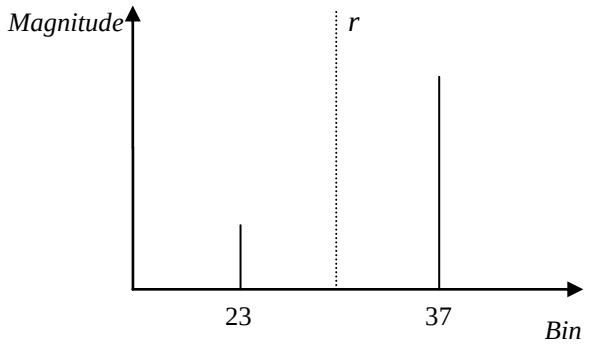


Figure 3.17 The demonstration of the embedding rule as shown in Equation (3.20)

3.10.4 Modification of the selected components

When the bin locations of the candidate components identified at the embedding stage do not satisfy Equation (3.20), manipulation has to be applied so that the bin location, either

lbin or *rbin*, would be changed. Instead of being a negative effect, the obscurity effect mentioned in Section 3.10.2 can be utilized to manipulate the candidate component to change one of the bin locations.

More specifically, assume the bit to embed is a '1', *lbin* = 23 and *rbin* = 37, as shown in Figure 3.18 (a). As Equation (3.20) is not satisfied, manipulation has to be applied to change either *lbin* or *rbin* into an even number. Assume the manipulation applied is to reduce the magnitude of *rc*, this will make *rc* obscured and unable to be identified. Then, a different candidate bin location, will be identified at the detection stage as shown in Figure 3.18 (b), which will fulfill Equation (3.20).

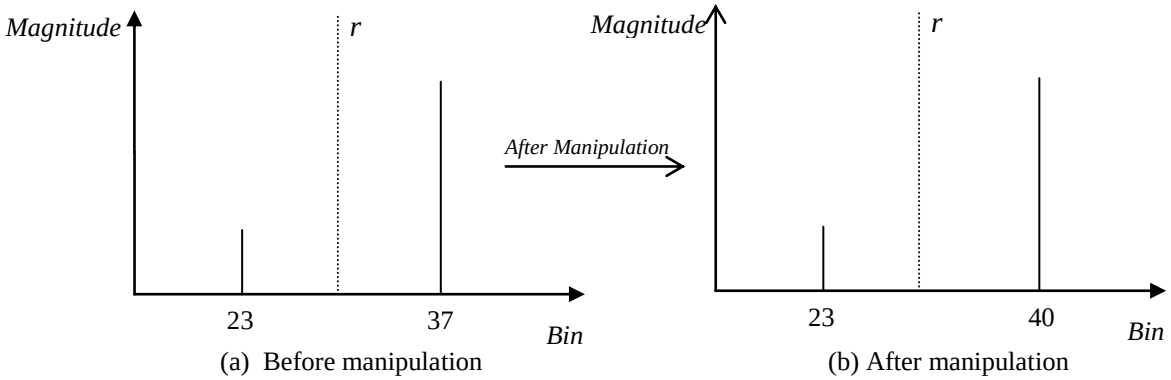


Figure 3.18 The demonstration of bin location change by manipulation in order to satisfy Equation (3.20)

However, there is no guarantee that Equation (3.20) would be fulfilled after the manipulation. Thus, a component verification process was incorporated to ensure that Equation (3.20) would be fulfilled both at the embedding stage and detection stage without any attack. This will be detailed in the next section.

3.10.5 Component verification process

As an additional step to improve detection accuracy, at the embedding stage, the watermarked signal after modification of the components at locations *lbin* or *rbin* was analyzed again to see if the newly selected candidate components' bin locations from the modified signal meet Equation (3.20). If not, the magnitude of one of the candidate components will be reduced again to render it obscure. This process will be performed iteratively until Equation (3.20) is met. By performing this step, an overhead was added to the embedding stage but it is a worthwhile compromise for the improved accuracy.

3.10.6 Transparency and audible artifacts

In an audio watermarking system intended for use with real music, audible artifacts are unacceptable as they allow listeners to deduce that there is a watermark present as well as reducing the commercial value of the audio. One negative characteristic of this watermarking scheme is the introduction of unexpected audible artifacts in the form of 'clicks'. Various frames where these artifacts were present were analyzed. It was noticed that two distinct types of artifacts occurred for two different reasons. The resultant artifacts were defined as 'Type I' or 'Type II' clicks. The analysis of two exemplar frames (for example, 82^{th} frames and 313^{th} frame) from a watermarked signal is used to demonstrate both types of 'click'.

3.10.6.1 The reason for 'Type I' click

The reason for the 'Type I' click can be illustrated by Figure 3.19. It shows the spectrum of the 82^{th} frame of the original, intermediate, and watermarked signal

respectively. The solid line in Figure 3.19 denotes the spectrum of the 82th frame of the original signal. The dashed bold line denotes the spectrum of the 82th frame of the signal after it has been modified once (denoted as the 'intermediate signal'). The dotted line denotes the spectrum of the 82th frame of the watermarked signal (denoted as the 'final signal'), in which the selected bins satisfy the criteria. The x-axis denotes the bin number and the y-axis denotes the magnitude value.

Recall that in order to represent the watermark, the algorithm only reduces the magnitude of – at most – one component in any frame. Any difference between the modified signal and the original signal would therefore centre on the bin in which the component has been changed.

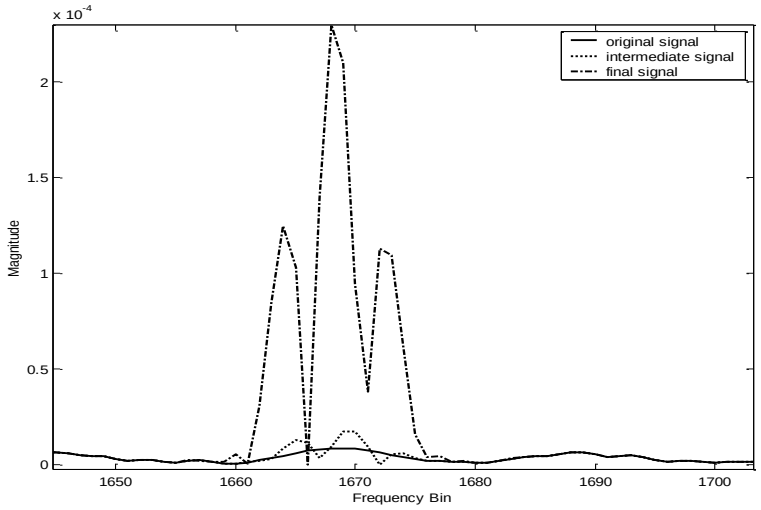


Figure 3.19 Spectrum of the 82th frame for original signal, signal after first change and final signal, illustrating Type 1 click.

As can be seen from Figure 3.19, the selected component's magnitude in the final signal is apparently much larger than that in the original signal and noticeably larger than

those of its neighboring bins' magnitudes. Given that the algorithm only reduces the magnitude, instead of increasing the magnitude, this should not happen. By investigating this phenomenon, it was found that it occurs because candidate components, which actually do not exist in the signal, are identified by the CSPE. This is termed as 'ghost components'. When a 'ghost component' is identified by the CSPE as a candidate, if the criterion is not satisfied, the algorithm then intends to decrease this ghost component's magnitude. However, this modification actually creates the reverse effect, that is, it adds a new component into the signal whose phase and frequency are same as this 'ghost component', but the magnitude is equal to the value that it should be reduced by. This added component causes the 'Type I' click.

3.10.6.2 The reason for 'Type II' click

The reason for the 'Type II' click is subtly different from that for the 'Type I' click. The spectrum of the 313^{th} frame, shown in Figure 3.20, demonstrates the 'Type II' click. The cause of the 'Type II' click can be described as follows: as with the 'Type I' phenomenon, a 'ghost component' is identified by CSPE and reducing its magnitude results in adding this component into the signal. However, in this scenario, the added component's bin location still does not satisfy the criteria so its magnitude is further reduced. This should make its magnitude very low in the final signal. However, from the spectrum in Figure 3.20, it can be seen that the component's magnitude is not as low as expected. Conversely, it cannot be identified by the CSPE and another component that

meets the criteria is then selected. The artifact left behind by the added ‘ghost component’ is audible. This is called ‘Type II’ click.

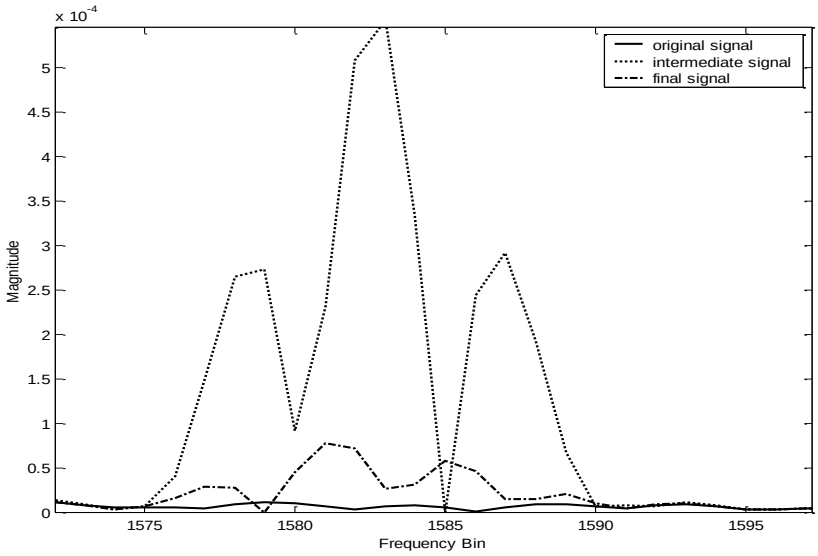


Figure 3.20 Spectrum of the 313th frame for original signal, signal after first change and final signal, illustrating Type II click.

3.10.6.3 The occurrence probability of both clicks

By repeated analysis of files that displayed the audible artifacts, it was found that all artifacts can be categorized as one of these two types. ‘Type II’ clicks occur very rarely, with an average probability of 0.2%~0.4% (for example, only 2~4 ‘Type II’ clicks occur when embedding 1000 bits). ‘Type I’ clicks are more common, with a probability of approximately above 1% (for example, 10 ‘Type I’ clicks occurs when embedding 1000 bits). The number of ‘Type I’ and ‘Type II’ clicks depends on the components present in any frame and their modification according to the watermark bits to embed. It

is therefore impossible to predict where, or in what type of audio frame, either type of click may occur.

3.10.6.4 The solution to the audible artefact issue

Since 'Type II' clicks occur only very occasionally, the recommended solution to this artefact is simply to return the magnitude of this component to its original value. This can result in a 0.2% to 0.4% reduction of the *Precision* but this is a small price to pay for increased perceptual transparency. For 'Type I' clicks, which occur more often, it was decided to find a proper solution to minimize what would have a big impact on the *Precision.* As shown in Figure 3.19, with regard to 'Type I' clicks, the modified component has a noticeably higher magnitude. Thus, it can be removed by reducing its magnitude to zero. Before removing either type of click, an additional step has to be added to the algorithm first to identify the type of click. The following rules were defined based on listening tests and signal analysis to identify the type of click:

1. If the modified signal contains a bin whose magnitude is 10 times greater than that of the original signal, this can be identified as 'Type I' click.

2. If the spectrum of the watermarked signal contains peaks whose magnitudes are different from those of the corresponding peaks in the original spectrum, then these peaks from both spectra are identified. If one of these peaks whose magnitude in the spectrum of the watermarked signal is 3 times greater than that in the original spectrum, and is greater than that of the neighbouring bins in the original spectrum, then this can be

identified as a ‘Type II’ click. The threshold used to identify the click as stated above can be defined dynamically, to adapt it to different genres of music.

A block diagram of the improved algorithm, which automatically detects, categorizes and removes ‘Type I’ and ‘Type II’ artefacts, is shown in Figure 3.21. As can be seen from the Figure 3.21, when the click is identified as ‘Type I’, its amplitude will be reduced again to remove this click. When the click is identified as ‘Type II’, the corresponding frame will be recovered to the original to remove this click.

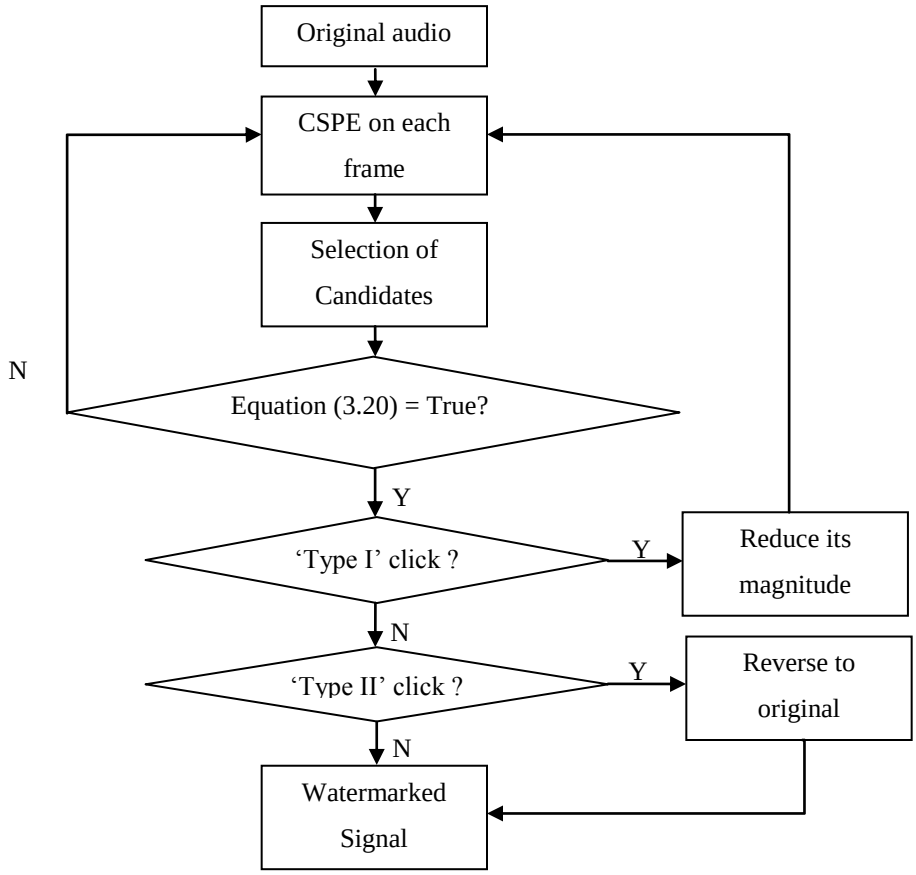


Figure 3.21 The embedding process of the improved algorithm

3.10.7 Watermark detection procedure

The detection process is very simple and computationally efficient. The flow chart of the detection process is shown in Figure 3.22. That is, the watermarked audio is analyzed by the CSPE on a frame-by-frame basis. Then, two candidates are selected and have their bin locations checked according to Equation (3.20), as a result, the watermark bit sequence can be derived.

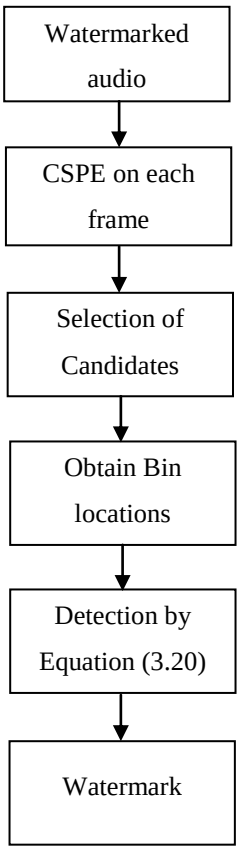


Figure 3.22 The detection process for each frame

3.10.8 Results

25 candidate music files of different genres from a collection of more than 300 were selected and the full embed-detection cycle on each file was performed. The 25 watermark bit sequences were randomly generated.

The *Precision* of the 25 files was above 99%. The few errors that were encountered were a direct result of the addition of the step to remove audible artefacts. Such a low rate of error is a satisfactory compromise when the alternative is to have audible artefacts in the watermarked signal. The *Precision* of the 25 files using 'repetition' process, as mentioned in Section 2.2.3 of Chapter 2, was 100%. This means that every watermark was successfully detected.

The PEAQ was used to evaluate the perceptual transparency of the scheme. The distribution of ODG scores is shown in Figure 3.23. From Figure 3.23, it can be seen that all of the ODG scores are consistently close to 0, meaning that the effect of watermarking is almost always imperceptible.

As can be seen from Figure 3.23, more than half of results from the PEAQ test were above 0. These anomalous results were therefore a cause of interest. A PEAQ test on five pairs of identical unwatermarked files was performed and these uniformly produced results above 0, which were same as the results shown in [CC07]. The resolution of the ODG is limited to one decimal. One should not expect that a difference between any pair of ODG of a tenth of a grade is significant [CC07]. Thus, some of our watermarked files were evaluated as being identical to their original counterparts. This was a positive outcome.

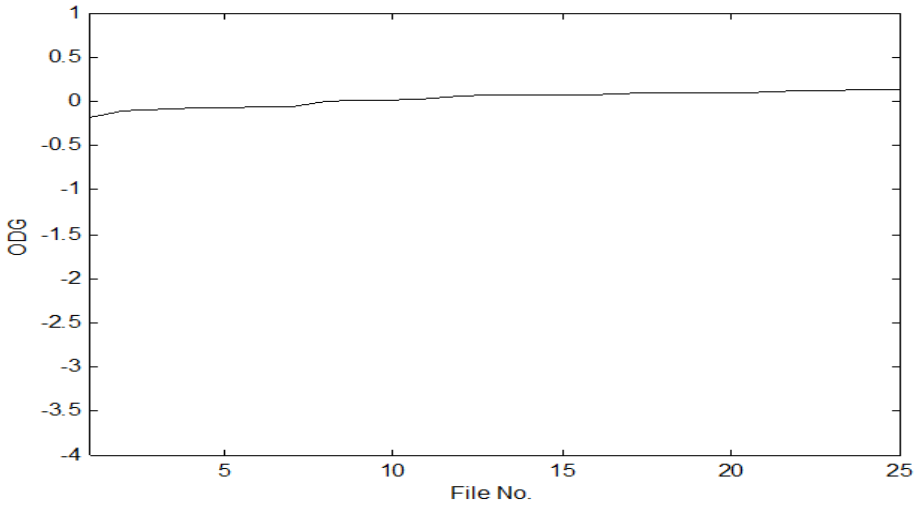


Figure 3.23 The distribution of ODG scores for 25 sample watermarked files

3.11 Summary

This chapter has provided a detailed explanation of the CSPE, and proposed an improvement that addressed a limitation of the original so that it is able to identify frequency components that do not appear over the entire frame under analysis. A CSPE based watermarking algorithm was then initially proposed using synthetic signals as host, and achieved a perfect accuracy. However, when this watermarking algorithm was applied to the real audio signals, the *Precision* was not satisfactory. Hence, further improvements were proposed such as incorporating a component verification process and an audible clicks removal process, which makes the algorithm extremely perceptual transparent and accurate. The next step was to assess its robustness against attacks and this will be considered in Chapter 4.

Chapter 4 robustness improvement of the Complex Spectral Phase Evolution based audio watermarking algorithm

4.1 Introduction

The audio watermarking algorithm based on the improved CSPE, proposed in Chapter 3, is extremely imperceptible and accurate. In this chapter, the robustness of this algorithm is investigated and found to be very low. This makes the algorithm unsuitable for those applications that have a high requirement for robustness such as copyright protection and the tracking of illicit distribution. Therefore, the aim of this chapter is to improve the robustness of this algorithm while maintaining an acceptable imperceptibility and accuracy. A series of experiments are then carried out to examine the improved algorithm's overall performance by measuring its accuracy, imperceptibility, robustness, capacity and computational efficiency.

4.2 Investigation on the robustness of the previous algorithm

As far as music is concerned, one of the most common attacks is MP3 compression, which has become the standard for transmission and storage of audio for both World Wide Web (WWW) and portable media applications [Pai00]. Robustness against MP3 compression has become a common benchmarking measure for audio watermarking [DSLZ04]. Thus, an investigation on whether the CSPE based audio watermarking algorithm proposed in Chapter 3 can survive MP3 64 kbps compression is carried out.

Twenty five music files were randomly selected from a variety of genres and then watermarked by the watermarking algorithm presented in the previous chapter. The robustness against MP3 64 kbps compression is shown in Figure 4.1 where the x-axis denotes the file number and the y-axis denotes the robustness. The upper panel in Figure 4.1 represents the *Precision* distribution without using the 'repetition' process. The lower panel in Figure 4.1 represents the *Precision* distribution after using the 'repetition' process. The 'repetition' process is defined in Section 2.2.3.

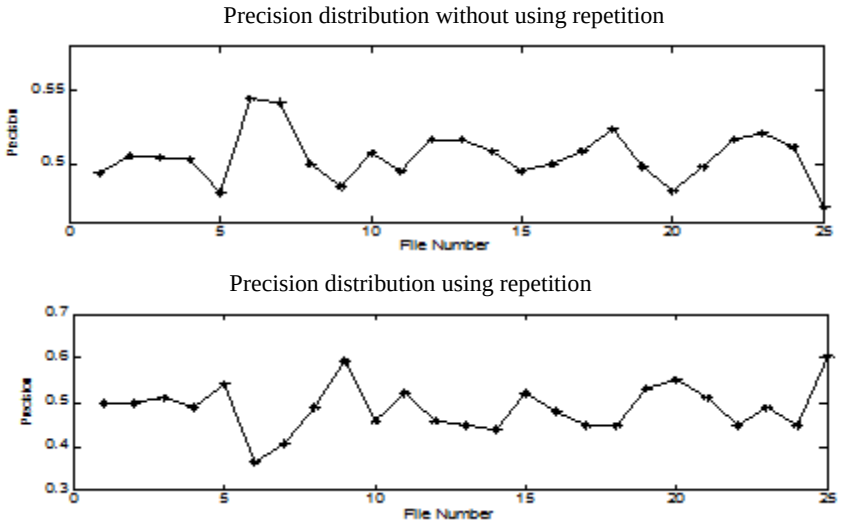


Figure 4.1 The *Precision* distribution after 64 kbps MP3 compression

It can be seen from the upper panel of Figure 4.1 that the *Precision* after MP3 64 kbps is very low, fluctuating around 50%. Furthermore, from the lower panel of Figure 4.1, it can be seen that there is no apparent improvement in the *Precision* after incorporating the 'repetition' process, as it still fluctuates around 50%. This percentage actually suggests that the detection result is similar to a random guessing. The reason for

this low robustness is that after undergoing MP3 compression, the signal spectrum has been changed to such an extent that the CSPE is unable to identify the same components at the detection stage, as the CSPE is very sensitive to spectral alterations. However, despite this low robustness being an apparent deficiency, it actually makes the algorithm potentially suitable as a fragile watermarking algorithm.

In order to improve the robustness of this algorithm, the improved CSPE is applied on an exemplar real audio signal, $x_0(n)$, to see if there are some intrinsic stable data in its CSPE spectrum. Denote $x_1(n)$ as the one-sample shifted version of $x_0(n)$. The magnitude of the CSPE spectrum M_{CSPE} is computed using Equation (4.1).

$$M_{CSPE} = \left\| F_{wx_0} F^*_{wx_1} \right\| \tag{4.1}$$

where F_{wx_0} is the windowed Fourier transform of $x_0(n)$, F_{wx_1} is the windowed Fourier transform of $x_1(n)$, '$*$' denotes the conjugation. The magnitude of the FFT spectrum M_{FFT} is computed using Equation (4.2).

$$M_{FFT} = \left\| F_{wx_0} \right\| \tag{4.2}$$

The spectra of the signal generated by CSPE and FFT respectively are shown in the upper panel and lower panel of Figure 4.2 respectively. The x-axis denotes the bin number and the y-axis denotes the magnitude of each bin. As seen from Figure 4.2, the peaks in the CSPE spectrum are more distinct than those in the FFT spectrum. Thus, it is reasonable to assume that manipulation on the peaks from the CSPE spectrum would

introduce less distortion than applying a similar manipulation on the peaks from the FFT spectrum. The reason for this is that the peak estimation by CSPE is more precise.

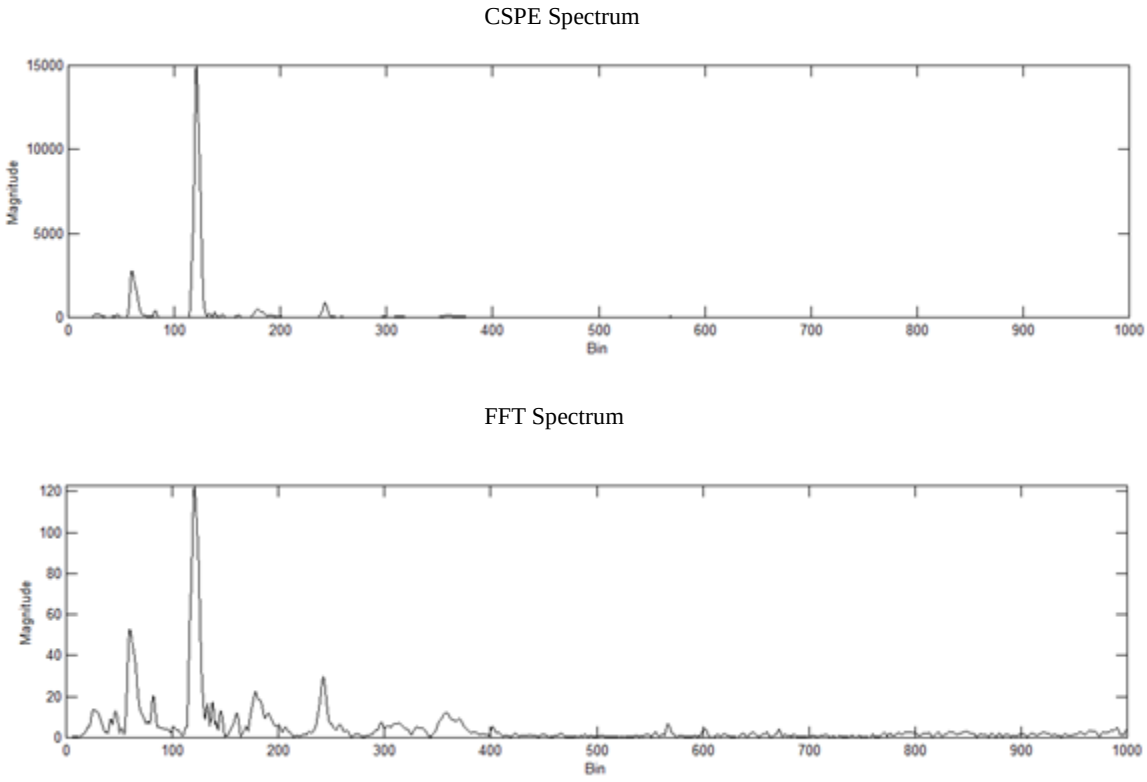


Figure 4.2 Magnitude spectrum using CSPE and FFT

In order to test its robustness, this exemplar signal undergoes a MP3 64 kbps compression and decompression process. The spectrum of the processed signal generated by the CSPE is shown in Figure 4.3. The x-axis denotes the bin number and the y-axis denotes the magnitude of each bin.

As shown in Figure 4.3, it can be seen that peaks of the original CSPE spectrum are retained after the compression-decompression process. The distinction and stability of

the peaks in the CSPE spectrum suggest that watermark information can be embedded by manipulating these peaks.

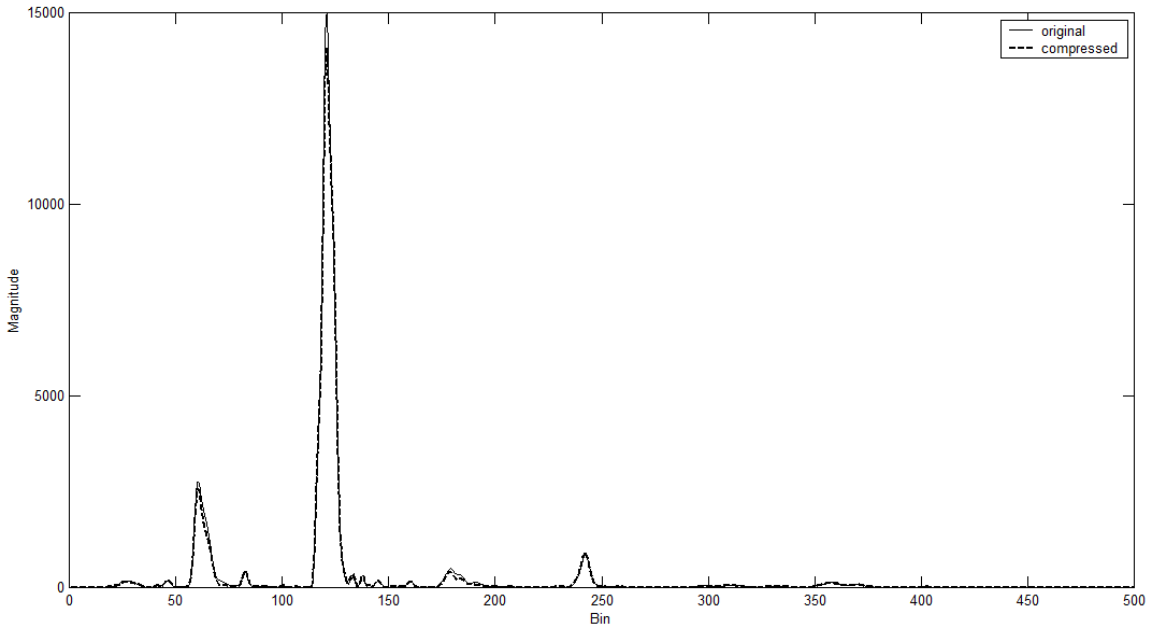


Figure 4.3 The consistency of CSPE spectrum peaks before and after 64 kbps MP3

4.3 Embedding watermark information by manipulating the peaks

As far as music is concerned, it contains a large number of frequency components ranging from low frequency to high frequency [ZF90]. Generally, components within a very high frequency range are at more risk of being removed by lossy compression or lowpass filtering. Thus, it is better to manipulate the peaks in the low frequency range (for example below 8000 Hz) in order to have a better chance of recovering the embedded information following different attacks [KS05]. Furthermore, the energy of the signal is often distributed non-uniformly across the frequency range. In general, the energy in high frequency range is lower than that in the low frequency range [ZF90].

Manipulation of a component from a high-energy region could introduce less audible distortion than manipulation of a component from the low-energy region. All these suggest that peaks in the low frequency range are more suitable for embedding the watermark bits in order to achieve a strong robustness and a high perceptual transparency.

4.3.1 Using *r* to select the candidate peak

Similar to the algorithm proposed in Chapter 3, a reference bin value *r* is used as the basis for selecting the candidate peak to manipulate.

4.3.2 Embedding rules

The embedding rules are defined as follows:

$$\begin{aligned} &\text{If } bit=1 \text{ let } |cbin\%2| = 1 \\ &\text{If } bit=0 \text{ let } |cbin\%2| = 0 \end{aligned} \tag{4.3}$$

where *cbin* denotes the bin location of the candidate peak, which is the first peak above *r* by more than *Th*. The embedding rules can be demonstrated in Figure 4.4.

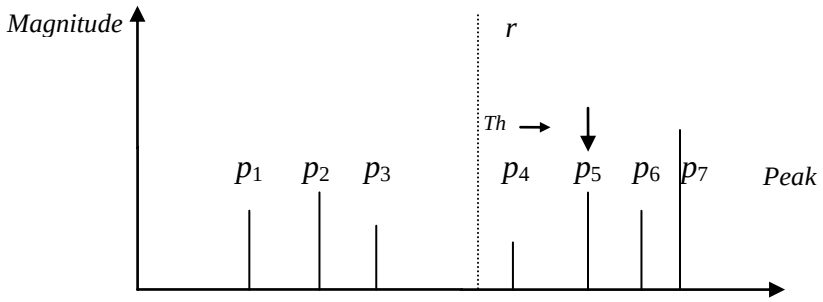


Figure 4.4 The demonstration of embedding rules

Assume p_1, p_2, p_3, p_4, p_5, p_6, p_7 are peaks on the CSPE spectrum. The p_5 would be selected as a candidate peak as it is the first peak that above *r* by more than *Th*. Assume its bin location is 37, that is, *cbin* is 37. If a bit '1' is to embed, Equation (4.3) is fulfilled. If a bit '0' is to embed, Equation (4.3) is not fulfilled and manipulation has to be applied.

4.3.3 Manipulation approach

If *cbin* does not satisfy Equation (4.3), manipulation has to be applied to satisfy Equation (4.3). The basic idea is to remove the candidate peak first, as shown in Equation (4.4),

$$x'(n) = x(n) + (-m_{cbin})cos\left(2 * \pi * \frac{f_{cbin}}{f_s} * n \ + \ \phi_{cbin}\right) \tag{4.4}$$

where f_{cbin}, m_{cbin} and ϕ_{cbin} is the frequency, magnitude and phase of the candidate peak respectively, $x'(n)$ is the manipulated signal.

Then, create a new frequency value f'_{cbin}, which is a small alteration of f_{cbin}, as shown in Equation (4.5), where *c* is a value used to alter the frequency value.

$$f'_{cbin} = (cbin + c) * \frac{f_s}{N} \tag{4.5}$$

Lastly, a new component with a frequency of f'_{cbin}, magnitude of m_{cbin}, and phase of ϕ_{cbin}, is then added to the signal, as shown in Equation (4.6). By this means, the bin location of the candidate peak might be changed to fulfill Equation (4.3).

$$x'(n) = x(n) + (m_{cbin})cos\left(2 * \pi * \frac{f'_{cbin}}{f_s} * n \ + \ \phi_{cbin}\right) \tag{4.6}$$

4.4 Using thresholds to select and manipulate the candidate peak

As seen from Figure 4.3, some peaks' magnitude values are too low for embedding as they can be distorted after signal processing. Thus, it is necessary to define a threshold value Th_a in selecting the candidate peak. In other words, the candidate peak's magnitude value should be above Th_a.

4.4.1 Issue with only using Th_a

However, as observed from the experiments, the magnitude of each peak in the CSPE spectrum may have changed after performing some normal processes, such as lossy compression. If the same threshold Th_a is used during the embedding and detection to select the candidate peak, it is assumed that after some signal processing, the peak selected at the detection stage might be different from that selected at the embedding stage. There are two scenarios making this assumption correct, which will be introduced in the next sections.

4.4.2 The first scenario that satisfies the assumption

The first scenario is that some peaks with a magnitude less than Th_a exist between the bin location of the candidate peak and r. These peaks are not selected at the embedding stage as their magnitudes are less than Th_a. However, some of these peaks could possibly have their magnitudes increased to be above Th_a by the process such as lossy compression, so that a new candidate peak will be selected at the detection stage. Figure 4.5 can be used to demonstrate this scenario. As can be seen from the figure, p_7 will be selected at the embedding stage as it is the first peak above r by more than Th, and

with a magnitude of above Th_a. However, after signal processing such as MP3 compression, the magnitude of peak p_5 can be increased to be above Th_a, which results in the peak p_5 being likely selected at the detection stage, as shown in Figure 4.6. Consequently, the *Precision* of the watermark detection will be reduced.

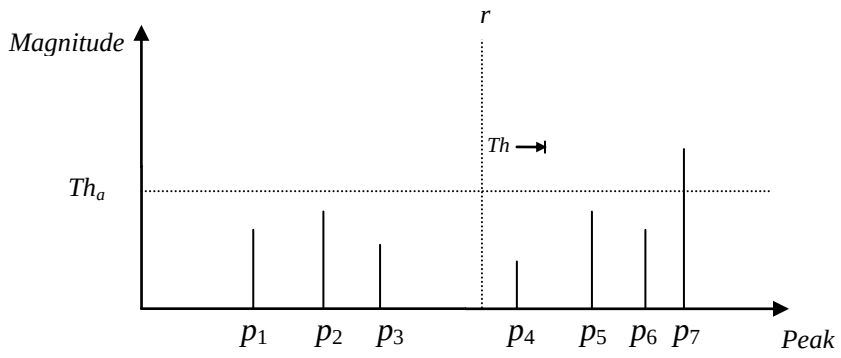


Figure 4.5 The demonstration of the peak being selected before MP3 compression

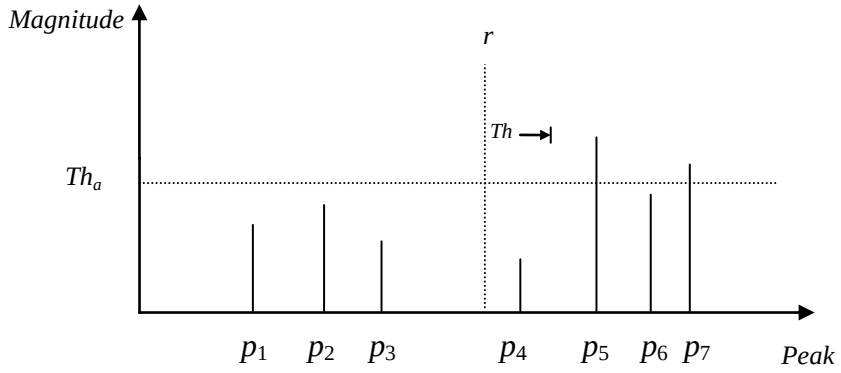


Figure 4.6 The demonstration of the peak being selected after MP3 compression

4.4.3 The second scenario that satisfies the assumption

The second scenario is that the magnitude of the candidate peak could be reduced to below Th_a by the process such as lossy compression so that it cannot be identified at

the detection stage. As can be seen from the Figure 4.7, the peak p_5 will be selected at the embedding stage. However, after signal processing such as MP3 compression, its magnitude could be reduced to be below Th_a, which results in the peak p_6 being likely selected at the detection stage, as shown in Figure 4.8. Consequently, the *Precision* of the watermark detection will be reduced.

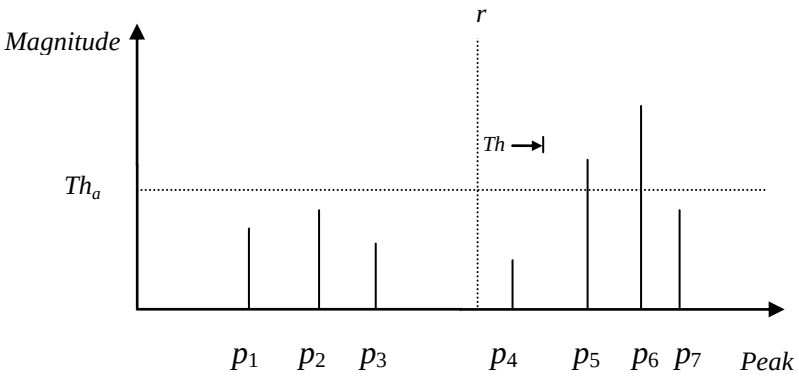


Figure 4.7 The demonstration of the peak being selected before MP3 compression

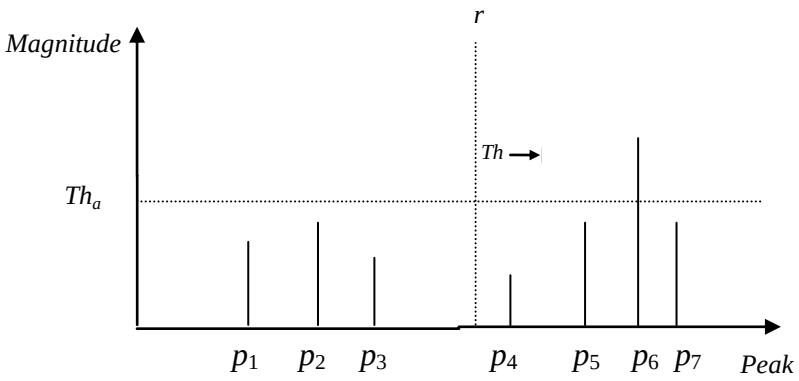


Figure 4.8 The demonstration of the peak being selected after MP3 compression

4.4.4 The solution to avoid the two scenarios

These two scenarios have to be dealt with in order to achieve a strong robustness. A solution to this problem is proposed as follows. Three different thresholds Th_a, Th_b, Th_c are defined. The thresholds Th_a and Th_b are used at the embedding stage while Th_c is used at the detection stage.

Using a Th_c that satisfies Equation (4.7) can guarantee that those peaks between r and the candidate peak cannot have their magnitude increased to be equal to or greater than Th_c after signal processing. Thus, these peaks cannot be selected at the detection stage, which avoids the first scenario.

$$Th_c = Th_a + \tau_1 \tag{4.7}$$

where τ_1 denotes the magnitude variation caused by a signal processing operation such as MP3 compression.

In addition, Th_b has to be greater than Th_c to an extent, to guarantee that the altered magnitude of the candidate peak remains equal to or greater than Th_c after signal processing such as lossy compression. This can help to avoid the second scenario.

4.5 The embedding procedure

The procedure to embed the watermark is depicted in Figure 4.9, and can be described as follows:

1. Define the value of r and initialize the counter variable *cnt*. The *cnt* controls the number of iterations of the algorithm and by experiment an initial value of 20 was found to achieve a good trade-off between computational efficiency and performance.

2. Apply the improved CSPE to analyze the signal on a frame-by-frame basis and derive the corresponding CSPE spectrum.
3. Find a candidate peak with its bin location *cbin* closest on the right to *r* and with its magnitude above the threshold Th_a.
4. Check if the bin location of the candidate peak satisfies Equation (4.3). If Equation (4.3) is not satisfied, go to step 5. Otherwise, go to step 6.
5. Obtain the magnitude and phase of the candidate peak according to Equation (3.5) and (3.6) respectively. Then, manipulate this candidate peak according to Equations (4.4-4.6). In Equation (4.5), *c* is set to be *cnt**0.1. The reason for using 0.1 is that it introduces a minor frequency shift which is important for achieving a high imperceptibility.
6. Decrease *cnt* by 1 and check whether *cnt* is equal to 0. If it is, finish the entire process. Otherwise, check if the candidate peak's magnitude is above Th_b or not. If it is above Th_b, then finish the entire process. If not, go to step 7.
7. Increase the peak's magnitude to Th_b by Equation (4.8) and repeat from step 3.

$$x'(n) = x(n) + (-m_{cbin} + Th_b)cos\left(2 * \pi * \frac{f'_{cbin}}{f_s} * n + \phi_{cbin}\right) \quad (4.8)$$

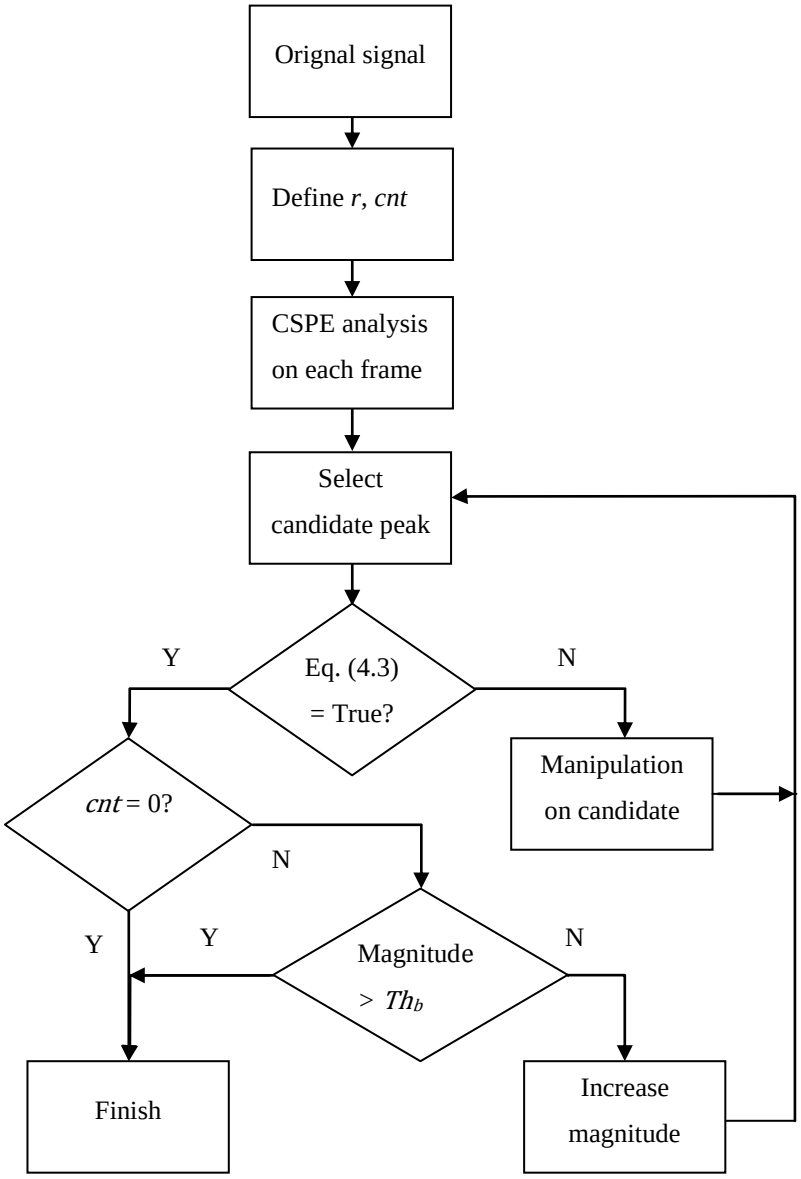


Figure 4.9 The embedding procedure

4.6 The detection procedure

The basic procedure to detect the embedded watermark is described as follows:

1. Apply the improved CSPE to the signal on a frame-by-frame basis and derive its CSPE spectrum.

2. Use r along with threshold Th_c to identify the candidate peak.
3. If the parity of the candidate peak's bin location is 0, as per Equation (4.3), the bit to be detected is a '0'. Otherwise, the bit to be detected is a '1'.

4.7 Investigation on the impact of the parameters values on performance

Experiments were conducted to analyze the impact of the threshold values and the reference value on the performance of watermarking algorithm.

4.7.1 Questions to be answered

In general, an increase in Th_b would result in a higher robustness because the candidate peak becomes more significant. However, an increase in Th_b would result in a worsened imperceptibility because the candidate peak becomes more perceptually obtrusive. The value of Th_b should be defined dynamically according to the perceptual properties of each frame of each specific signal. In addition, Th_c can be determined if Th_a and τ_1 are known in terms of Equation (4.7). Thus, it is only necessary to analyze the impact of three parameters, r , Th_a and τ_1 on the watermarking performance.

In order to investigate how the value r affects the performance, r was set from 500 to 4000, which corresponds to frequency value in Hz from 1465 to 11718. The conversion from bin value to frequency value in Hz can be derived from Equation (4.9). For this experiment, the sampling frequency = 48000 and the window length = 16384.

$$f = \frac{r * f_s}{N} \tag{4.9}$$

As stated previously, any component that lies at a frequency above 8000 Hz is at risk of being removed by signal processing such as lossy compression [KS05]. Thus, this range of r is sufficient for the investigation.

In order to investigate how the value τ_1 affects the performance, τ_1 was set to be 5 and 10 respectively. The reason for this setting was based on an experimental finding, that is, any variation in the magnitude of peaks after signal processing, such as MP3 compression, generally lies in this range. Note that all the signals under analysis are normalized so that their amplitudes lie from -1 to 1. However, the spectrum is not normalized with respect to the frame length in order to avoid using small decimal numbers.

In order to investigate how the value Th_a affects the performance, Th_a was set from 5 to 50. The purpose of this setting is to investigate if there is any trend in how Th_a affects the performance. As for Th_b, in this experiment, the difference between Th_b and Th_a for all the test music files is set as a constant value. The purpose of doing this is to remove the impact of Th_b on the performance analysis. Since Th_b has to be greater than Th_c to an extent, while Th_c was already set to be greater than Th_a by 5 or 10, thus, the difference between Th_b and Th_a was set as 25.

40 groups were created and each group has a different combination of value Th_a, Th_b, Th_c and r, as listed in Table 4.1. The 20 music tracks were randomly selected with multiple genres. Each track was sampled at 48000 Hz. The frame size used was 8192 with 8192 samples of zero padding. 40 watermark bit sequences were randomly generated with a length of 480 bits. The initial value of *cnt* was set to be 20.

Table 4.1 Threshold and *r* setting of each different group

Group	*r*	Th_a	Th_b	Th_c	Group	*r*	Th_a	Th_b	Th_c
1	500	5	30	10	**21**	500	5	30	15
2	1000	5	30	10	**22**	1000	5	30	15
3	2000	5	30	10	**23**	2000	5	30	15
4	3000	5	30	10	**24**	3000	5	30	15
5	4000	5	30	10	**25**	4000	5	30	15
6	500	10	35	15	**26**	500	10	35	20
7	1000	10	35	15	**27**	1000	10	35	20
8	2000	10	35	15	**28**	2000	10	35	20
9	3000	10	35	15	**29**	3000	10	35	20
10	4000	10	35	15	**30**	4000	10	35	20
11	500	20	45	25	**31**	500	20	45	30
12	1000	20	45	25	**32**	1000	20	45	30
13	2000	20	45	25	**33**	2000	20	45	30
14	3000	20	45	25	**34**	3000	20	45	30
15	4000	20	45	25	**35**	4000	20	45	30
16	500	50	75	55	**36**	500	50	75	60
17	1000	50	75	55	**37**	1000	50	75	60
18	2000	50	75	55	**38**	2000	50	75	60
19	3000	50	75	55	**39**	3000	50	75	60
20	4000	50	75	55	**40**	4000	50	75	60

4.7.2 Accuracy of the proposed algorithm

Figure 4.10 describes the *Precision* of each group without undergoing any attack. Note that there was no difference between the *Precision* of the first 20 groups and the second 20 groups, because no attack was applied. Thus, only the *Precision* of the first 20 groups were plotted. For clarity, the *Precision* of each five files is shown in each panel of Figure 4.10. The x-axis denotes the Group number given in Table 4.1, and the y-axis denotes the *Precision*.

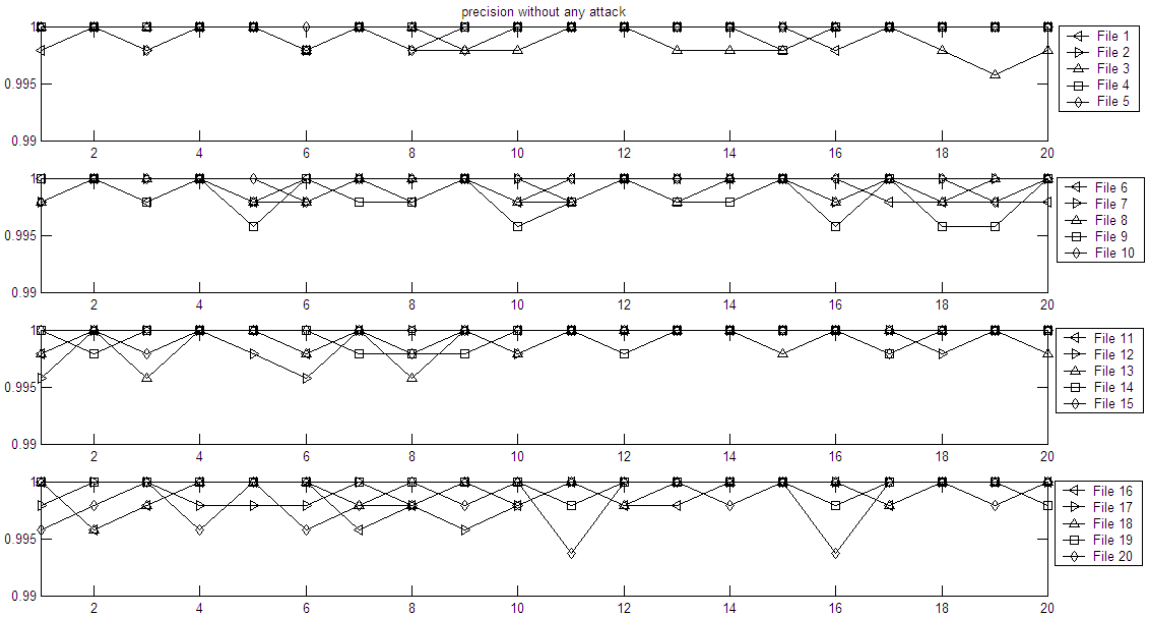


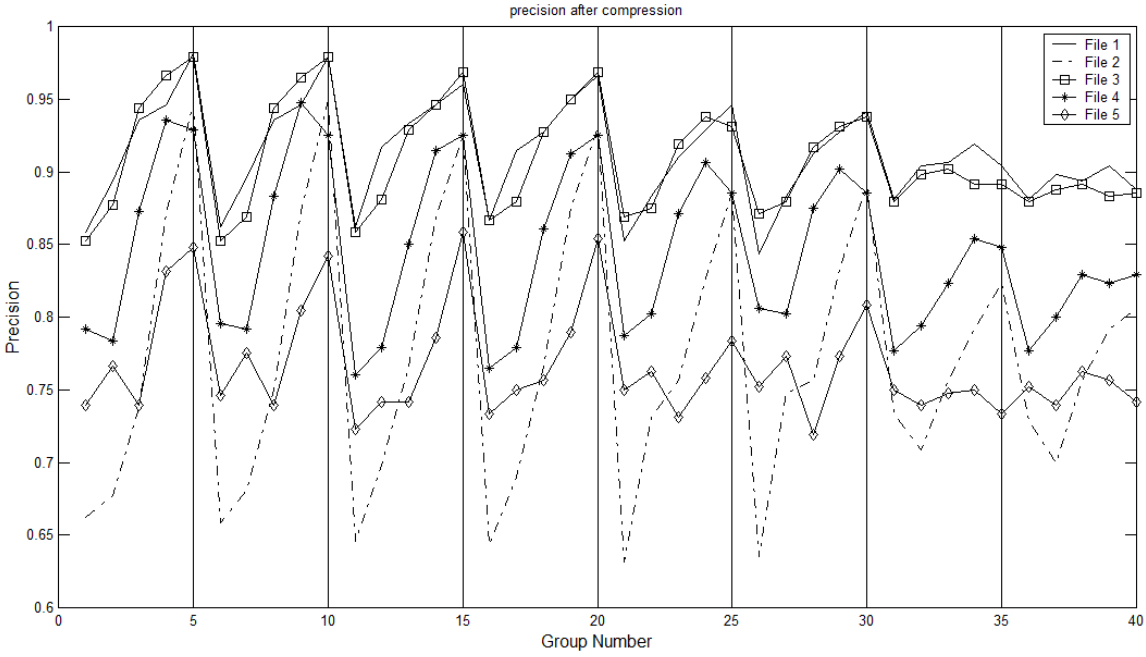
Figure 4.10 The *Precision* distribution without attack for 20 audio files with the parameters given by Group 1-20 from Table 4.1

It can be seen from Figure 4.10 that the *Precision* is almost 100% for all files. Some bit errors appear because an initial value of *cnt* must be set, and this limits the number of iterations. As a result, the embedding procedure could stop when *cnt* is 0

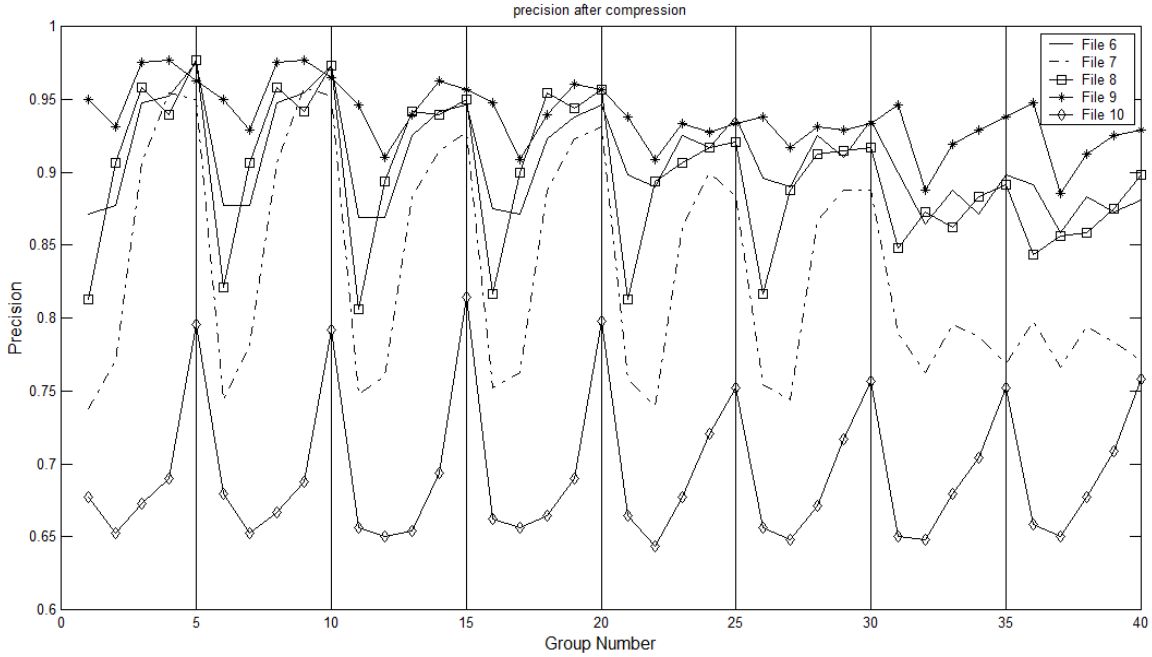
without fulfilling Equation (4.3). Increasing the initial value of *cnt* might help to improve the *Precision*, but results in a higher computational cost.

4.7.3 Robustness against MP3 attack

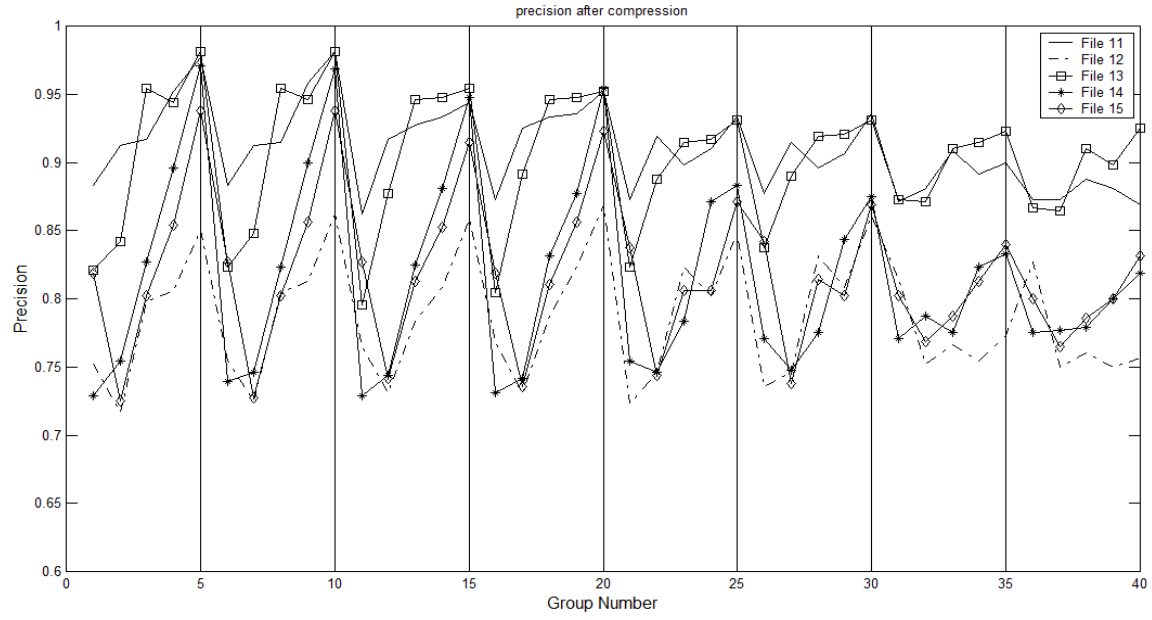
Robustness of the 20 audio files was tested against MP3 64 kbps and the result is depicted in Figure 4.11 (a)-(d). The x-axis denotes the Group number given in Table 4.1, and the y-axis denotes the *Precision*.



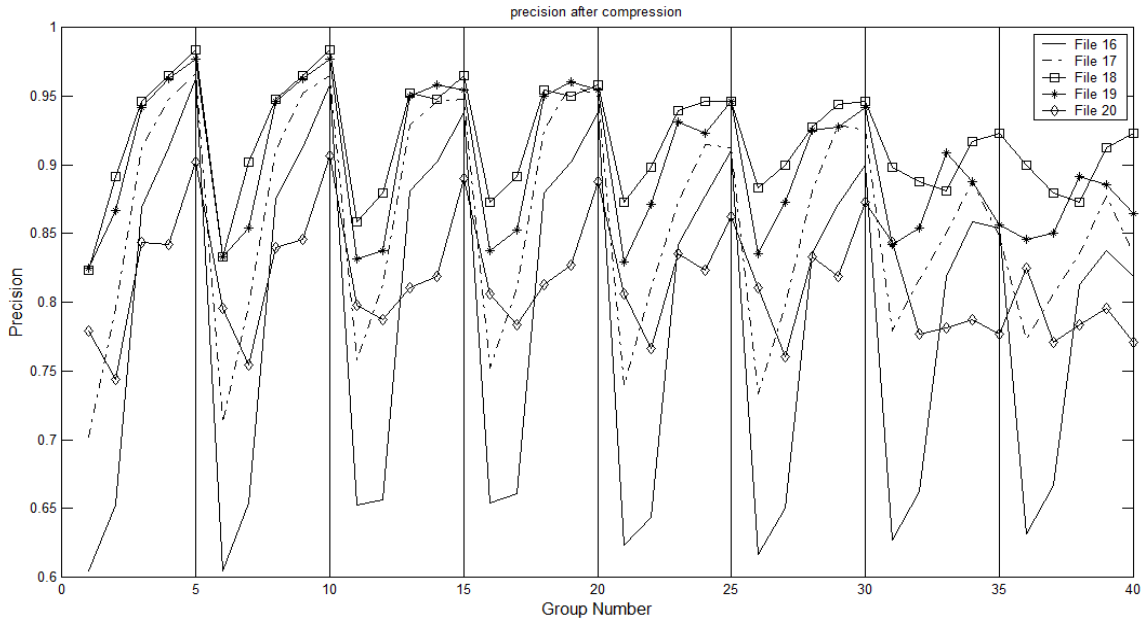
(a) The precision distribution of File 1-5 after 64 kbps MP3 compression



(b) The precision distribution of File 6-10 after 64 kbps MP3 compression



(a) The precision distribution of File 11-15 after 64 kbps MP3 compression



(d) The precision distribution of File 16-20 after 64 kbps MP3 compression

Figure 4.11 (*a* to *d*): The *Precision* distribution after 64 kbps MP3 attack for 20 audio files with the parameters given by Group 1-40 from Table 4.1

From Figure 4.11, it can be seen that, generally, with an increase in r, the robustness of each file increases. This can be observed within each successive five groups. For example, the robustness increases from Group 1 to Group 5, or from Group 6 to Group 10.

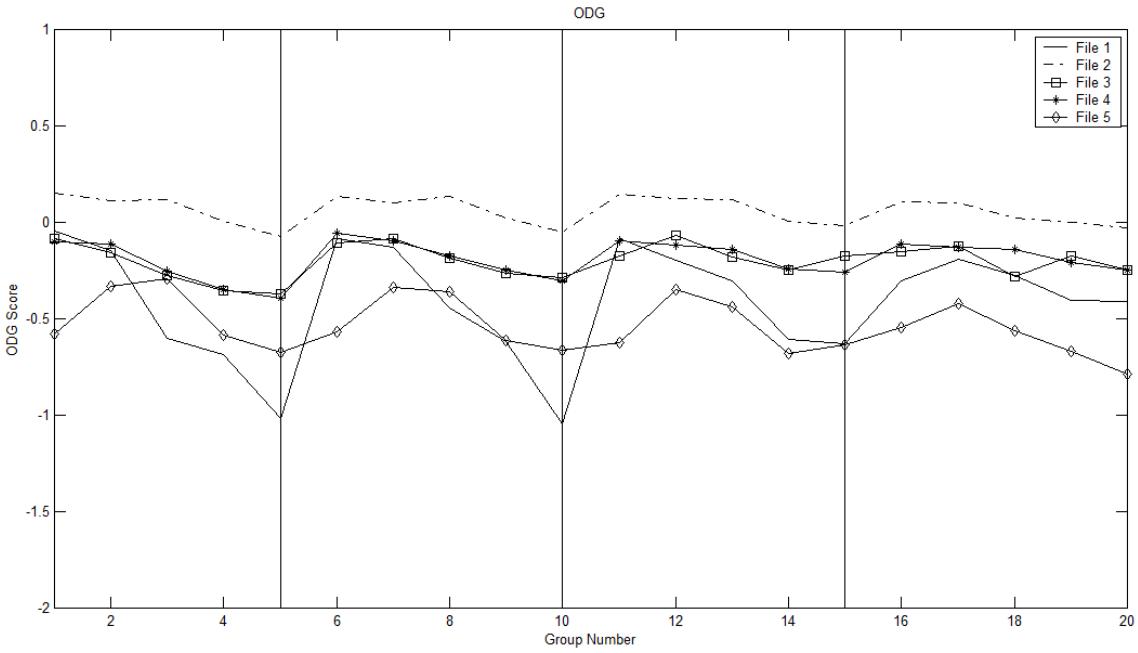
It also can be seen that there is no major difference in the robustness when using different values of Th_a. This can be observed by comparing those groups whose only difference is the value of Th_a. For example, between Group 1, Group 6, Group 11 and Group 16, or between Group 2, Group 7, Group 12 and Group 17 in each panel.

Furthermore, it can be seen that the value of τ_1 has an impact on the *Precision*, which can be observed by comparing those Groups whose only difference is the value of τ_1. For example, between Group 1 and Group 21, Group 2 and Group 22. In general, the robustness decreases when the value of τ_1 decreases.

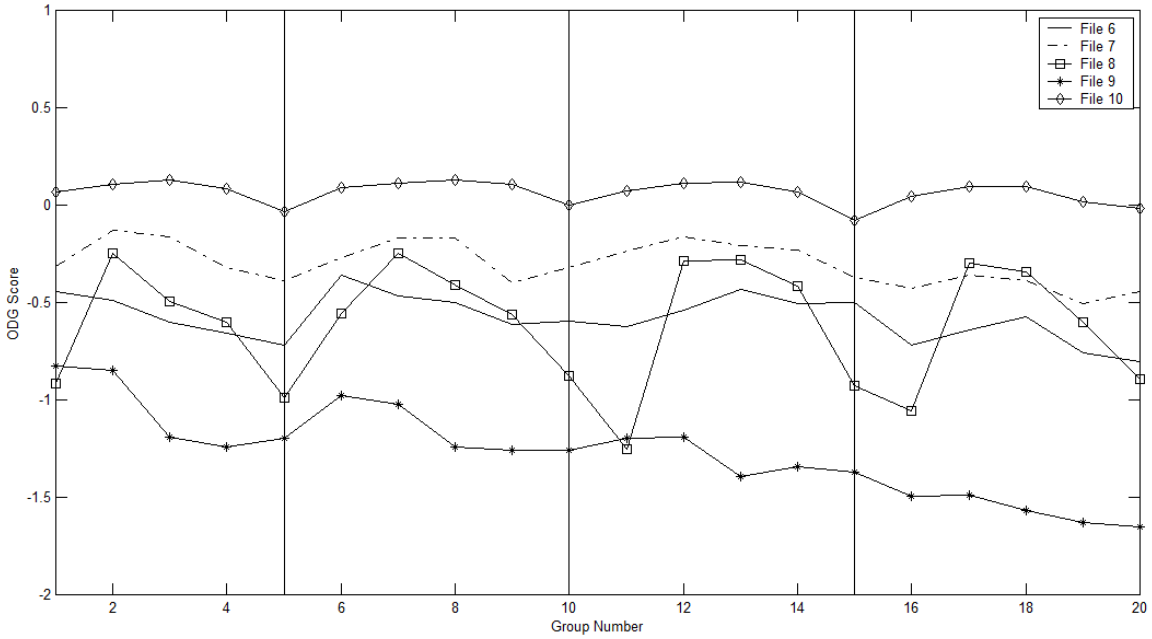
4.7.4 Imperceptibility

The PEAQ was used to test the imperceptibility. As the value of τ_1 has no impact on the imperceptibility, thus only the first 20 groups were examined.

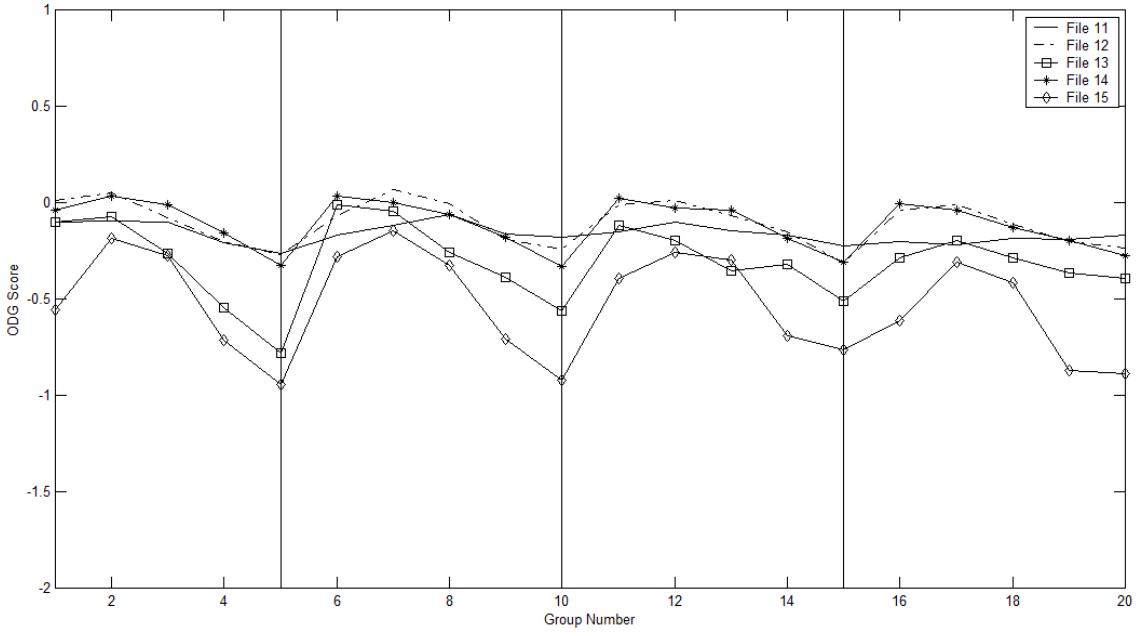
The ODG score distribution for the 20 audio files is shown in Figure 4.12 (a)-(d), where the x-axis denotes the group number and the y-axis denotes the ODG score. The vertical line in each panel is used to mark the boundary between every five groups.



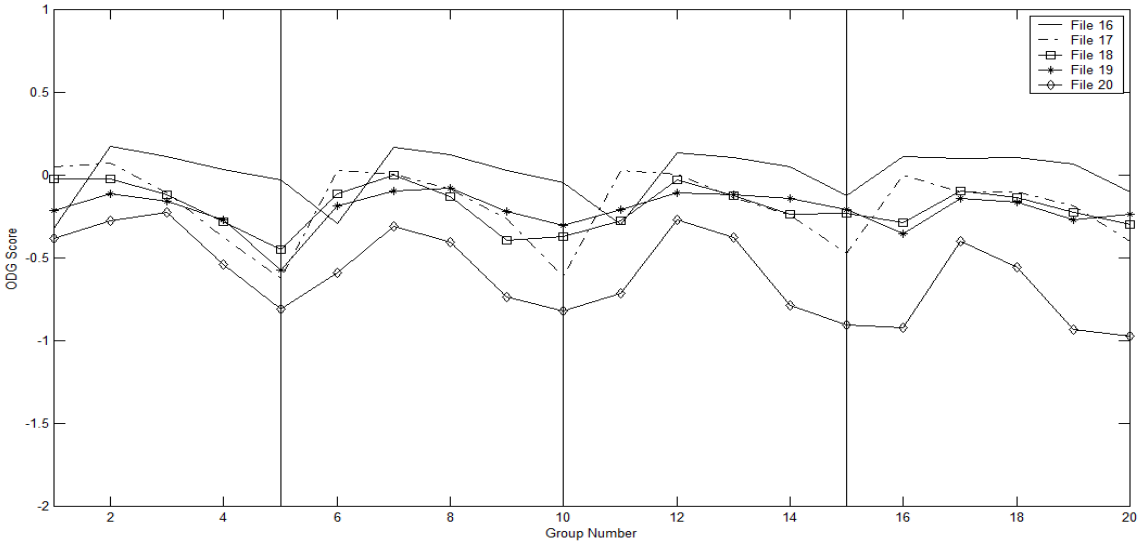
(a) The ODG score distribution of File 1-5



(b) The ODG score distribution of File 6-10



(c) The ODG score distribution of File 11-15



(d) The ODG score distribution of File 16-20

Figure 4.12 The ODG score distribution without attack for 20 audio files with the parameters given by Group 1-20 from Table 4.1

As can be seen from Figure 4.12 (a)-(d), the ODG score generally becomes worse as r increases. However, this is dependent on each specific music track. In addition, there is no major difference in the ODG score when using different Th_a values.

4.7.5 Brief summary

In general, the higher the value of r the robustness increases but the imperceptibility decreases. There is no major difference in the performance when using different values of Th_a. This feature allows the use of Th_a as a private key to improve the security of the algorithm. Furthermore, an increase in τ_1 would decrease the robustness, but this does not affect the imperceptibility.

4.8 Using a peak sharpness measure to improve the robustness

A reason for the watermark detection errors after MP3 64 kbps compression was sought. It was found that many errors were due to a particular phenomenon, that is, the bin location of the candidate peak identified at the embedding stage was shifted by one bin at the detection stage following the attacks. This means that the bit detected is actually the inverse of the true value because the bit detected is determined by the parity of the identified bin location, thus reducing the robustness.

4.8.1 The probability of one bin shift phenomenon

The occurrence probability of this phenomenon is calculated as the ratio of bit errors due to one bin shift to the total number of bit errors. The distribution of the occurrence probability of this phenomenon for each file in Group 3 is shown in Figure 4.13, where the x-axis denotes the file number and the y-axis denotes the probability. As can be seen from the Figure 4.13, the probability is higher than 50% on average. Thus, solving this one bin shift phenomenon can improve the robustness.

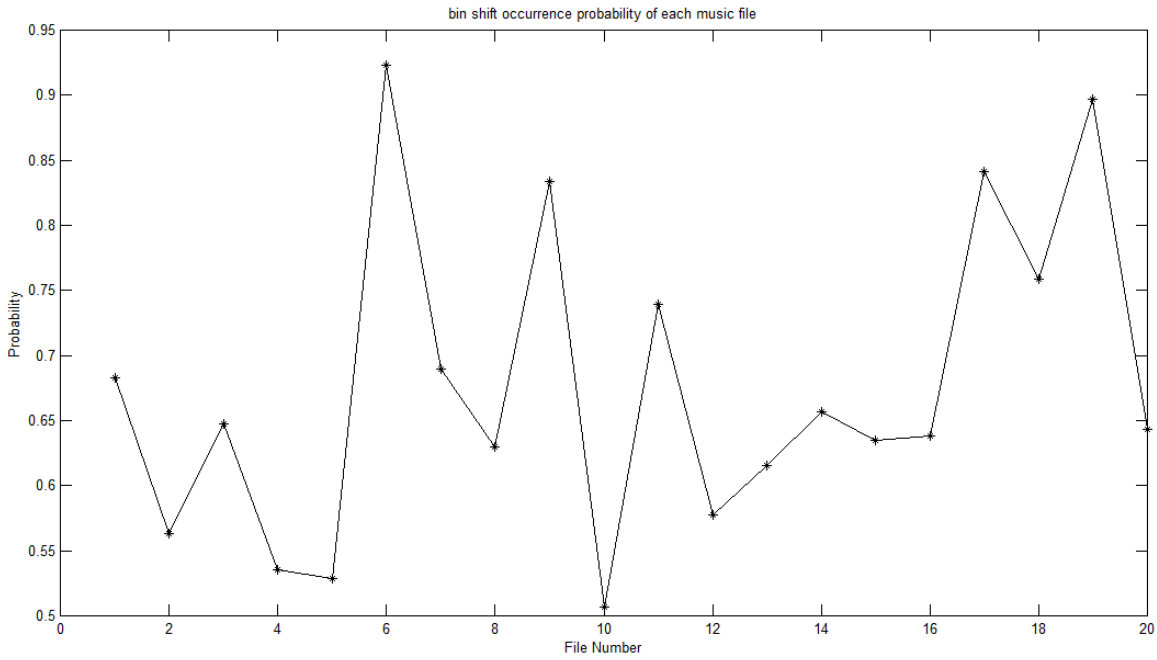


Figure 4.13 The probability distribution of the bin shift phenomenon for 20 audio files

4.8.2 The reason for this one bin shift phenomenon

The reason for this one bin shift is that the magnitude of the candidate peak is not significantly higher than the magnitudes of its two neighbouring bins. This results in the spectrum having a very smooth peak instead of a sharp one. As a result, following an attack such as MP3 compression, the magnitudes of the neighbouring bins could be greater than that of the original candidate peak, which results in a new peak being found at the detection stage. In Figure 4.14, the spectrum of one frame of a music file was used as an example to illustrate this phenomenon, where the x-axis denotes the bin number and the y-axis denotes the magnitude. As shown in Figure 4.14, the bin location of the candidate peak identified before compression is 1916. While after compression process, the bin location of the candidate peak identified is 1915, thus one bin shift has occurred.

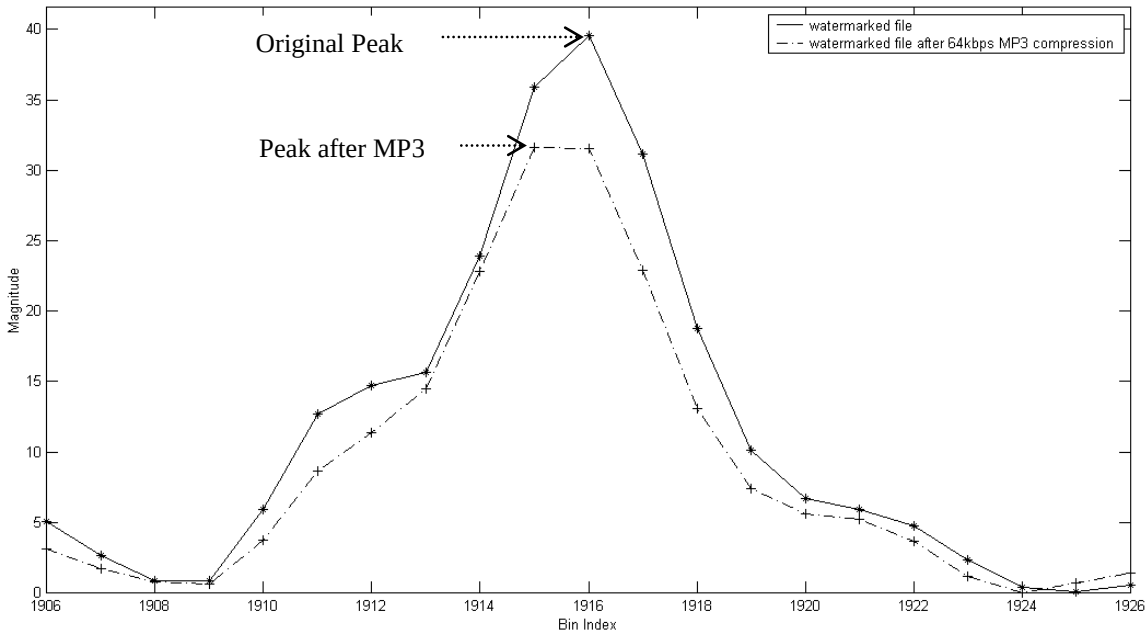


Figure 4.14 The demonstration of bin shift phenomenon

4.8.3 The solution to this one bin shift phenomenon

In order to reduce the occurrence probability of this phenomenon, a new set of constraints, were added to the embedding procedure. The aim of these constraints was to guarantee that the magnitude of the candidate peak is bigger than its neighbouring bins' magnitudes to a certain extent, as shown by Equations (4.10) and (4.11).

$$\frac{m_{cbin}}{m_{cbin-1}} > Th_d \tag{4.10}$$

$$\frac{m_{cbin}}{m_{cbin+1}} > Th_d \tag{4.11}$$

where Th_d denotes a threshold, m_{cbin-1} denotes the magnitude of the bin left to the candidate peak, and m_{cbin+1} denotes the magnitude of the bin right to the candidate peak. Thus, before finalizing the candidate peak, its magnitude will be verified according to

Equation (4.10) and (4.11). If Equation (4.10) and (4.11) are not fulfilled, the iteration continues until they are met. This added process is called 'peak sharpness measurement'.

4.8.4 Experimental validation

An experiment on the same 20 music files as that in Section 4.7.1 was performed to verify if the sharpness measure can improve the robustness. The threshold Th_d was set as three different values 1.1, 1.3, 1.5 while *r* was 2000 and Th_a, Th_b, Th_c were 5, 30 and 10 respectively. The *Precision* of each file after MP3 64 kbps is shown in Figure 4.15, where the x-axis denotes the file number and the y-axis denotes the *Precision*. As can be seen from the Figure 4.15, the *Precision* achieved is the best when Th_d is 1.1. More specifically, the individual improvement ranges from 2% to 8% and the average improvement is 4%. The $Precision_{mean}$, as per Equation (2.5), is 91.13% after incorporating the sharpness measure, while it is 87.75% without using this measure.

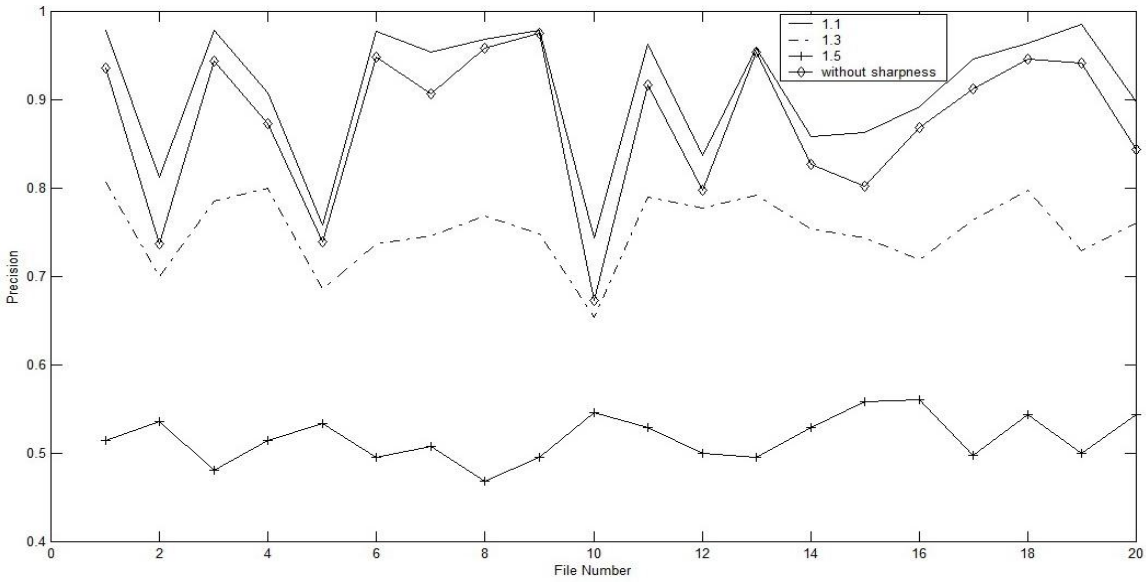


Figure 4.15 The *Precision* after using sharpness process

The ODG score of the algorithm with and without incorporating the sharpness measure were compared on the same 20 music files. It was found that there is degradation in the ODG score when using the sharpness process. The mean ODG score of the 20 music files without using sharpness is -0.2425, while it is -0.3447 after using sharpness. It is worth noting that both ODG scores are acceptable. Therefore, it is worth including this process to increase the robustness.

4.9 Using an error-correction scheme to improve the robustness

Imperceptibility is crucial for an audio watermarking algorithm, thus, a process that can increase the robustness without degrading the imperceptibility is desirable. Error correction schemes can be included for this purpose and have been employed widely in audio watermarking algorithms [DP09, MRF10, TS01, WYWQ10]. The only disadvantage of these schemes is that they decrease the capacity. However, compared with the imperceptibility and robustness, capacity is less crucial. Some different error correction schemes such as 'Reed-Solomon' and 'repetition', which was mentioned in Section 2.2.3 of Chapter 2, will be investigated to verify their ability to improve the robustness.

4.9.1 Using Reed-Solomon coding to improve the robustness

In 1960, I. Reed and G. Solomon developed Reed-Solomon coding (RS) by using error detection and correction codes. This means that extra code symbols are added to the message to provide the necessary error detection and correction functionalities [Gei90, RS60]. Each symbol is made up with m bits. A RS code can be described as an [n, k]

tuple, where *n* represents the encoded message length in symbols, and *k* represents the original message length in symbols, thus (*n*-*k*) represents the redundancy message length in symbols. For example, the RS [31, 11] means the original message length is 11 symbols and the redundancy message length is 31-11 = 20 symbols. The mathematical relation between *n* and *m* is described as Equation (4.12),

$$n = 2^m - 1 \tag{4.12}$$

An experiment was carried out to test the error correction performance of a RS error correcting code after the signal undergoes the MP3 compression and decompression process. The procedure of incorporating the RS error correcting code is described as below:

1. Generate the watermark bit sequence B_w
2. Use RS to encode B_w and generate the encoded bit sequence B_e
3. Use the proposed watermarking algorithm to embed B_e into the signal
4. Perform 64 kbps MP3 compression and decompression on the watermarked signal
5. Detect the encoded watermark bit sequence B'_e from the watermarked signal
6. Using RS to correct B'_e and get the detected watermark bit sequence B'_w

Three settings of [*n*, *k*] were used: [31,11], [63,19] and [15,7], which offer different error correction capability. In this experiment, the sharpness measure was not incorporated. In this experiment, the same twenty files as that in Section 4.7.1 were used for test. *r* was 2000 and Th_a, Th_b and Th_c were 5, 30 and 10 respectively. Figure 4.16 shows the

Precision before and after incorporating RS coding respectively, where the x-axis denotes the file number and the y-axis denotes the *Precision*. Each sub panel in Figure 4.16 denotes the *Precision* after using each RS coding with different values for [*n*, *k*].

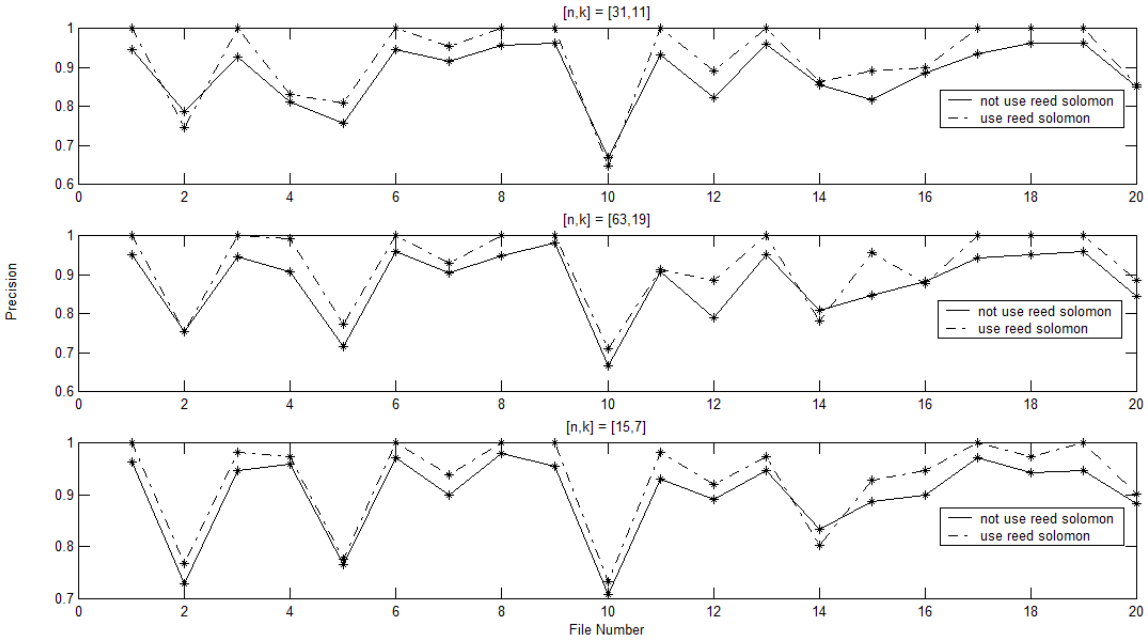


Figure 4.16 The *Precision* after using Reed-Solomon

From Figure 4.16, it can be seen that RS error correction did not achieve much improvement in the robustness. In order to find the reason for this, the experimental result when using RS [63,19] was examined as shown in the Table 4.2. For this RS correcting code, n = 63 and k = 19, thus, m = 5 as per Equation (4.12). That is, each symbol is made up with 5 bits. The length of total message is 378 bits. In Table 4.2, the number of 'Bit errors' and 'Symbol errors' were calculated, the maximum number of symbol errors that can be corrected is $\frac{n-k}{2}$ [Gei90]. Thus, for RS [63, 19], a maximum of 22 'Symbol errors' can be corrected.

Table 4.2 Bit and Symbol errors distribution for each file

File	Bit Errors	Symbol errors
1	19	15
2	93	51
3	21	17
4	35	24
5	108	55
6	16	16
7	36	29
8	20	16
9	7	6
10	126	57
11	35	26
12	80	45
13	19	16
14	73	44
15	58	33
16	45	33
17	22	20
18	19	15
19	16	13
20	59	41

It can be seen from the table that the number of ‘Bit errors’ for most files are similar to that of ‘Symbol errors’, which means that error bits in these files were spread across many symbols. Thus, the error bits in each of these files cannot be termed as ‘burst error’, because a ‘burst error’ means that the error bits are within a cluster of adjacent bits [Bos86]. This is the reason for poor error correction performance by RS coding because it

is only good at correcting 'burst error' [Gei90, RS60]. Thus, a different error correction scheme should be examined, as explained below.

4.9.2 Using Repetition to improve the robustness

Considering the nature of the 'Bit errors' distribution, the 'repetition' process can be added to improve the robustness, as defined in Section 2.2.3. In order to observe the impact of the number of 'repetition' times *d* on the error correction capability, an experiment was performed on the same twenty files as that in Section 4.7.1 where *d* was set to 1, 3, 5, 7, and 9 respectively. In this experiment, *r* was 2500, Th_a, Th_b and Th_c were 5, 30 and 10 respectively. Firstly, 20 individual bit sequences were randomly generated with a length of 120 bits. Then, each sequence was repetitively embedded into each music file *d* times. For example, if *d* is 1, only embed 120 bits. If *d* is 9, the 120 bits will be embedded repetitively 9 times (a total of 1080 bits).

The robustness against 64 kbps MP3 compression-decompression after incorporating the 'repetition' process is shown in Figure 4.17 where the x-axis denotes the file number and the y-axis denotes the *Precision*. It can be seen that *Precision* has been largely improved and when *d* is 9 the *Precision* of each file is very close to 100%. One point worth noting that the *Precision* of File 10 has been increased to nearly 100% when *r* was set to 2500, while it is only about 70% when *r* was set to 2000. This suggests that the value of *r* should not be defined uniformly for different music files. In fact, the value *r* should be defined dynamically according to each frame's spectrum from each signal.

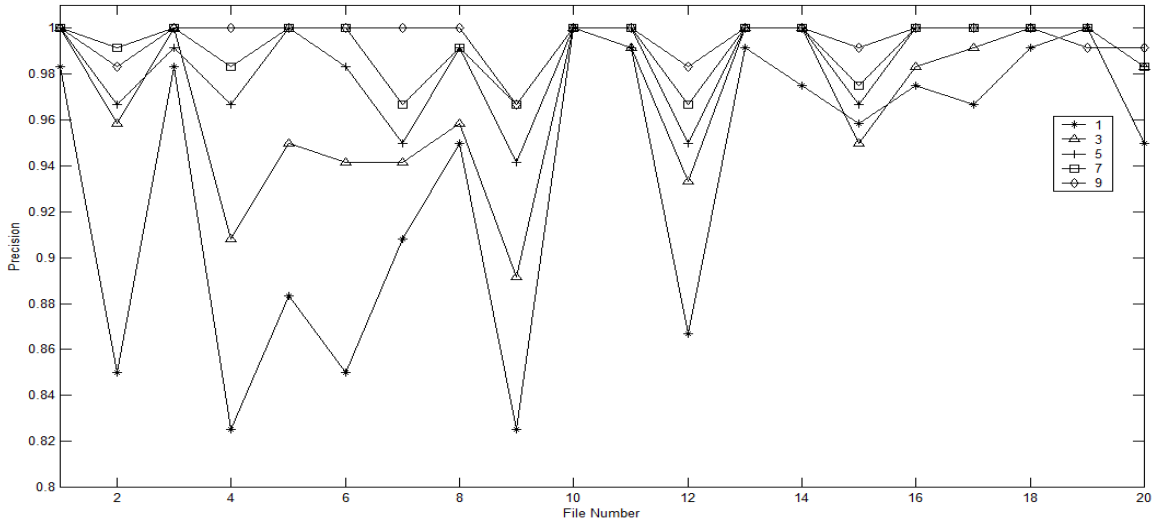


Figure 4.17 The distribution of *Precision* after MP3 64 kbps when using different values for the repetition *d*

The distribution of $Precision_{mean}$ for the 20 music files when the repetition time *d* varies from 1 to 9 is depicted in Figure 4.18, where the x-axis denotes the repetition time and the y-axis denotes the $Precision_{mean}$. It can be seen clearly that the $Precision_{mean}$ increases when *d* increases. As mentioned previously, there is a trade-off between the robustness and the capacity when using the 'repetition' process. For example, when *d* is 9, the capacity will be decreased to be 1/9 of that when *d* is 1.

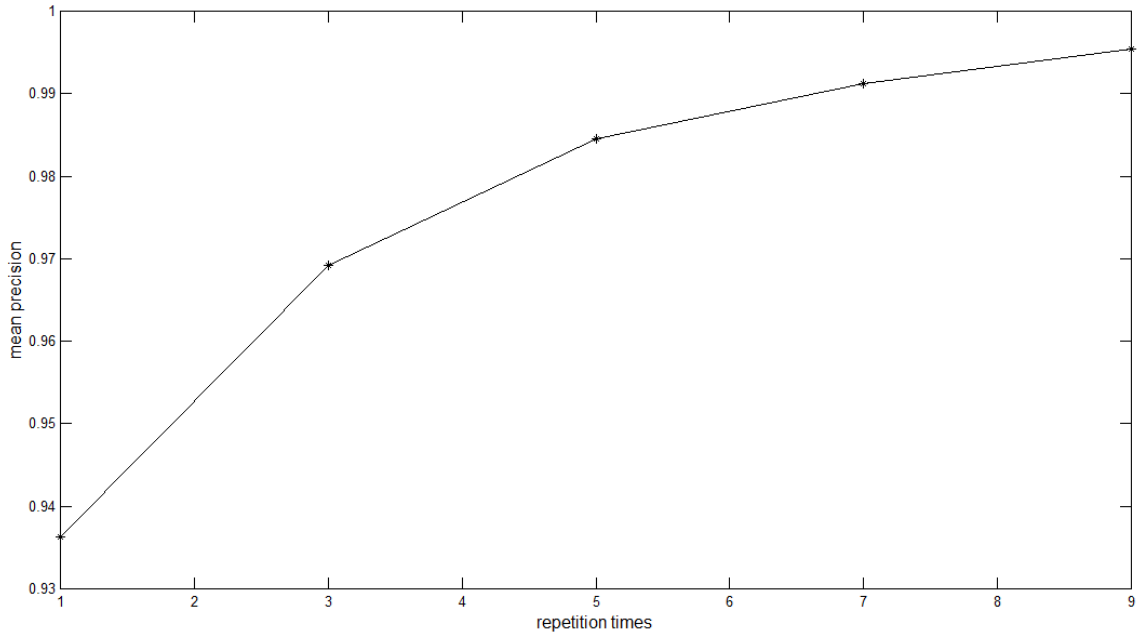


Figure 4.18 The distribution of $Precision_{mean}$ when using different values for the repetition *d*

4.10 Improving the capacity of the scheme

The capacity in the algorithm mentioned above is around 6 bps. However, this can be improved quite simply by using smaller frame size. By setting the frame size to 2048 and 4096 respectively, the capacity will become 24 bps and 12 bps respectively. Experiments were performed on the same 20 music files as that in Section 4.7.1 to examine if the performance is affected when using different frame sizes. The value of *r* is set to 625 and 1250 respectively for the frame size of 2048 and 4096, which corresponds to the same frequency value as *r* is 2500 for the frame size of 8192, as per Equation (4.9). The Th_a, Th_b, Th_c and Th_d were 5, 30, 10 and 1.1 respectively for all the experiments with different frame sizes. The performance of the imperceptibility, capacity, accuracy, robustness against MP3 64 kbps using different frame sizes is shown in Table 4.3.

Table 4.3 Performance comparison when using different frame lengths

Frame size	**Capacity**	**Accuracy**	***Precision*$_{mean}$** (MP3 64 kbps, no repetition)	***Precision*$_{mean}$** (MP3 64 kbps, repetition)	**ODG**
2048	24	0.9919	0.9553	0.9982	-0.472
4096	12	0.9943	0.9396	0.9945	-0.380
8192	6	0.9942	0.9218	0.9927	-0.353

From Table 4.3, it can be seen that with the decrease of the frame size, the capacity and the robustness increase but the imperceptibility decreases.

4.11 Validation on another test data set

In order to validate whether the performance results achieved were applicable to other music files or not, another twenty music files were randomly selected from a variety of genres. All these files were sampled at 48000 Hz. The frame size for both groups is 8192, with 8192 zero-padding. Two groups were set up with the different parameter settings, as shown below:

Group 1: r = 2000, Th_a = 5, Th_b = 30, Th_c =10, Th_d = 1.1, d = 5

Group 2: r = 2500, Th_a = 5, Th_b = 30, Th_c =10, Th_d = 1.1, d = 5

The reason for the parameters setting of Group 1 is that this setting had achieved a satisfactory performance from the experiments mentioned above. The parameters setting of Group 2 only differs from that of Group 1 in the value of r. The purpose of doing this is to ascertain which value of r can achieve a better trade-off.

The 96 watermark bits for each music file were randomly generated. In this experiment, the standard deviation was also calculated. This will indicate how far each individual result is from the mean value for all the 20 music files. The smaller the standard deviation, the closer each individual result is to the mean value, the more likely that the result is independent of the music file.

4.11.1 Accuracy

The *Precision* distribution of these 20 music files without any attack on both groups is shown in Figure 4.19, where the x-axis denotes the file number and the y-axis denotes the *Precision.*

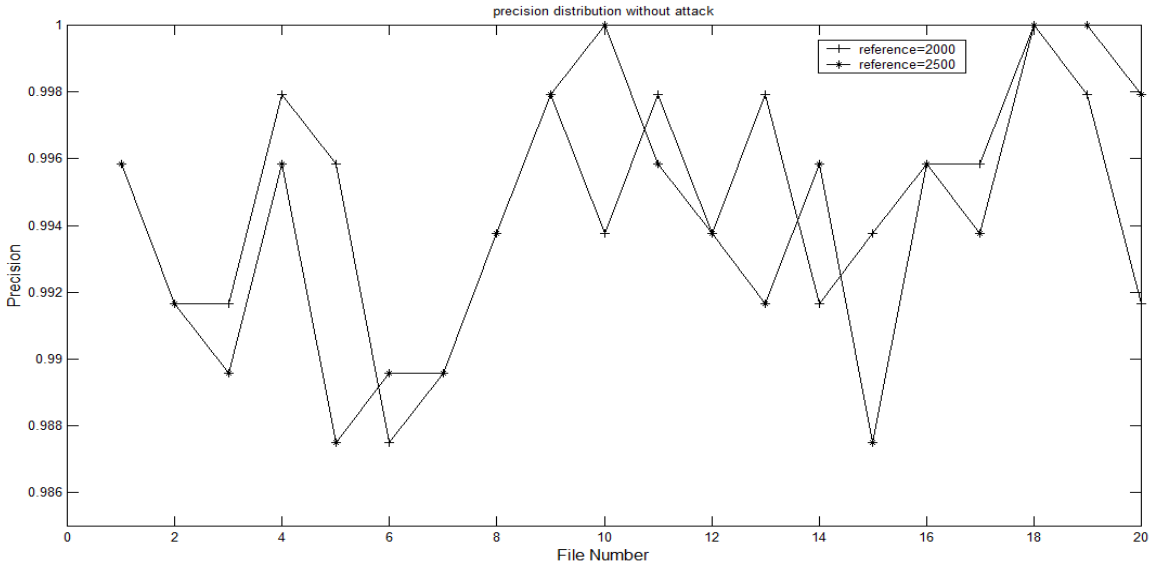


Figure 4.19 The *Precision* distribution without any attack

From Figure 4.19, it can be seen that each *Precision* is around 99%. The $Precision_{mean}$ for both groups is 99.46% and 99.42% respectively. The standard deviations for both groups are 0.0033 and 0.0040 respectively, which indicates a consistency in the performance

across the test files. There is a very little difference in the result of detection when the only difference between test sets is the value of *r*.

4.11.2 Robustness against a variety of attacks

In this Section, the attacks studied were MP3 compression, lowpass filtering, highpass filtering, noise adding and noise removal. The details for these attacks were given in Section 1.3.2.

4.11.2.1 Robustness against MP3 attack

The *Precision* distribution, without using repetition, after MP3 64 kbps compression and decompression on both groups is shown in Figure 4.20. From Figure 4.20, it can be seen that most files have a *Precision* above 85% following MP3 64 kbps. The $Precision_{mean}$ is 0.8985 and 0.9218 respectively. The standard deviations are 0.0789 and 0.0627 respectively, which indicates that the robustness performance fluctuates across different music files. The reason for this is that using the uniform reference value *r* does not consider the specific energy distribution of each different music file, so that it is likely that the manipulation was applied in a very high energy region, which can be masked by its surrounding components and thus the *Precision* is reduced. This can be partly improved by incorporating the 'repetition' process.

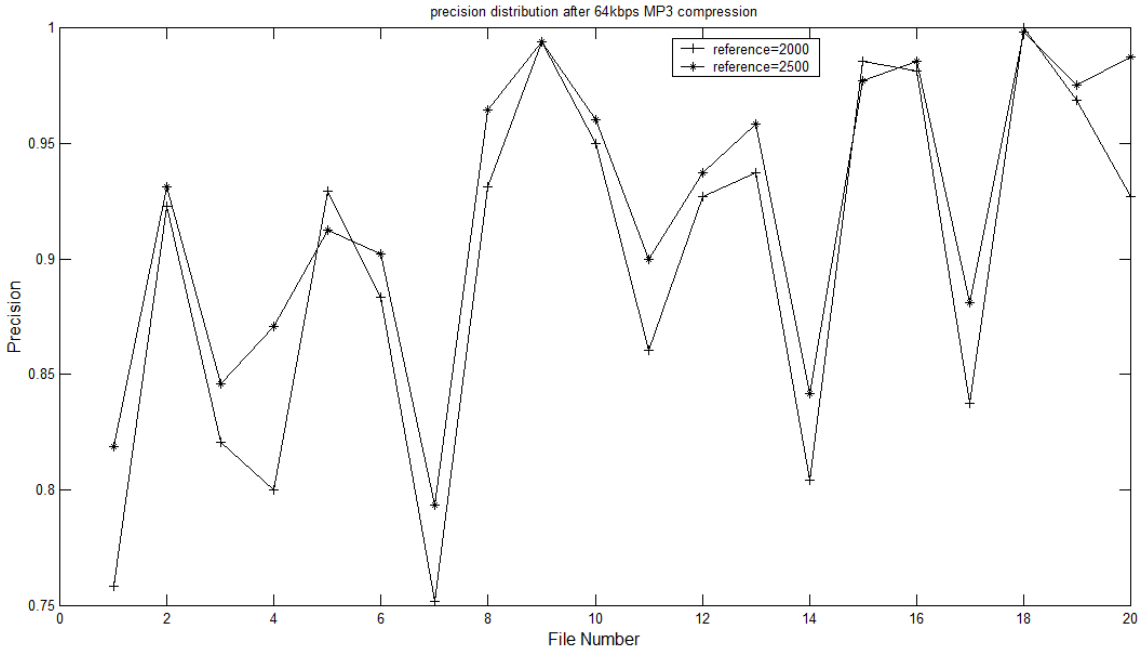


Figure 4.20 The *Precision* distribution after 64 kbps MP3 attack without using repetition

When 'repetition' process was incorporated with d = 5, the *Precision* distribution for both groups, after MP3 64 kbps compression and decompression, is shown in Figure 4.21. From Figure 4.21, it can be seen that there is a large improvement in the *Precision* after adding the repetition process. The $Precision_{mean}$ for both groups is 0.9786 and 0.9927 respectively. The standard deviation for both groups is 0.0340 and 0.0144 respectively.

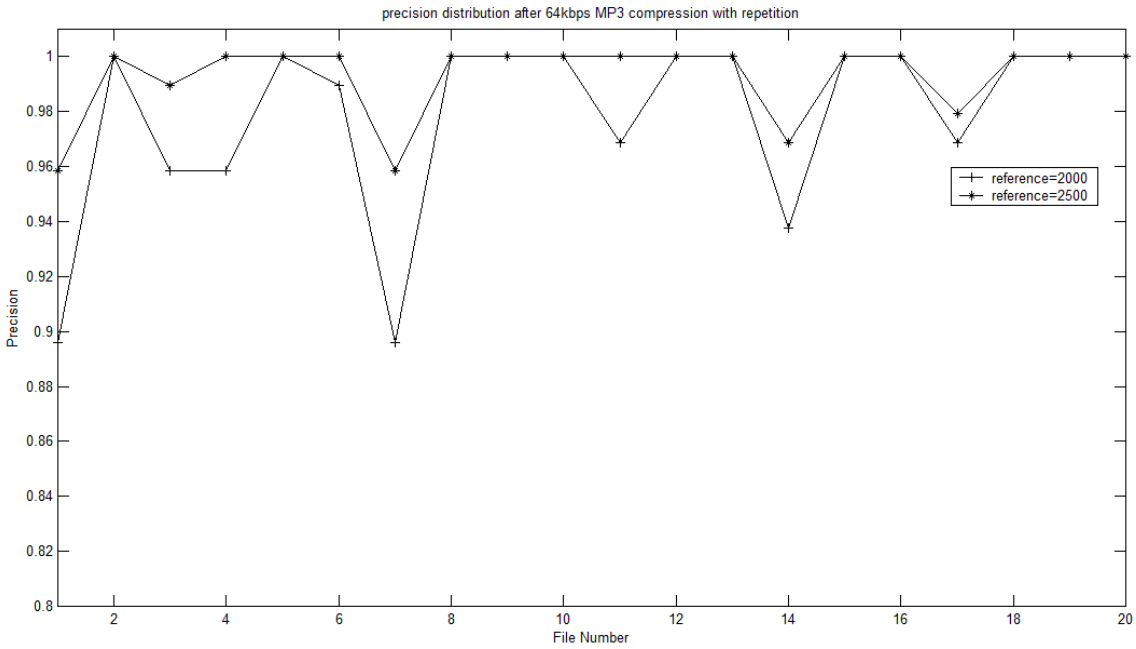


Figure 4.21 The *Precision* distribution after 64 kbps MP3 attack, using repetition (*d* = 5)

However, from Figure 4.21, it can also be seen that File 1 and File 7 did not achieve a sufficiently high robustness yet. This could be further improved by defining a dynamic reference value *r*, which is adapted to each specific signal according to the energy distribution of its CSPE spectrum.

Again, it is noticeable from the results that, with the advantage of the 'repetition' process, there is very little difference in the robustness between both test groups. As a result of this observation, the subsequent experiments were only performed on Group 2, unless stated otherwise.

4.11.2.2 Robustness against AAC attack

AAC is a different lossy compression scheme for digital audio. Designed to be the successor of the MP3 format, AAC generally achieves better sound quality than MP3 at

similar bit rates [Che10]. The AAC export function provided by Audacity [AAC11] was used in this experiment, where a quality scale setting from 10 (worst quality) to 500 (best quality) is available to control the quality of the compressed sound and the bit rate. Here, the quality scale was set to 87, which corresponds to a 64 kbps bit rate. The *Precision* distribution after the AAC compression and decompression process, without using 'repetition', is shown in Figure 4.22 where the x-axis denotes the file number and the y-axis denotes the *Precision*. From Figure 4.22, it can be seen that most files have a *Precision* above 85% following AAC compression. However, still there are few files have a *Precision* below 85%. The reason for this is similar to that stated in Section 4.11.2.1. The $Precision_{mean}$ is 0.9132 and the standard deviation is 0.0645.

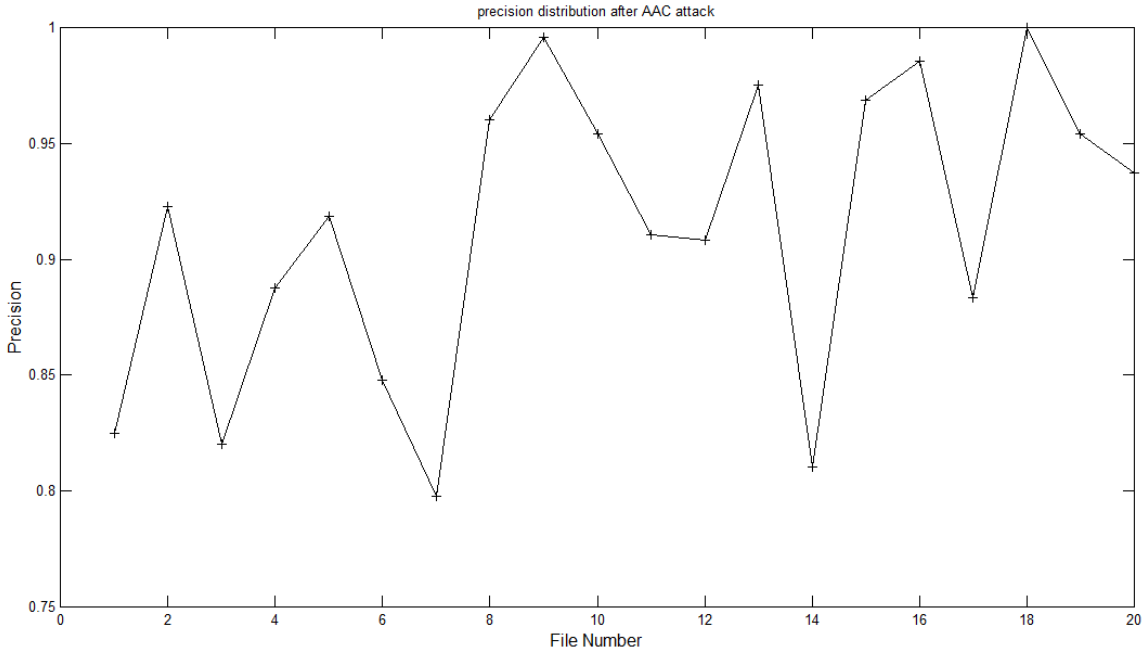


Figure 4.22 The *Precision* distribution after AAC attack without using repetition

The ‘repetition’ process was then added with *d*=5, and the *Precision* distribution is shown as Figure 4.23. From Figure 4.23, it can be seen that there is a large improvement in *Precision* after adding the repetition process. The $Precision_{mean}$ is 0.9860 and the standard deviation is 0.0231.

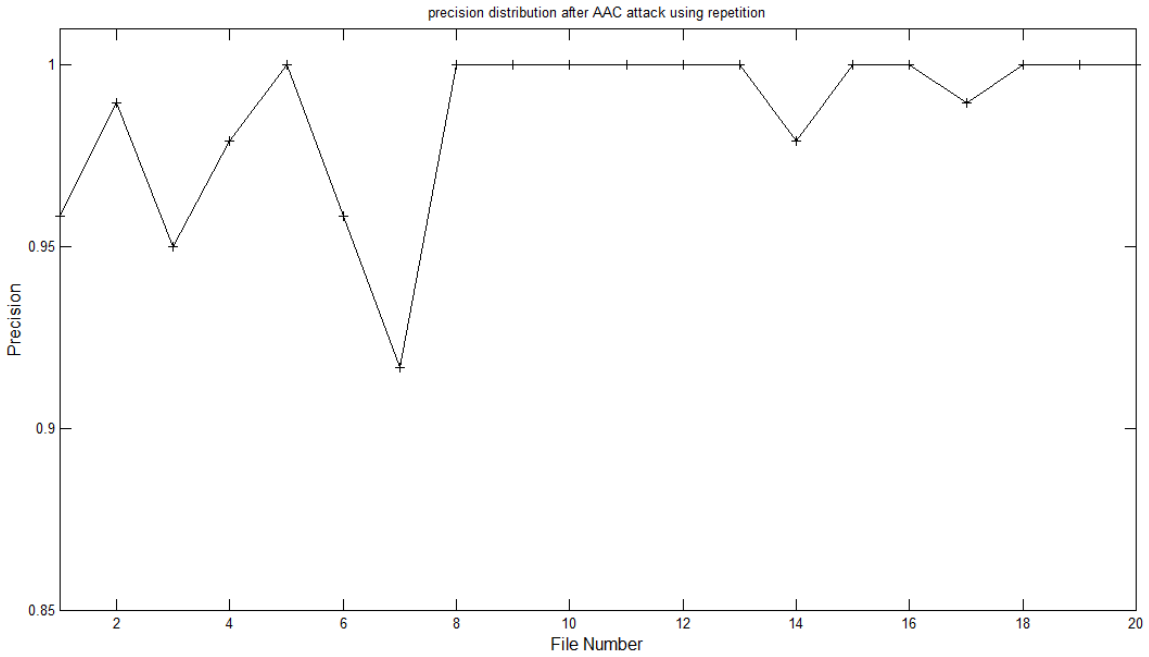


Figure 4.23 The *Precision* distribution after AAC attack using repetition (*d* = 5)

4.11.2.3 Robustness against Additive White Gaussian Noise attack

Additive White Gaussian Noise (AWGN) [BDG03] is a signal with a constant spectral density and a Gaussian distribution of amplitude. The addition of different levels of AWGN has a different perceptual effect on audio quality. As mentioned in Section 1.3.2, the most powerful attacks are those that can remove or distort the watermark information without severely degrading the audio quality. In other words, if an attack is made by adding an AWGN signal to the watermarked signal, and results in it being perceptually

obtrusive, then this attack is not effective as the commercial value of the watermarked signal is lost. A PEAQ test was conducted to assess the perceptual degradation caused by the addition of AWGN.

A particular level of AWGN whose power is -40 dB was added to each of the 20 watermarked signals. If assuming the signal power is 0 dB, after adding this level of AWGN, the SNR of the signal will become 40 dB. The ODG score of each resulting file is distributed as Figure 4.24, where the x-axis denotes the file number and the y-axis denotes the ODG score.

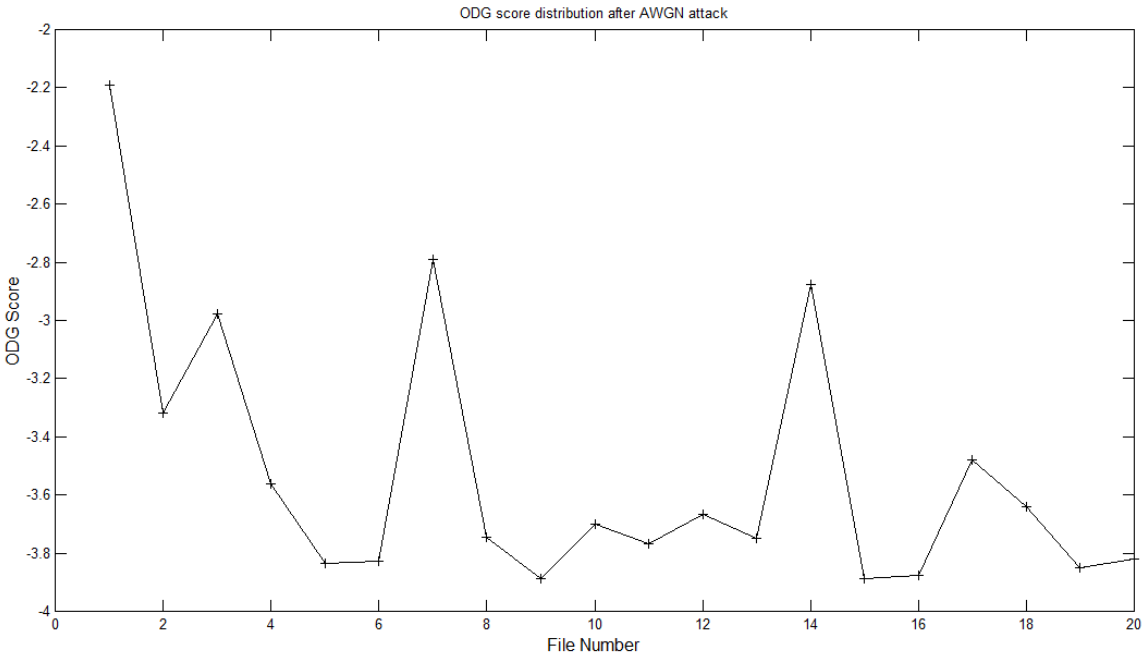


Figure 4.24 The ODG score distribution of each resulting signal file after adding a particular level of AWGN

From Figure 4.24, it can be seen that the ODG scores were very poor. The mean ODG score is -3.5225. This means that, after adding this particular level of AWGN to

each signal, the listening quality of signals was degraded severely and thus their commercial value was destroyed. There is no necessity to add an AWGN with a higher power, as this would worsen the listening quality.

These AWGN-corrupted watermarked signals were used to test the watermark robustness. The *Precision* distribution without using 'repetition' is shown in Figure 4.25 where the x-axis denotes the file number and the y-axis denotes the *Precision*. From Figure 4.25, it can be seen that all files can strongly survive this particular level of AWGN attack. The $Precision_{mean}$ is 0.9844 and the standard deviation is 0.0071. It is certain that the watermark can survive an AWGN attack with a lower power.

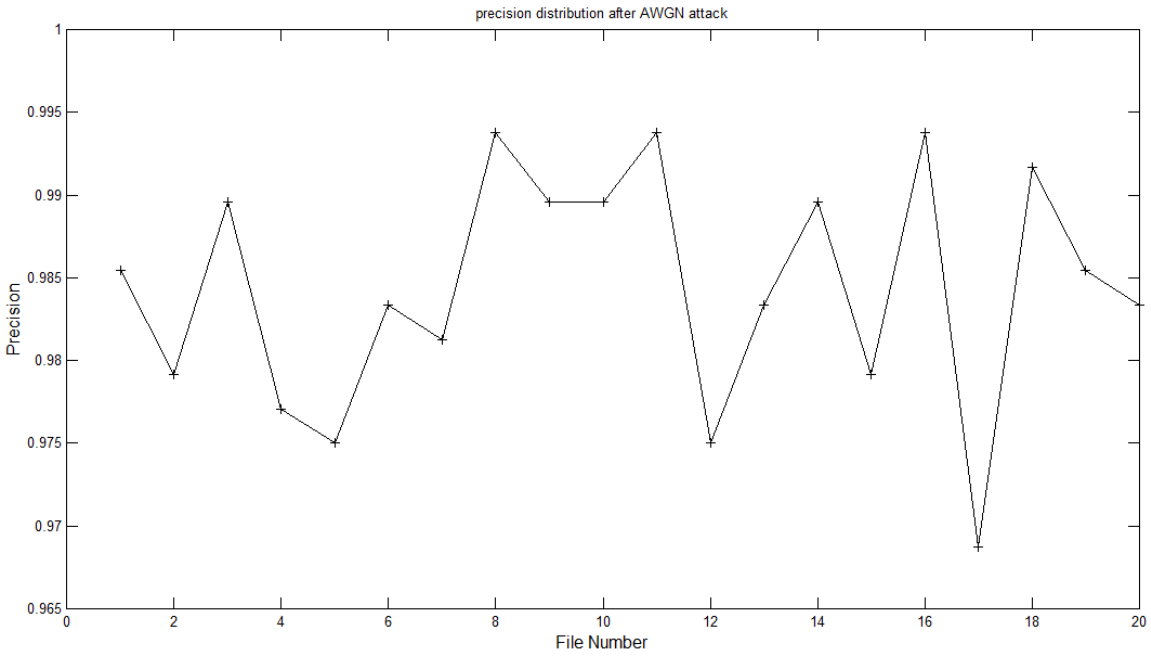


Figure 4.25 The *Precision* distribution after a particular level of AWGN attack without using repetition

Likewise, the 'repetition' process was incorporated with d = 5, and the *Precision* distribution is shown in Figure 4.26. From Figure 4.26, it can be seen that all the test files can completely survive this particular AWGN attack after including the 'repetition' process. The $Precision_{mean}$ is 1 and the standard deviation is 0.

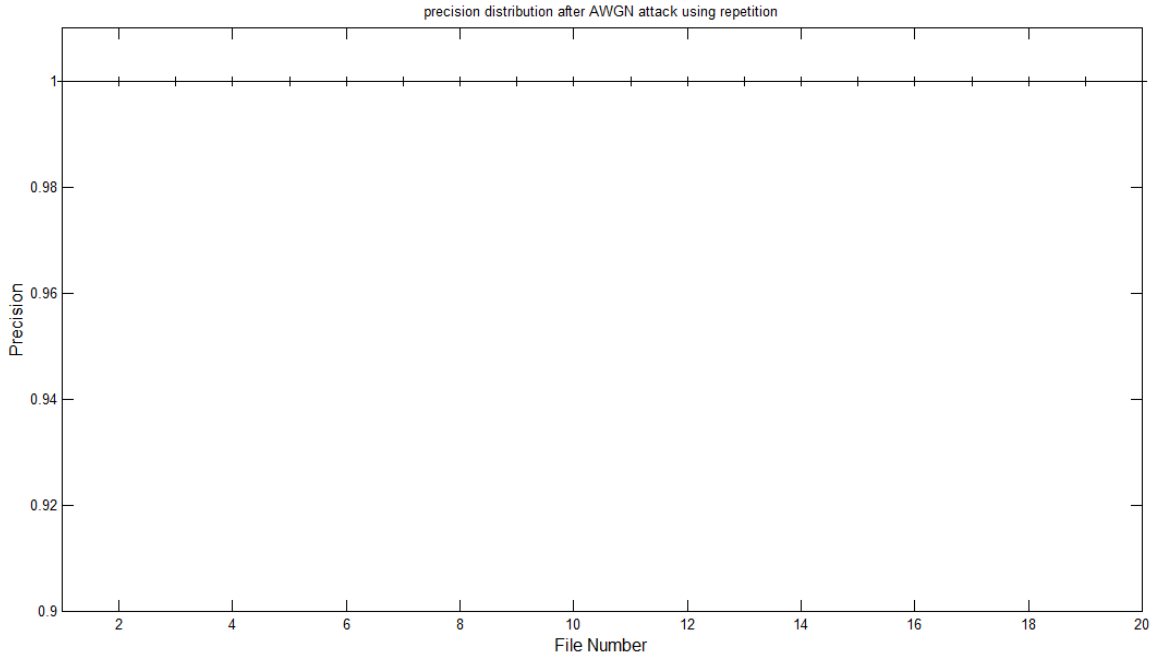


Figure 4.26 The *Precision* distribution after a particular level of AWGN attack using repetition (*d* = 5)

4.11.2.4 Robustness against a lowpass filtering attack

An experiment was conducted to investigate whether the watermark can survive lowpass filtering or not. The lowpass filtering attack was provided by the Audacity tool [Aud10]. The roll-off was set to be 6dB per octave and the cut-off frequency was set to be 8000 Hz, as the frequency components above 8000 Hz are at risk of being removed by signal processing such as lossy compression [KS05]. The result of this experiment, without

using repetition, is shown in Figure 4.27 where the x-axis denotes the file number and the y-axis denotes the *Precision*. From Figure 4.27, it can be seen that each test file can strongly survive a lowpass filtering attack. The $Precision_{mean}$ is 0.9942. The standard deviation is 0.0040, which indicates that there is no big fluctuation between the robustness of different files.

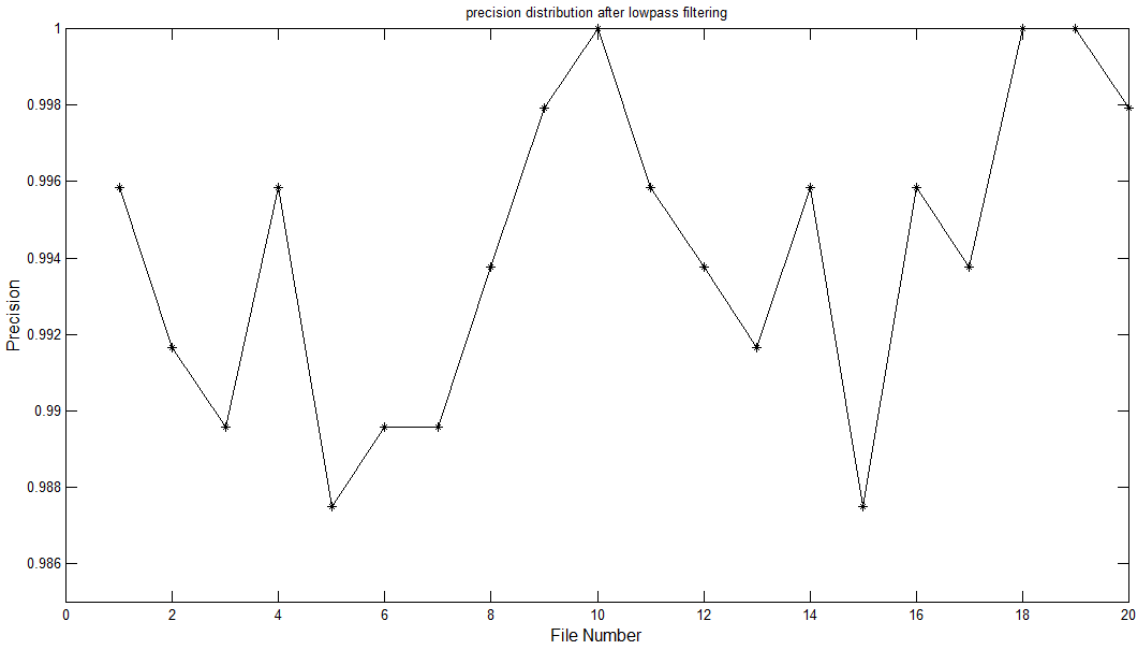


Figure 4.27 The *Precision* distribution after lowpass filtering attack without using repetition

Likewise, the 'repetition' process was incorporated with d = 5. The *Precision* of each file is distributed as Figure 4.28. From Figure 4.28, it can be seen that all the test files can completely survive the lowpass filtering attack after adding the 'repetition' process. The $Precision_{mean}$ is 1 and the standard deviation is 0, which means that the watermark of each file can survive a lowpass filtering attack completely after incorporation of the 'repetition' process.

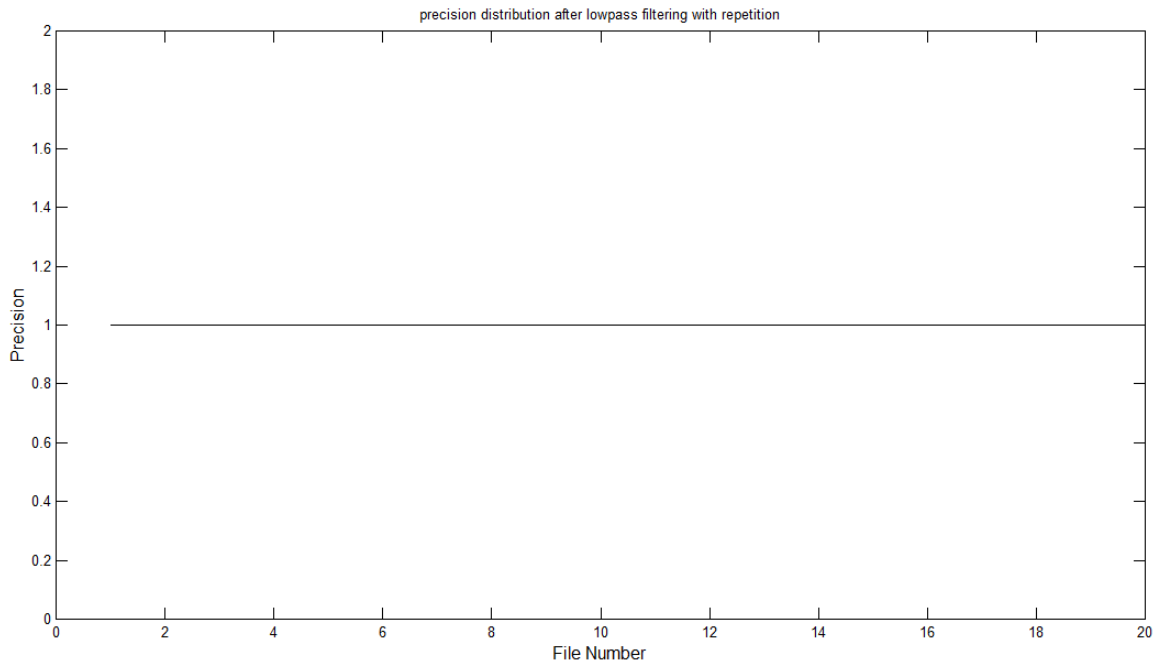


Figure 4.28 The *Precision* distribution after lowpass filtering attack using repetition (*d* = 5)

4.11.2.5 Robustness against a highpass filtering attack

An experiment was conducted to investigate whether the watermark can survive highpass filtering or not. The highpass filtering attack was provided by Audacity tool [Aud10]. The roll-off was set to be 6 dB per octave and the cut-off frequency was set to be 2000 Hz. The result of this experiment, without using repetition, is shown in Figure 4.29 where the x-axis denotes the file number and the y-axis denotes the *Precision*. From Figure 4.29, it can be seen that most files can strongly survive this highpass filtering.

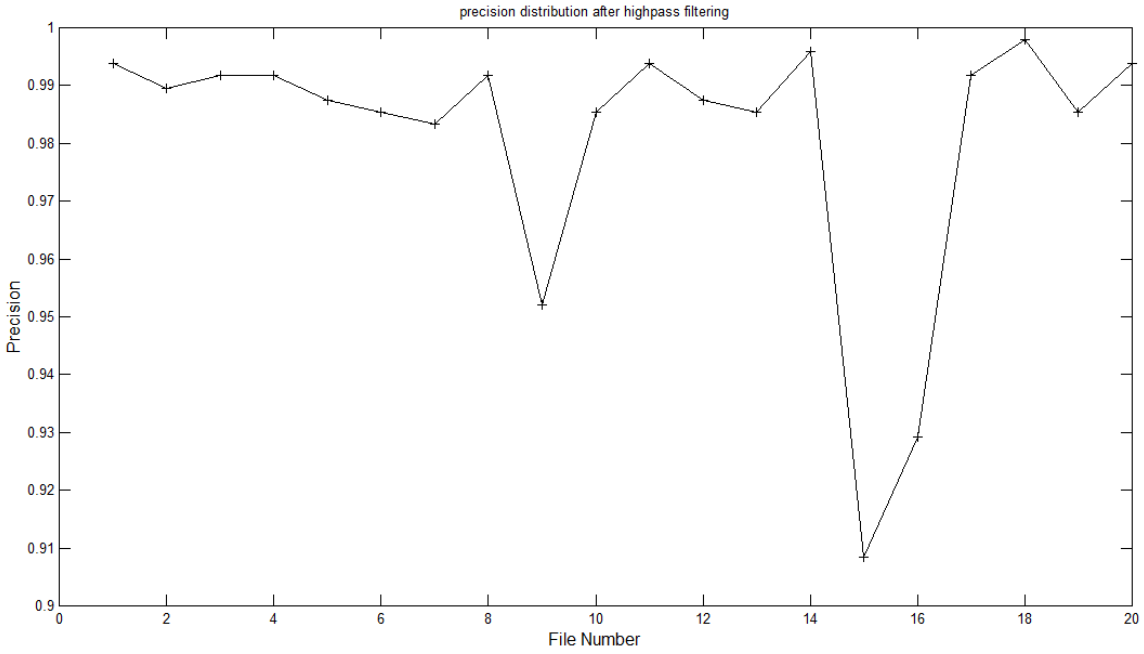


Figure 4.29 The *Precision* distribution after highpass filtering attack without using repetition

Likewise, the 'repetition' process was incorporated with d = 5. The *Precision* of each file is distributed as Figure 4.30. From Figure 4.30, it can be seen that all the test files can completely survive the highpass filtering attack after adding the 'repetition' process. The $Precision_{mean}$ is 1 and the standard deviation is 0, which means that the watermark of each file can survive a highpass filtering attack completely after incorporation of the 'repetition' process.

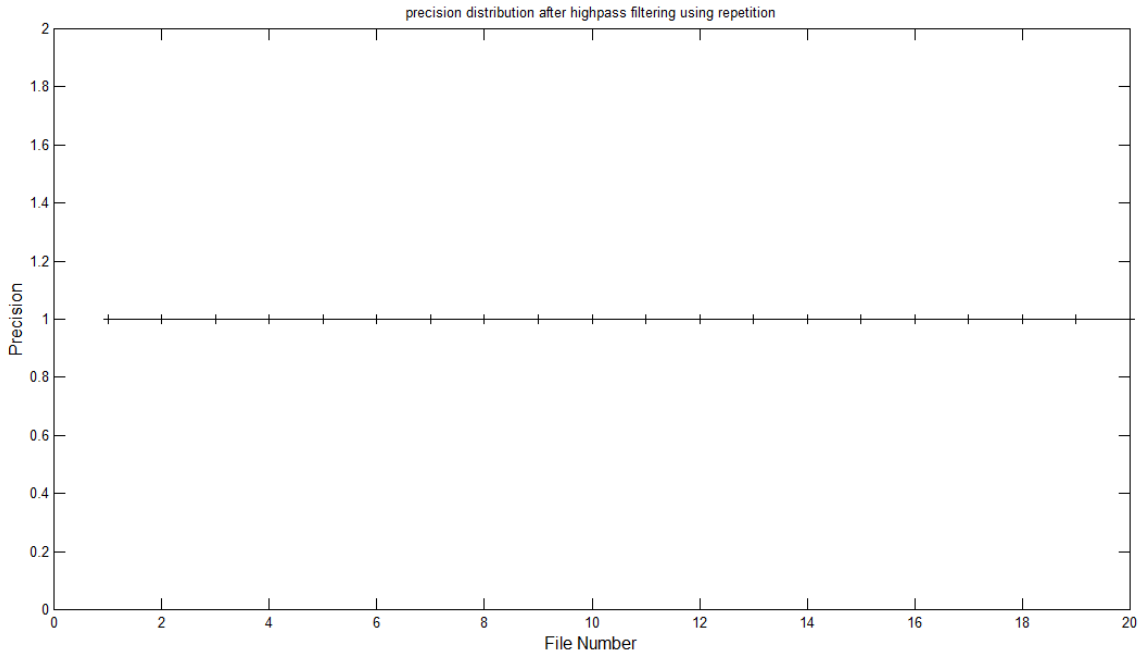


Figure 4.30 The *Precision* distribution after highpass filtering attack using repetition (*d* = 5)

4.11.2.6 Robustness against Noise Removal attack

An experiment was conducted to test the robustness of the algorithm against a Noise Removal attack. One typical noise removal technique, hiss-removal, was provided by Goldwave [Gol11]. The *Precision* of each file, without using repetition, is distributed as Figure 4.31 where the x-axis denotes the file number and the y-axis denotes the *Precision*. From Figure 4.31, it can be seen that each test file can strongly survive this attack. The $Precision_{mean}$ is 0.9856 and the standard deviation is 0.0057, which means that each test file can strongly survive this attack.

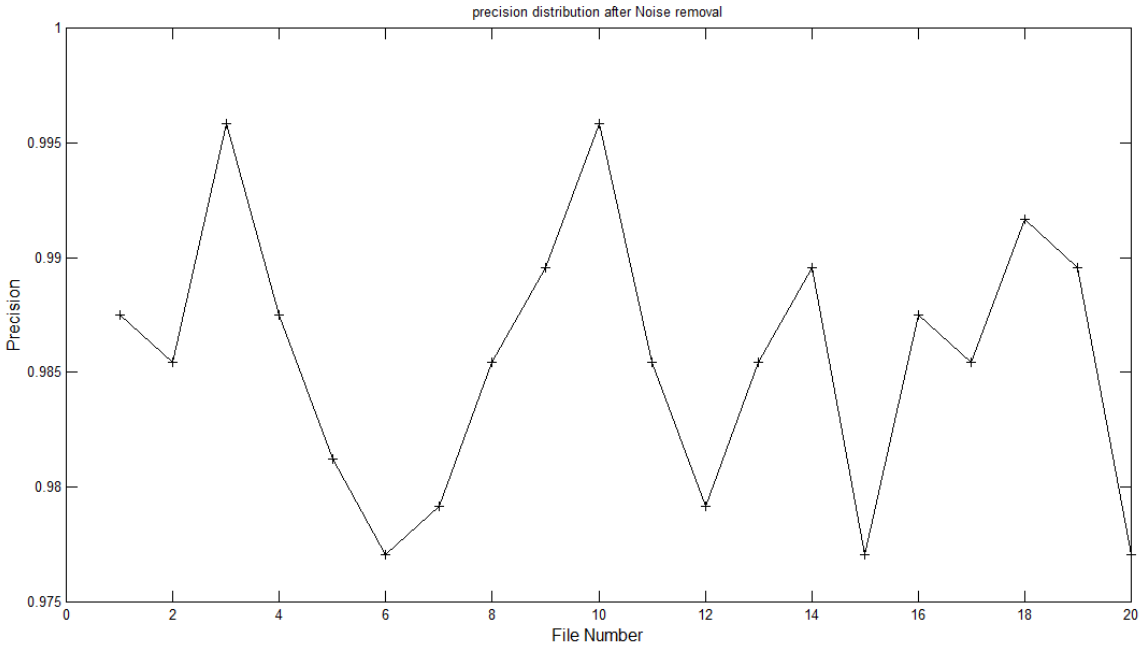


Figure 4.31 The *Precision* distribution after noise removal attack without using repetition

Then, the 'repetition' process was incorporated with $d = 5$ and the result is shown in Figure 4.32. As can be seen from the Figure 4.32, each file can survive this attack completely.

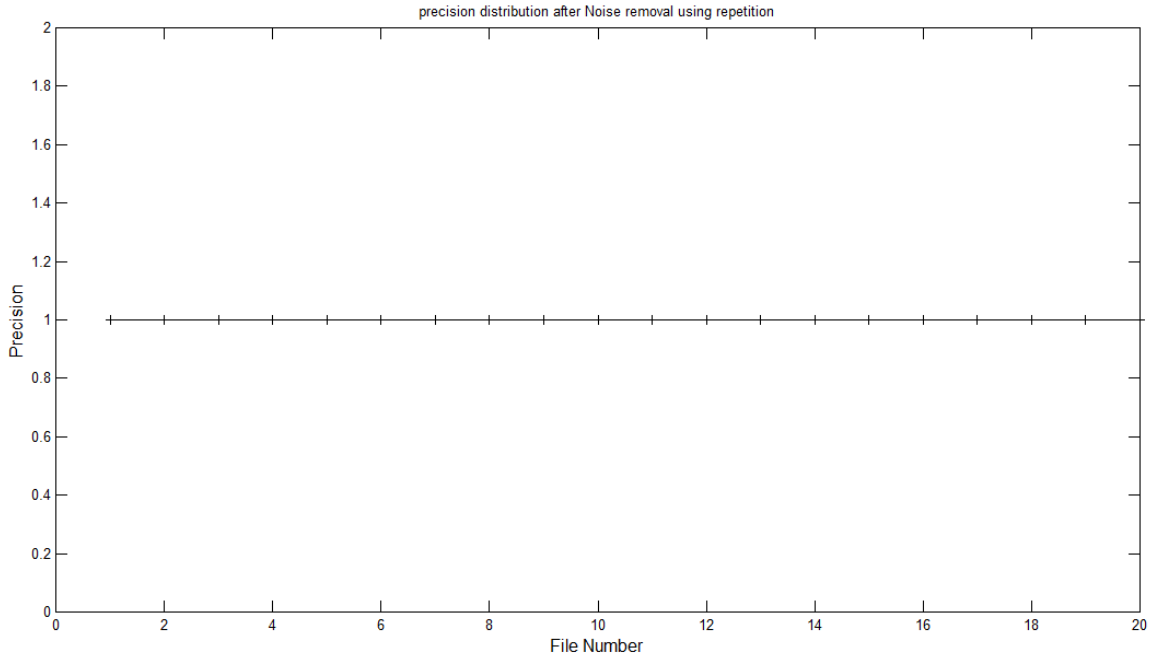


Figure 4.32 The *Precision* distribution after noise removal attack using repetition (*d* = 5)

4.11.3 Imperceptibility

Experiments were conducted to test the imperceptibility of the watermarking algorithm. The SNR and SNRseg, as defined in Equation (2.1) and (2.2) respectively, were calculated. The distribution of the SNR and SNRseg are shown in Figure 4.33. From Figure 4.33, it can be seen that SNR and SNRseg are very high. The mean SNR and SNRseg are 36.0567 dB and 33.3832 dB respectively and the standard deviations are 3.4949 dB and 1.5112 dB respectively, which conform to the IFPI standard.

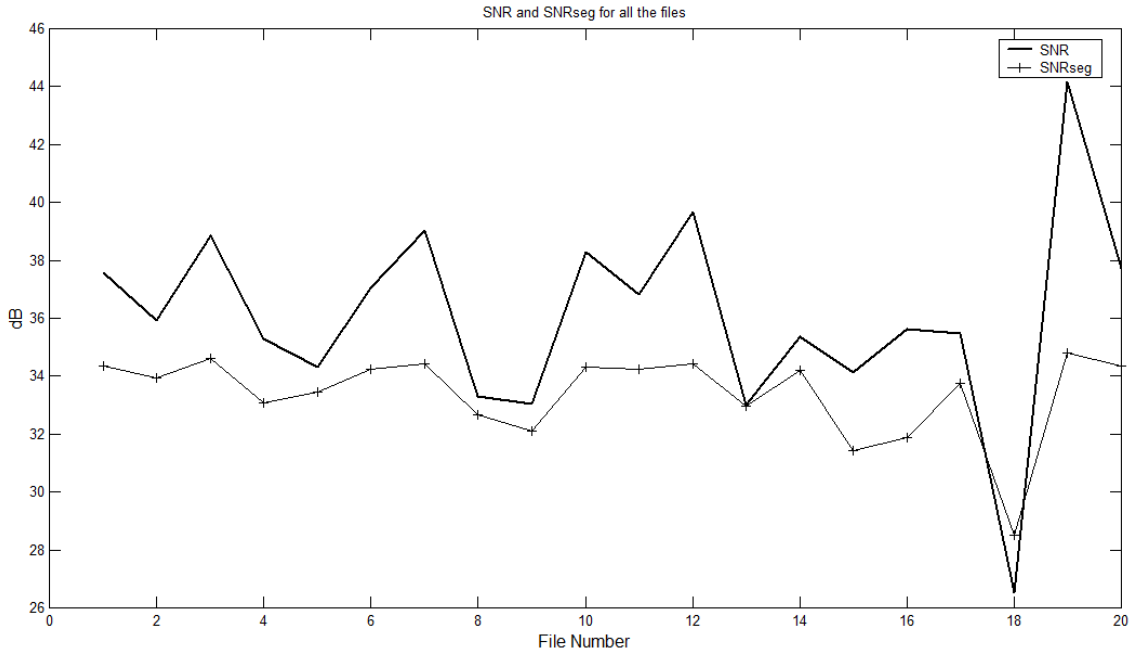


Figure 4.33 (a) SNR distribution for all the 25 files (b) SNRseg distribution for all the 25 files

The ODG scores of each file in Group 1 and Group 2 are distributed as Figure 4.34, where the x-axis denotes the file number and the y-axis denotes the ODG score. The mean ODG for both groups is -0.2699 and -0.3536 respectively and the standard deviation of the ODG is 0.3601 and 0.3834 respectively. This relatively high standard deviation is because few test files did not achieve satisfactory ODG scores, such as File 5 and 9. The reasons for this is that a uniform reference *r* was used, which really should be defined in a non-uniform manner, that is, according to each file's CSPE spectrum energy distribution. In addition, the manipulation of the magnitude of each candidate peak might benefit from fine-tuning of the thresholds, which can also improve the imperceptibility.

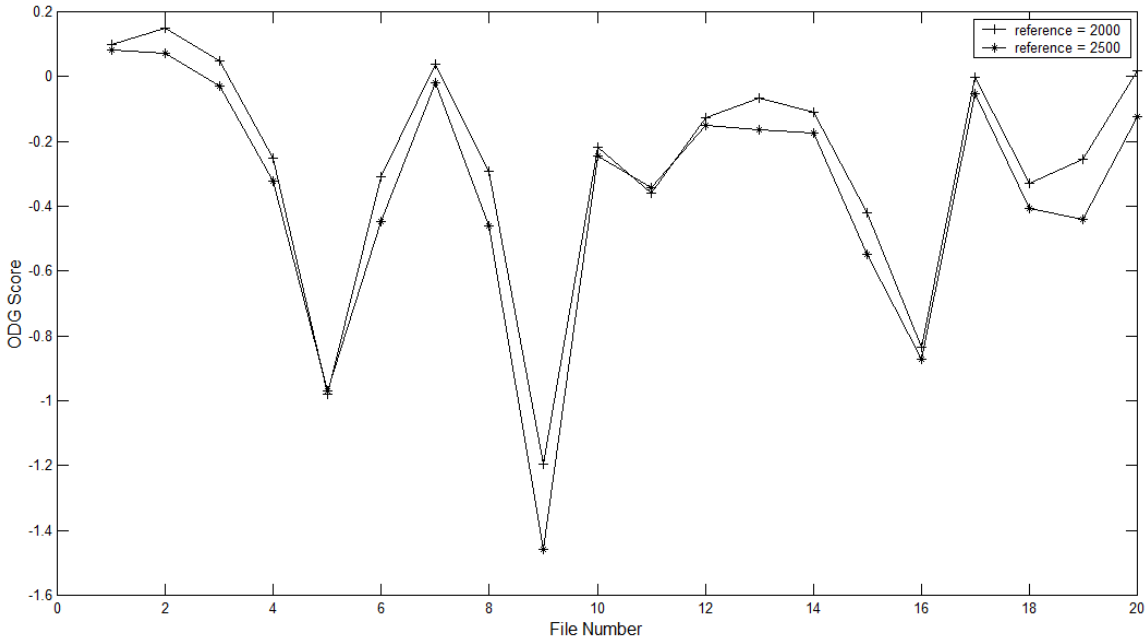


Figure 4.34 The ODG distribution for Group 1 and 2

4.11.4 Computational efficiency

In some time-critical scenarios such as broadcast monitoring, a real-time detector is highly desirable [MK11]. Generally, the computational efficiency of the embedding process is less crucial than that of the detection process. The reason for this is that the embedding process is a one-off process whereas the detection process could occur very regularly. Due to the fact that CSPE is a computational efficient approach, and the detection processes are performed independently in each frame, which is possible to execute in parallel, thus, it would be likely to achieve a high computational efficiency for detection process.

4.12 Comparison with other algorithms

The performance of the proposed watermarking algorithm was compared to the two algorithms from [BSD10] and [FM09]. These two algorithms achieved a comparatively better overall performance than others as analyzed in Section 2.4 of Chapter 2. The comparison result is shown in Table 4.4 and the ODG score was mapped to the MOS score according to Table 2.3.

From Table 4.4, it can be seen that the proposed algorithm achieved a higher imperceptibility, but a lower capacity. As mentioned, the embedding position is dynamically generated according to each specific frame, thus the watermark bits embedded by the proposed algorithm are harder to remove than the algorithm proposed in [FM09]. In addition, the algorithm used bin location instead of magnitude as the basis of embedding rule, thus it should be stronger than the algorithm proposed in [FM09], to those attacks that interfere with the magnitudes. The computational efficiency of the algorithm is higher than that proposed in [BSD10], as analyzed in Section 4.11.4.

Table 4.4 Performance comparison of audio watermarking schemes

	Method	**Capacity**	**Blind**	**Robustness to 64 kbps MP3**	**MOS**	**SNR**
Proposed	CSPE	24	Yes	Yes	4.65	36.05
[BSD10]	SVD	45.9	Yes	Yes	4.46	24.37
[FM09]	FFT	3000	Yes	Yes	4.50	28.55

4.13 Summary

This chapter proposed an enhanced CSPE-based watermarking algorithm. It was developed because the robustness of the algorithm proposed in Chapter 3 was not sufficient. The peaks identified in the CSPE spectrum were found to be robust following signal processing such as MP3 compression, thus they were used to embed the watermark information. The candidate peak's frequency was altered to satisfy the embedding rule, which was based on the parity of the candidate peak's bin location. A novel threshold measure was proposed to avoid the potential issues that can degrade the performance of the algorithm. Some processes such as a sharpness measure and 'repetition' were employed to improve the algorithm's performance.

Experiments were carried out to investigate the performance in terms of accuracy, robustness, imperceptibility, capacity and computational efficiency. As shown by the experiments, the performance of this algorithm was found to be:

1. Almost 100% accurate for watermark detection without any attack
2. Robust against attacks such as lossy compression (MP3, AAC), lowpass filtering, highpass filtering, AWGN and Noise Removal
3. Perceptually transparent with a mean ODG score of around -0.3
4. A capacity of around 24 bps
5. Potentially computationally efficient
6. Not easy to remove

A different watermarking algorithm will be developed from a different perspective, which will be detailed in Chapter 5.

Chapter 5 Singular Value Decomposition based audio watermarking Algorithm

5.1 Introduction

The capacity achieved by the CSPE based watermarking algorithm, which was proposed in Chapter 4, is not sufficient for some application scenarios. Considering this, a different algorithm should be developed which would achieve a higher capacity and also maintain a sufficient imperceptibility and robustness. From the literature review, it can be seen that SVD has much potential to achieve this aim. Therefore, a new audio watermarking algorithm based on SVD is proposed. This algorithm is based on a novel observation, that is, watermarks can be embedded by manipulating the elements in the second column of the matrix U, as defined in Equation (2.11). The advantage of doing this is that it introduces less distortion and can achieve a high robustness. Some additional processes are incorporated into the watermarking algorithm to minimize the audible distortion. Compared with the CSPE-based watermarking algorithm proposed in Chapter 4, this algorithm is easier to adjust to achieve an acceptable level of performance, especially in terms of robustness because of the inherent stability of the SVD. Furthermore, a higher capacity can be attained because there are more elements available from the decomposition to manipulate.

5.2 SVD

The SVD is a well known matrix decomposition tool that has a wide variety of applications in signal processing. The Compact Singular Value Decomposition (CSVD) is a more compact representation of the SVD and compared with the full SVD described in Chapter 2, this version is more computationally efficient [TB97]. With the CSVD, a matrix A can be decomposed as Equation (5.1) where U and V are $p \times r$ and $r \times q$ unitary matrices respectively and S is a $r \times r$ diagonal matrix with positive elements. The difference between the full SVD and the CSVD is that the decomposed matrix S from the CSVD only contains positive values, while S might contain zero values in the full SVD [TB97].

$$A = USV^T = \begin{bmatrix} u_{1,1} & \cdots & u_{1,r} \\ \vdots & \ddots & \vdots \\ u_{m,1} & \cdots & u_{m,r} \end{bmatrix} \begin{bmatrix} s_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{r,r} \end{bmatrix} \begin{bmatrix} v_{1,1} & \cdots & v_{1,r} \\ \vdots & \ddots & \vdots \\ v_{n,1} & \cdots & v_{n,r} \end{bmatrix}^T \quad (5.1)$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{r} u_{i,k} * s_{k,k} * v_{k,j}$$

There is an intriguing analogy between the SVD and Fourier analysis [Kal96]. Particularly in the discrete case, Fourier analysis can be viewed as representing input data using orthogonal basis. The basis element here is the sine function. The Fourier decomposition thus represents the input data as a superposition of vibrations on the basis elements. Often, there are a few principal frequencies that account for most of the variability in the original data. However, the SVD captures the best possible basis for the particular data, rather than using one standard basis for all cases, because the basis is derived from the data itself. A particular observation made was that the modification on

the elements in the first column of U can be almost retained following signal processing operations [FL08]. This observation has been used to develop video watermarking algorithms. Its effectiveness in the audio watermarking will be investigated.

5.3 SVD based watermarking algorithm

In this section, a new watermarking algorithm based on the SVD will be proposed. Each process involved will be described in detail.

5.3.1 Matrix organization

Before applying the CSVD, the data has to be organized in a matrix form. Firstly, the signal is split into frames with length l. The magnitude spectrum for each frame can be obtained by the FFT. Each spectrum contains $l/2$ frequency bins below the Nyquist frequency. The magnitude values of these $l/2$ frequency bins can be arranged as a matrix. In this chapter, the matrix shape is defined as 64×8, l is defined as 1024. Then, the first 64 magnitude values will be put into the first column of the matrix, the second 64 magnitude values will be put into the second column of the matrix, until all of the magnitude values are put into the matrix. This matrix then can be decomposed by the CSVD and three matrices U, S and V are obtained.

5.3.2 Using the second column of U to embed the watermark

As in Equation (5.1), $s_{1,1}$ is the largest value in S and it is only multiplied with the column elements $u_{1..m,1}$, which results in the most significant peaks within the matrix A. A minor alteration of any element in $u_{1..m,1}$ may therefore result in significant distortions when the signal is reconstructed. Similar to the point made about $s_{1,1}$, $s_{2,2}$ is the second largest

value in the *S* matrix and it is only multiplied with column elements $u_{1..m,2}$. Thus, modifying the elements in $u_{1..m,2}$ rather than those in $u_{1..m,1}$ will result in less distortion in the reconstructed signal.

An experiment was conducted to verify this statement. Twenty music files were randomly selected from a variety of genres. The pseudo code of the experimental procedure is outlined below. Note, *File_No* means file number ranging from 1 to 20. *Col_No* means column number ranging from 1 to 8.

1. for (*File_No* =1; *File_No*<=20; *File_No*++)
    I. A frame with length of 1024 samples was selected from the current file. Note that the frame number was randomly generated and was the same for all the files.
    II. The magnitudes of the first half of this frame's FFT spectrum were generated, as per Equation (4.2), and arranged as a 64*8 matrix *A*.
    III. This matrix was decomposed by the SVD into matrices *U*, *S* and *V*.
    IV. for (*Col_No* = 1; *Col_No* <=8; *Col_No* ++)
        i. The elements of the current column of the *U* matrix were multiplied by 0.1 and produced the corresponding altered matrix *U'*. Then a new matrix *A'* was created by multiplying *U'*, *S*, *V*.
        ii. Calculate the absolute magnitude difference between the matrix *A* and *A'*, denoting as $Diff(File\_No, Col\_No)$, which is formulated as Equation (5.2).

$$Diff(File\_No, Col\_No) = \sum_{j=1}^{8} \sum_{i=1}^{64} \left| A(i,j) - A'(i,j) \right| \quad (5.2)$$

2. Calculate the mean absolute difference $AveDiff(Col\_No)$ of all the files, as per Equation (5.3):

$$AveDiff(Col\_No) = \frac{\sum_{File\_No=1}^{20} Diff(File\_No, Col\_No)}{20} \quad (5.3)$$

The distribution of $AveDiff$ can be depicted in Figure 5.1. From Figure 5.1, it can be seen that the same alteration, applied on the elements of different column of *U*, creates different levels of distortion. More specifically, alteration on the elements of the first column creates the largest distortion, followed by that on the elements of the second column. Therefore, it is possible to embed watermarks by manipulating the elements in all the columns of *U* except the first column, which introduces less distortion.

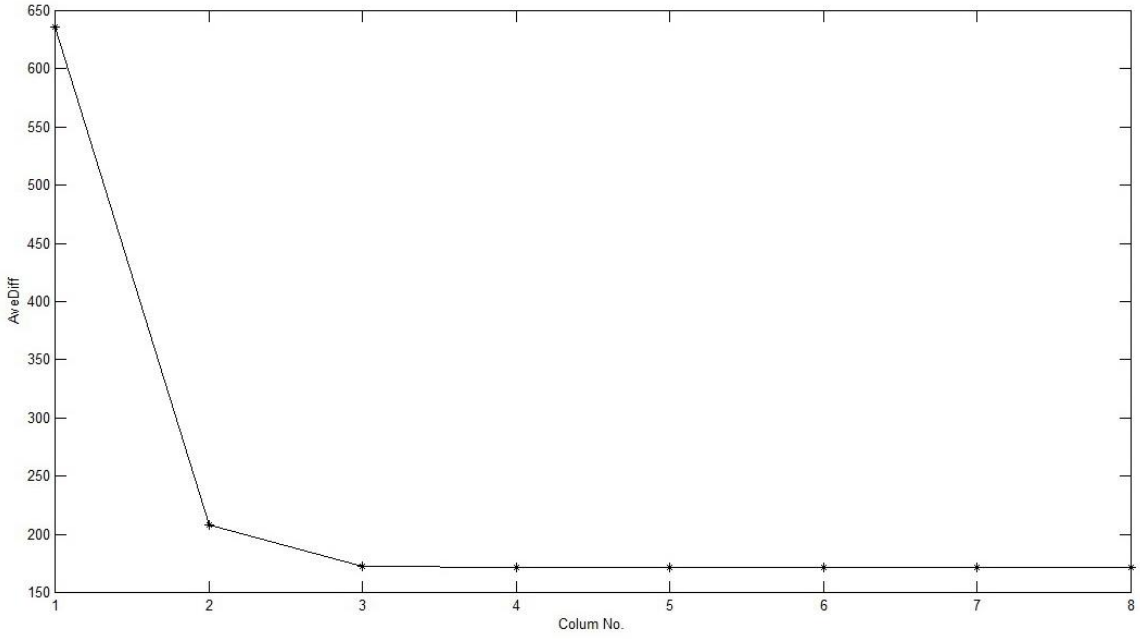


Figure 5.1 The demonstration of the distortion introduced by manipulation different columns of the matrix *U*.

However, by experiments, it was found that only the peaks, which were created by manipulation, in $u_{1..m,1}$ and $u_{1..m,2}$ can be retained after MP3 compression. The robustness of the peaks created in $u_{1..m,2}$ against MP3 64 kbps compression was demonstrated by the following experiment. The procedure of this experiment was outlined as follows:

1. A random 'original' signal was selected from a collection of music tracks with a variety of genres.
2. A random frame was selected from this file with length 1024 samples.
3. The magnitudes of this frame's FFT spectrum were generated, as per Equation (4.2), and arranged as a 64*8 matrix *A*.

4. *A* was decomposed by the SVD into *U*, *S* and *V*.
5. Four elements at positions 11, 25, 46 and 61 of the second column of *U* were manipulated by decreasing the magnitudes of their neighbouring elements, so that these four elements were peaks. By multiplying this altered matrix with *S* and *V* according to Equation (5.1), a new matrix *A'* was generated.
6. *A'* was reconstructed to the time domain frame, so that the 'modified' signal was formed. A matrix *U'* can be derived by running step 2-4 on this 'modified' signal.
7. The modified signal was processed by MP3 64 kbps compression and decompression, which produced the 'processed' signal. A matrix *U''* can be derived by running step 2-4 on this 'processed' signal.

Elements at positions 11, 25, 46 and 61 of the second column of *U*, *U'*, and *U''* respectively were depicted by arrows in the first and second panel of Figure 5.2, where the x-axis denotes the position and the y-axis denotes the element value.To ensure good visibility in the figure, only the relevant regions are displayed. The first panel in Figure 5.2 shows the position ranges from 8 to 27 and the second panel shows the position ranges from 42 to 62. The comparison in magnitude between the original signal and the modified signal was shown in third panel of Figure 5.2, where the x-axis denotes the Frequency bin and the y-axis denotes the magnitude value in dB.

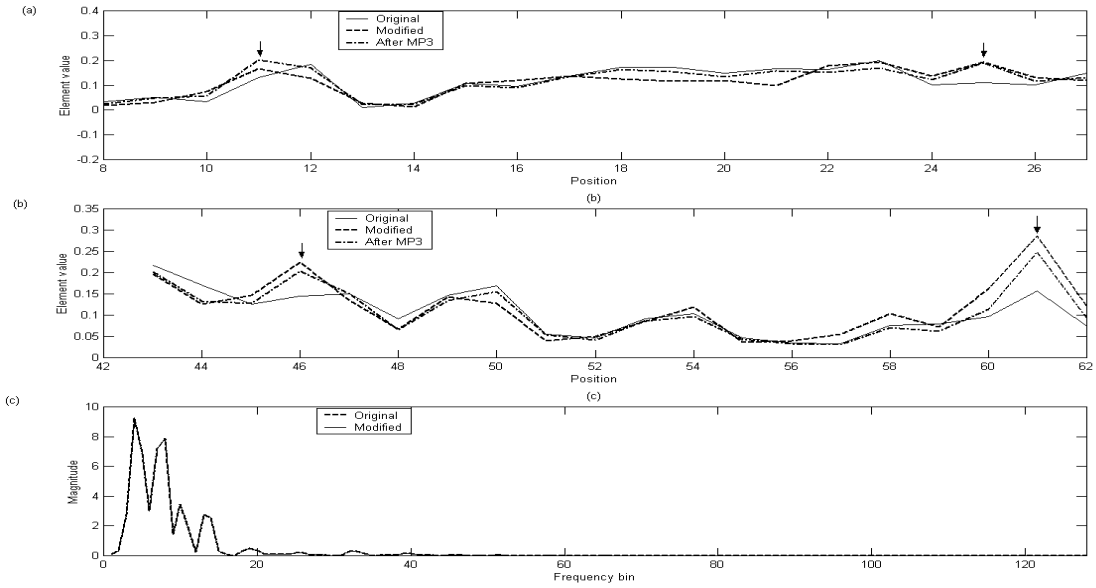


Figure 5.2 (a) Robustness of the created peaks (b) Robustness of the created peaks (c) Spectrum of the original signal and the modified signal

From the first and second panel, it can be seen that all the 4 created peaks were retained after MP3 64 kbps compression, which are marked with arrows. From the third panel, it can be observed that only negligible distortions were introduced. This is the key advantage of using the second column of U to embed watermarks.

Therefore, considering two facts: the first fact is that manipulation on the elements from $u_{1..m,2}$ creates less distortion than that on the elements from $u_{1..m,1}$. The second fact is that those created peaks in $u_{1..m,2}$ are robust. Therefore, it was decided to create a local peak in $u_{1..m,2}$ to represent a watermark bit. The rule of embedding a watermark bit is described as follows.

If the bit to embed is a '1', the value of the element $u_{e,2}$ is increased, where e is a user-defined value of position. The neighbouring elements at $u_{e\text{-}1,2}$ and $u_{e+1,2}$ are reduced

to a negligibly low value Th_{svd1} such that $u'_{e,2}$ is a peak value, where $u'_{e,2}$ represents the modified $u_{e,2}$.

Similarly, if the bit to embed is a '0', the value of the element $u_{e+1,2}$ is increased and the neighbouring elements at $u_{e,2}$ and $u_{e+2,2}$ are reduced to the same negligibly low value Th_{svd1}, such that $u'_{e+1,2}$ is a peak value. This embedding rule can be formulated by Equation (5.4).

$$\begin{cases} bit = 1 \ \ let\ u'_{e,2} > u'_{e-1,2}\ , u'_{e,2} > u'_{e+1,2}\ ; u'_{e-1,2} = Th_{svd1}\ , u'_{e+1,2} = Th_{svd1} \\ bit = 0\ let\ \ u'_{e+1,2} > u'_{e,2}\ ,\ u'_{e+1,2} > u'_{e+2,2}\ ;\ u'_{e,2} = Th_{svd1}\ , u'_{e+2,2} = Th_{svd1} \end{cases} \tag{5.4}$$

Here, Th_{svd1} was set to be 0 and $u'_{e,2}$ was set to be 0.5 so that $u'_{e,2}$ was a peak.

5.3.3 Complete signal reconstruction using overlapping window frame series

After embedding the watermarks, an inverse FFT will be applied to produce the time domain signal. However, in order to remove the effect of window function when producing the time domain signal, a Constant Overlap Add (COLA) constraint has to be satisfied [CA07], that is, the sum of the time domain window samples in each frame has to be equal to 1. For example, when using a Hanning window, where the hop size is equal to half of the frame size, the COLA constraint can be fulfilled.

A new procedure was developed to reconstruct the time domain signal by cancelling the window effect using overlapped window frame series. This procedure is outlined as follows:

1. Before embedding, compute the non-overlapping Hanning windowed FFT of the original signal $x_1(n)$, which results in a spectra $X_1(\omega)$.
2. $X_1(\omega)$ is manipulated to embed watermark bits, from which the modified spectra $X'_1(\omega)$ can be obtained.
3. During time domain signal reconstruction, $X'_1(\omega)$ will be firstly converted to time domain signals $x'_1(n)$ using the inverse FFT. Then, $x'_1(n)$, $x_1(n)$, $w_1(n)$ and $w_2(n)$ are combined together to reconstruct the time domain signal, where $w_1(n)$, $w_2(n)$ are the non-overlapping and overlapping window frame series respectively.

Assume the frame length l is 1024. $x_1(n)$, $w_1(n)$ and $w_2(n)$ are shown in (a-c) of Figure 5.3 respectively, where the x-axis denotes sample number, and the y-axis denotes amplitude.

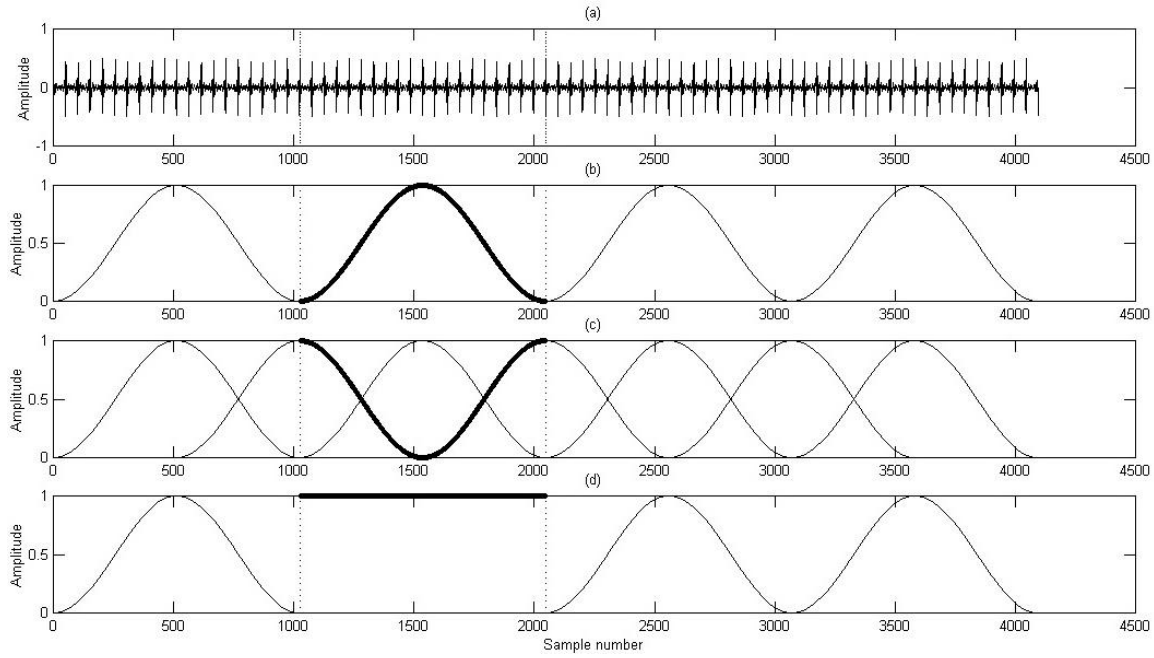


Figure 5.3 (a) $x_1(n)$ (b) $w_1(n)$: non-overlapping window frame series (c) $w_2(n)$: overlapping window frame series (d) Illustration of COLA principal

Take the second frame of the watermarked signal as an example to illustrate how to reconstruct its time domain signal.

$x_1(n_1)$, $x'_1(n_1)$ denote the second frame of $x_1(n)$ and $x'_1(n)$ respectively where $l+1 \leq n_1 \leq l*2$.

Firstly, $x'_1(n_1)$ can be rewritten as Equation (5.5), where $l+1 \leq n_{11} \leq \frac{l*3}{2}$, $\frac{l*3}{2}+1 \leq n_{12} \leq l*2$, τ_{svd} denotes the modification on the second frame time domain signal because of embedding the watermark.

$$\begin{aligned} x'_1(n_1) &= x_1(n_1)w_1(n_1) + \tau_{svd}(n_1) \\ &= x_1(n_{11})w_1(n_{11}) + x_1(n_{12})w_1(n_{12}) + \tau_{svd}(n_1) \end{aligned} \tag{5.5}$$

Based on COLA, Equation (5.6) and (5.7) can be derived, where $I = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{\frac{l}{2} \times 1}$ as shown between sample values of 1000 and 2000 in panel (d) of Figure 5.3.

$$w_1(n_{11}) + w_2(n_{11}) = I \tag{5.6}$$

$$w_1(n_{12}) + w_2(n_{12}) = I \tag{5.7}$$

Based on the above Equation (5.5)-(5.7), adding $x'_1(n)$, $x_1(n_{11})w_2(n_{11})$ and $x_1(n_{12})w_2(n_{12})$ element-wise, the second frame time domain data of watermarked signal can be reconstructed as shown below.

$$\begin{aligned} & x'_1(n_1) + x_1(n_{11})w_2(n_{11}) + x_1(n_{12})w_2(n_{12}) \\ & = x_1(n_{11})w_1(n_{11}) + x_1(n_{12})w_1(n_{12}) + \tau_{svd}(n_1) + \\ & x_1(n_{11})w_2(n_{11}) + x_1(n_{12})w_2(n_{12}) \\ & = x_1(n_{11})(w_1(n_{11}) + w_2(n_{11})) + x_1(n_{12})(w_1(n_{12}) + w_2(n_{12})) + \tau_{svd}(n_1) \\ & = x_1(n_{11}) + x_1(n_{12}) + \tau_{svd}(n_1) \\ & = x_1(n_1) + \tau_{svd}(n_1) \end{aligned} \tag{5.8}$$

$x_1(n_1) + \tau_{svd}(n_1)$ represents the second frame time domain data of the watermarked signal where the window effect has been cancelled completely.

One point worth noting is that an overlap based FFT transform cannot be used when embedding the watermark as it will have a negative impact on the watermarking accuracy. This can be explained as follows. Assume the spectrum of the frame *m* is modified to satisfy the embedding criteria. Its subsequent frame *m*+1 is also modified to

satisfy the embedding criteria. As an overlap exists between the spectrum of the frame m and m+1, it is likely that the spectrum modification applied to the frame m is distorted by that applied to the frame m+1. This will decrease the watermarking accuracy because the modification applied on each frame represents the embedded watermark information.

5.3.4 The watermark embedding process

A block diagram of the watermark embedding procedure is given in Figure 5.4. The spectrum of each frame of the original signal S is computed by FFT and then the magnitudes of spectrum are organized into a matrix A. CSVD is applied to this matrix to get three matrices U, S, V. The elements in the vicinity of row e in $u_{1..m,2}$ are modified to satisfy the embedding rule, followed by a reconstruction as a time domain watermarked signal S', as shown in Section 5.3.3. Note that the procedure described in Figure 5.4 only shows the case of embedding a bit '1'.

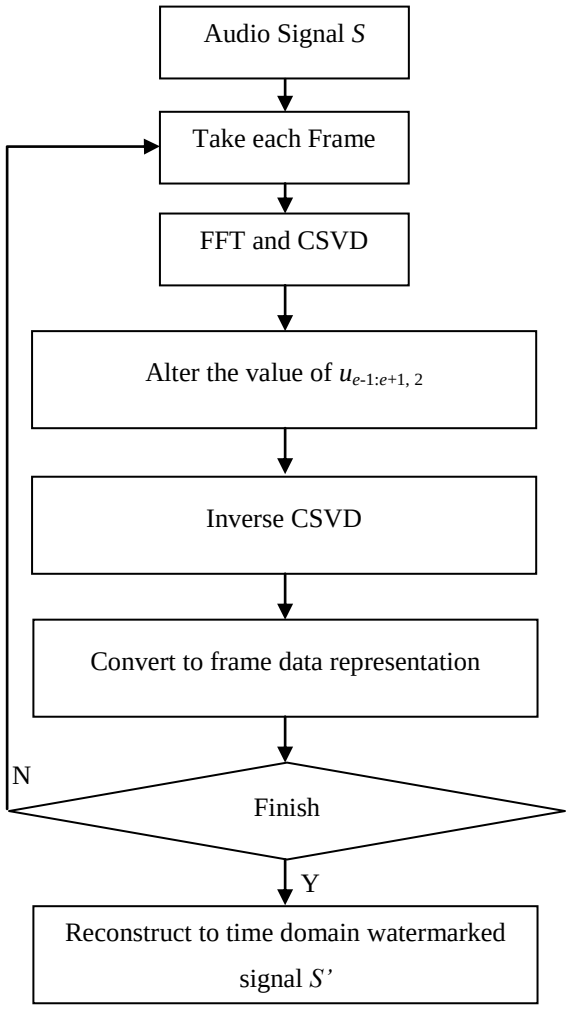


Figure 5.4 The flowchart of watermark embedding

5.3.5 The watermark detection process

The detection process can be described as follows: the spectrum of each frame of the watermarked signal is computed by FFT and then the magnitudes of spectrum are organized into a matrix *A'*. The CSVD is applied to this matrix to get three matrices *U'*,

S', *V'*. $u'_{e,2}$ and $u'_{e+1,2}$ are compared. If $u'_{e,2} > u'_{e+1,2}$, then the bit to be detected is a '1', otherwise the bit to be detected is a '0'.

5.3.6 Audible distortion detection and suppression

An initial experiment was performed to test the imperceptibility of this watermarking algorithm. Ten music files were randomly selected from a variety of genres. The results are plotted in Figure 5.5 where the x-axis denotes the file number and the y-axis denotes the ODG score.

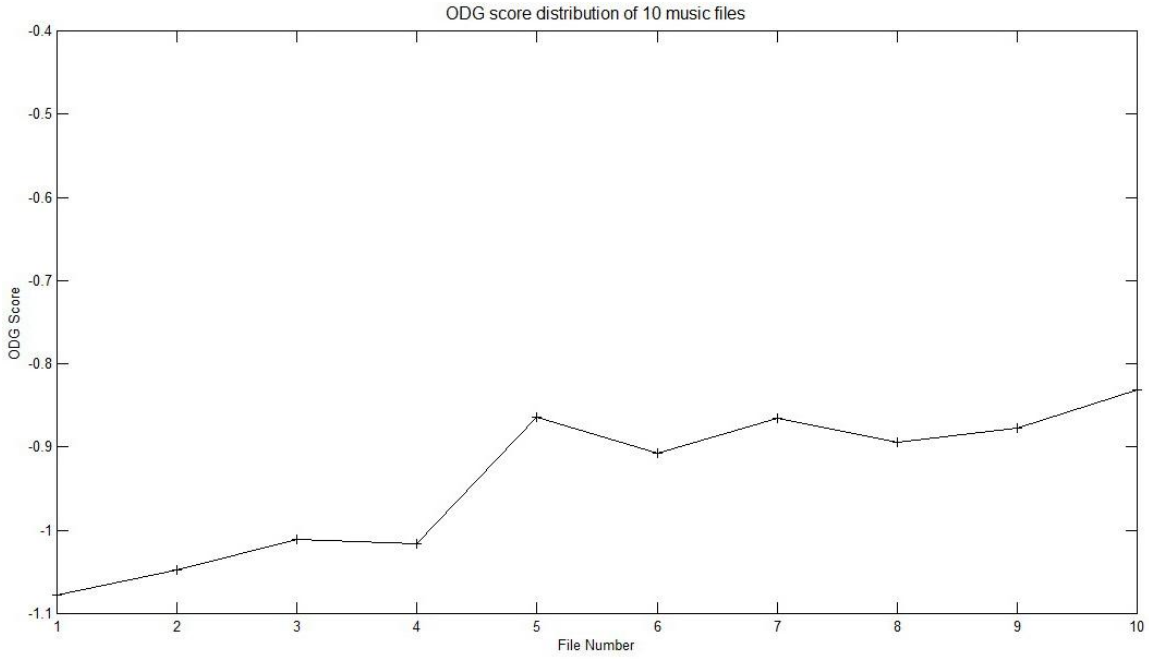


Figure 5.5 The ODG distribution of 10 music files

From Figure 5.5, it can be seen that ODG scores achieved fluctuate around -1. The mean ODG score is -0.9392. Thus, an improvement on the imperceptibility is needed.

An additional process was incorporated to improve the imperceptibility of this watermarking algorithm by checking the value of Noise to Mask Ratio (NMR). NMR has been used as a measurement of noise audibility [NV10, Arn02, Col99]. This section will firstly introduce the steps involved in deriving the NMR. The steps involved are [Pai00]:

1. The Power Spectral Density (PSD) is computed and the maskers are identified.
2. Decimation and reorganization of maskers is carried out to discard those inaudible maskers.
3. Individual masking thresholds are obtained, and then a global masking threshold is obtained.
4. The minimum masking threshold $LTmin$ is found.
5. The NMR, as per Equation (5.9), is computed for each sub-band p.

$$\mathrm{NMR}(p) = \mathrm{SMR}(p) - \mathrm{SNR}(p) \tag{5.9}$$

where the Signal to Mask Ratio (SMR) is defined for each sub-band as the ratio of signal energy to $LTmin$, and the SNR is defined for each sub-band as the ratio of signal energy to noise energy. Thus, the NMR of each sub-band can be rewritten as the difference between the noise energy E_{ns} and $LTmin$, that is,

$$\mathrm{NMR}(p) = \mathrm{E_{ns}}(p) - \mathrm{Ltmin}(p) \tag{5.10}$$

A negative NMR indicates that the noise energy is below $LTmin$, while a positive NMR indicates that the noise energy is above $LTmin$. As far as our watermarking scheme is concerned, the noise energy E_{ns} of each sub-band p is defined as follows:

$$\mathrm{E_{ns}}(p) = \max|\mathrm{E_o}(p,q) - \mathrm{E_w}(p,q)| \qquad q = 1{:}\,n \tag{5.11}$$

where q denotes the frequency bin number in each sub-band p, E_o denotes the original signal energy in dB and E_w denotes the watermarked signal energy in dB. The noise energy of each sub-band is determined by the largest noise introduced into each frequency bin of this sub-band.

Based on the experiments performed in [Arn02], it is known that by ensuring the NMR is less than -5 dB, any distortion introduced will be inaudible. More specifically, if the NMR exceeds -5 dB for a given frame, it indicates that the distortion may be audible and should be further reduced. Otherwise, no further suppression is required. The algorithm to identify and suppress these distortions is described as follows:

1. Calculate the NMR for each sub-band of each watermarked frame
2. Obtain position (p, q), representing the sub-band number and the frequency bin number respectively, from the sub-band of the frame where the corresponding NMR is greater than -5dB. If position (p, q) is found, go to step 3. Otherwise, repeat with the next sub- band from step 1.
3. The magnitude of the corresponding component in position (p, q) is tuned according to Equation (5.12) and (5.13).

$$E'_w(p,q) = E_o(p,q) - LTmin(p) + Th_{svd2} \tag{5.12}$$

$$A'_w(p,q) = 10^{\frac{1}{20}*(E'_w(p,q))} \tag{5.13}$$

where $E'_w(p,q)$ is the tuned energy of the frequency bin j in sub-band p, $A'_w(p,q)$ is the tuned magnitude of frequency bin q in sub-band p, and Th_{svd2} represents a threshold. Th_{svd2} is set to ensure that the corresponding NMR is tuned to be -5 dB.

This additional process aiming to improve the imperceptibility can be depicted in Figure 5.6. The procedure can be described as follows from Figure 5.6: the watermarked signal S' is analyzed by FFT on a frame-by-frame basis. Then, the audible distortion of each frame will be suppressed according to the steps 1-3 as listed above. Finally, the time domain watermarked signal S'' will be reconstructed.

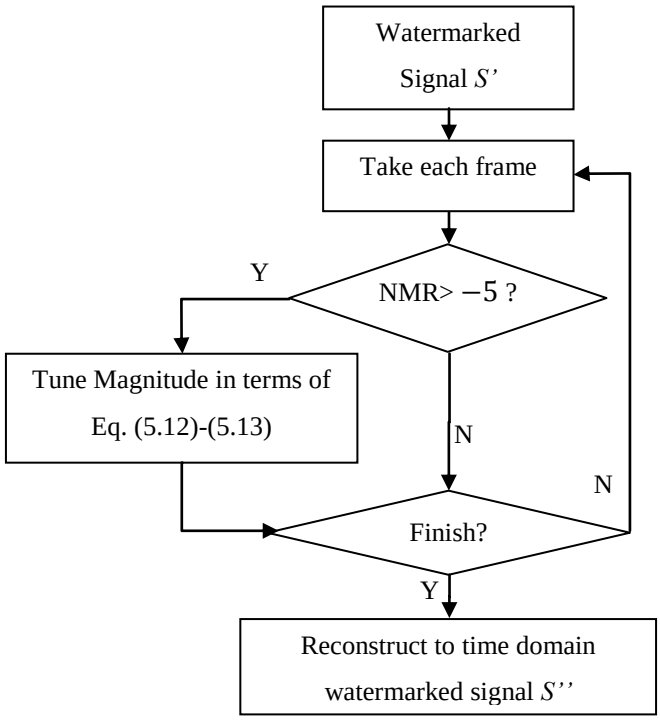


Figure 5.6 The additional process of improving the imperceptibility

5.3.7 Experimental validation

The ten same signals as those selected in Section 5.3.6 were used by way of an example to show the effect of incorporating this audible distortion suppression process. The results are plotted in Figure 5.7 where the x-axis denotes the file number, and the y-

axis denotes the ODG score and *Precision* in the upper panel and lower panel respectively.

From Figure 5.7, it can be seen that the ODG score of each music file has been improved, and the average improvement is 0.38. However, the *Precision* of each music file has been decreased, and the average decrease is 0.035, as shown in Figure 5.7 (b). It is worth noting that this added procedure results in an increased computational cost at the embedding stage. The decision to include this can be made dependent on the application domain to which the watermarking algorithm is being applied. If perceptual transparency is a primary concern (for example, in copyright protection), then the audible distortion suppression would outweigh any reduction in the watermark accuracy and the additional computational cost of embedding.

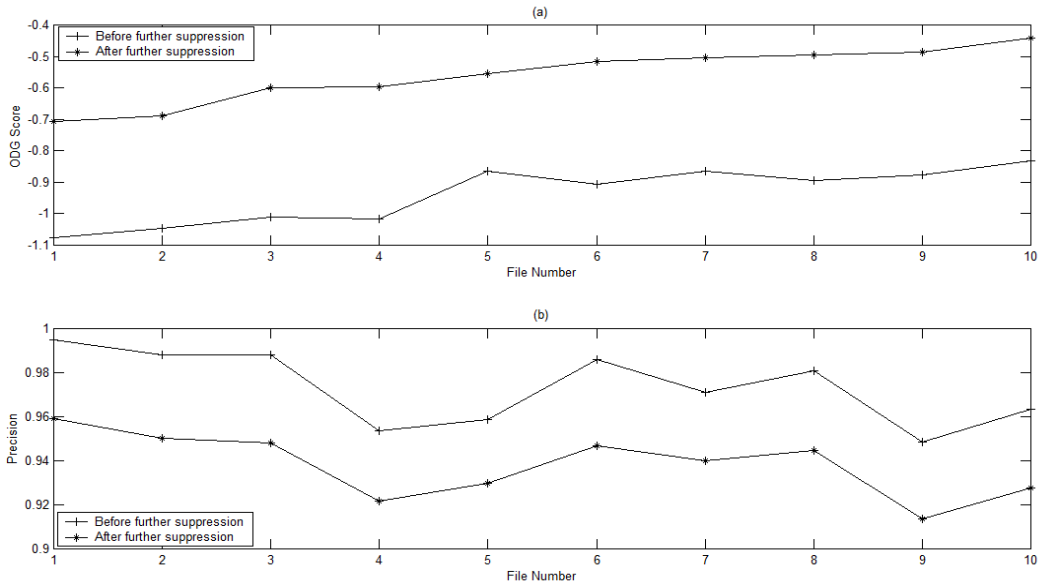


Figure 5.7 (a) ODG score before and after applying further suppression (b) Precision before and after applying further suppression

5.4 Evaluation

In this section, the perceptual transparency of the proposed watermarking scheme, along with its robustness and capacity were evaluated. Twenty five music files of different genres were randomly selected. Each file was sampled at 48000 Hz. Twenty five watermark bit sequences were randomly generated. In this experiment, four bits were embedded within each frame.

5.4.1 Imperceptibility

The imperceptibility was evaluated based on the SNR, SNR_{seg} and ODG, which are described as follows.

5.4.1.1 SNR

The SNR and SNR_{seg}, as defined in Equation (2.1) and (2.2), were calculated. These values can be used to compare the proposed scheme with many others. The distribution of SNR and SNR_{seg} are shown in Figure 5.8. From Figure 5.8, it can be seen that all SNR values are above 20 dB, which conforms to the IFPI standard. The mean of the SNR and SNR_{seg} are 27.2304 dB and 27.5826 dB respectively.

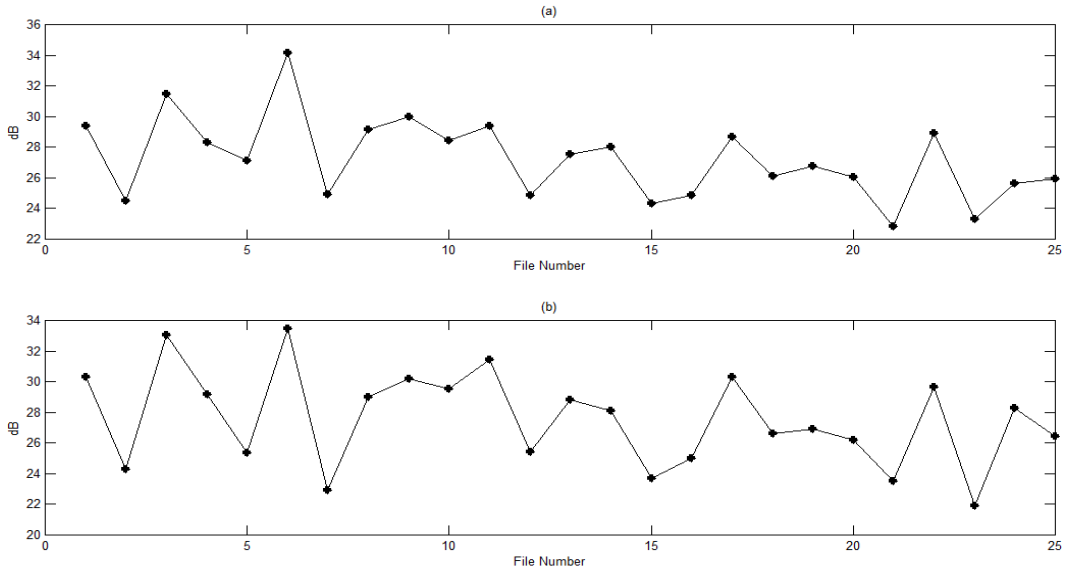


Figure 5.8 (a) SNR distribution for all the 25 music files (b) SNRseg distribution for all the 25 music files

5.4.1.2 ODG

A PEAQ test was carried out on 25 music files and the distribution of ODG scores was shown in Figure 5.9. From Figure 5.9 it can be seen that the ODG scores for all 25 files are between 0 and -1. The mean ODG score is -0.6352, which is in the imperceptible range [BO08].

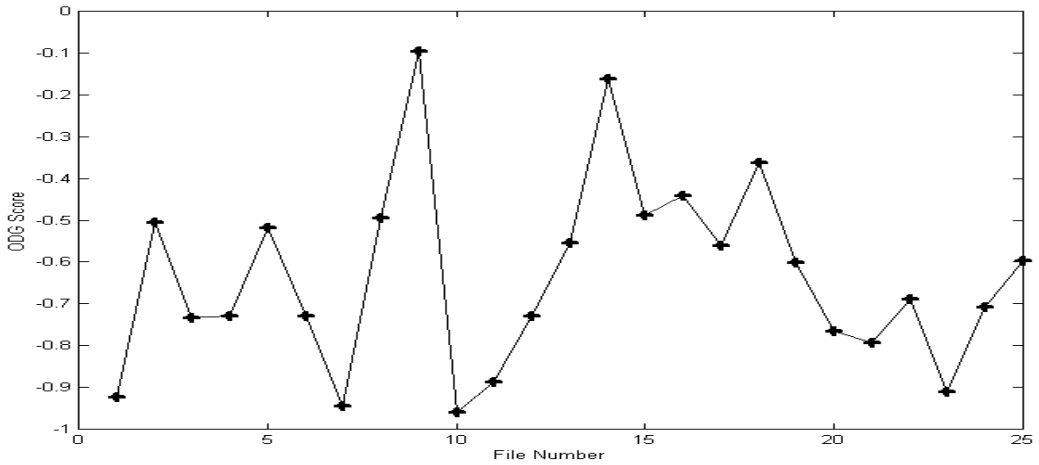


Figure 5.9 ODG scores distribution for all the 25 music files

Another experiment was conducted to verify the performance of imperceptibility when embedding watermark bits by manipulating the elements from the first column of *U*. The ODG distribution is shown as Figure 5.10:

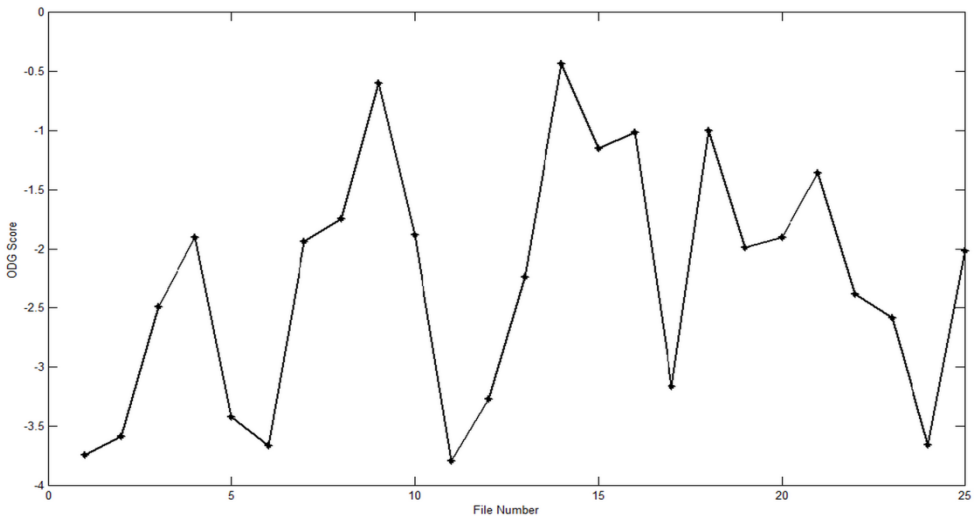


Figure 5.10 ODG scores distribution by manipulating the first column of *U*

From Figure 5.10, it can be seen that the ODG score is not acceptable for most test files, with an average ODG score of -2.27. This demonstrates clearly that

manipulating the second column of *U* achieves a much better imperceptibility than manipulating the first column of *U*.

5.4.2 Robustness

The attacks studied were MP3 compression, noise removal, noise adding, lowpass filtering and highpass filtering.

5.4.2.1 Robustness against MP3 Compression

The robustness of the proposed watermarking scheme against MP3 compression at different bit rates is shown in Figure 5.11. It can be seen that the proposed scheme produces watermarked signals that can survive MP3 compression from 64 kbps to 160 kbps to a high degree of *Precision*. When the bit rate is 128 kbps or higher, the *Precision* is almost unchanged when compared to that in the absence of any attack, while a 2% to 3% loss in *Precision* is encountered when the compression bit rate is 64 kbps.

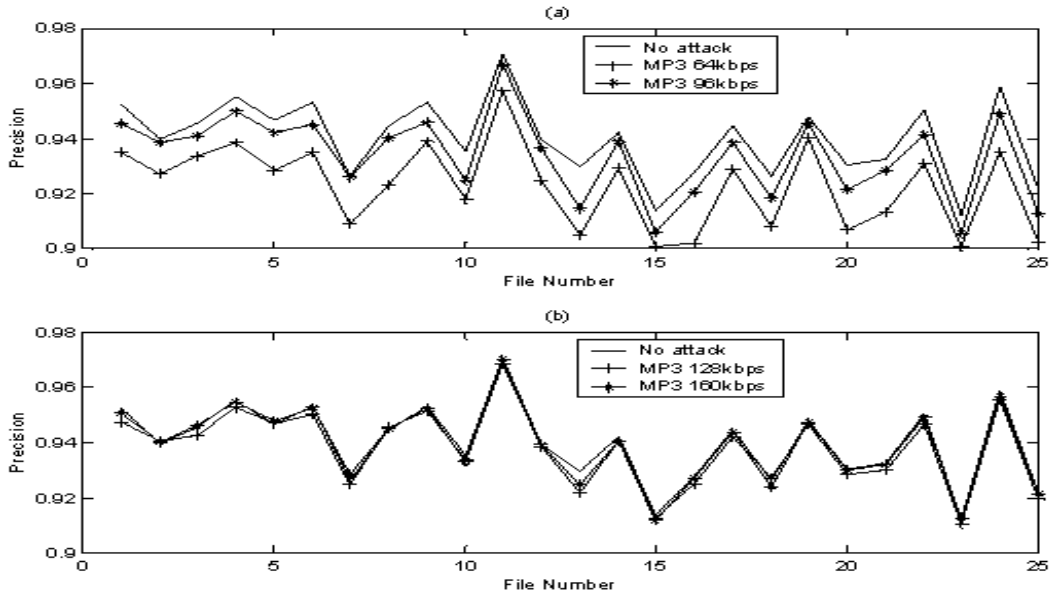


Figure 5.11 (a) Robustness against MP3 attacks at 64 kbps, 96 kbps (b) Robustness against MP3 attacks at 128 kbps, 160 kbps

5.4.2.2 Robustness against other selected attacks

The robustness of the scheme against attacks, which were listed as below, was also evaluated.

1. Noise removal: this was provided by the Audacity tool [Aud10]. The level of noise reduction is 24dB.
2. Noise adding: a particular level of AWGN was added to the signal whose power is 0.01. The reason of adding this level of AWGN is the same as that given in Section 4.11.2.3 of Chapter 4.
3. Lowpass filtering: the roll-off is 6dB per octave and the cut-off frequency is 8000 Hz.
4. Highpass filtering: the roll-off is 6dB per octave and the cut-off frequency is 2000 Hz.

The *Precision* achieved following there four different attacks was shown in Figure 5.12. As can be seen from Figure 5.12, the *Precision* achieved is quite high, generally above 85%.

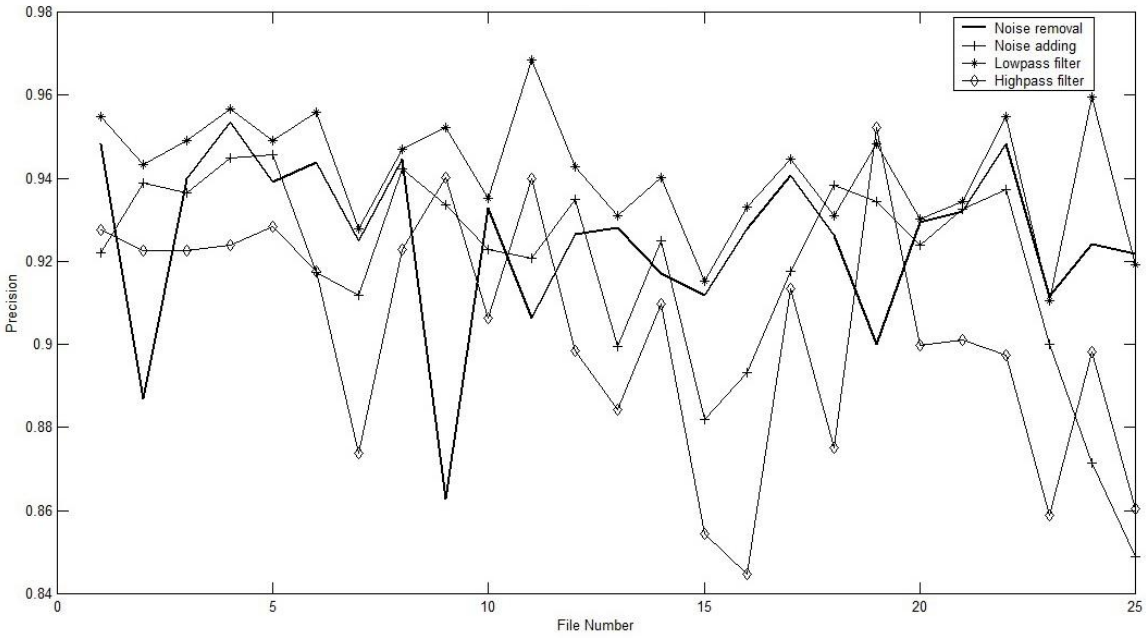


Figure 5.12 *Precision* achieved after four different attacks without using repetition

The mean and standard deviation of the *Precision* for all attacks tested is shown in Table 5.1. From the table, it can be seen that this scheme achieves a strong robustness against the different attacks listed. The robustness against all the listed attacks is around 93%, which can be thought to be strong enough as it approaches to the accuracy of 94% for no attack. It is reasonable to assume that if the accuracy is improved, the robustness will be improved simultaneously.

Table 5.1 Mean and standard deviation of *Precision* against different attacks

Attacks	Mean	Standard deviation
No attack	0.9400	0.0143
MP3 64 kpbs	0.9229	0.0154
MP3 at 96 kpbs	0.9338	0.0152
MP3 at 128 kpbs	0.9381	0.0143
MP3 at 160 kpbs	0.9393	0.0145
Noise removal	0.9251	0.0206
Noise adding	0.9083	0.0217
Lowpass filtering	0.9400	0.0145
Highpass filtering	0.9030	0.0288

5.4.2.3 Using the repetition to increase the robustness

The *Precision* of the same 25 music files after incorporating the 'repetition' process is distributed as Figure 5.13, where the repetition time is 5.

The $Precision_{mean}$ of the scheme after incorporating the 'repetition' process was 99.23% without any attack and 98.78% after MP3 compression at 64 kbps. The respective standard deviations were 0.0118 and 0.0138. Therefore, the *Precision* in the presence of a 64 kbps MP3 compression attack has been improved by around 6% after incorporating the 'repetition' process. The only disadvantage of incorporating the 'repetition' process is that the capacity is decreased.

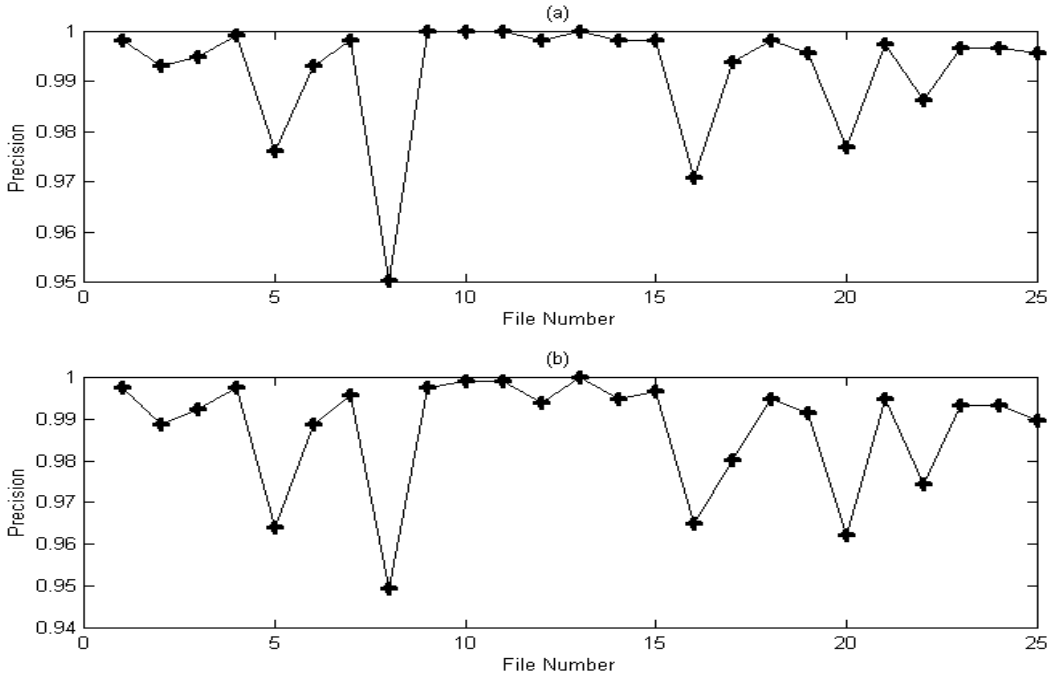


Figure 5.13 (a) *Precision* distribution of 25 music files without using repetition (without any attack) (b) *Precision* distribution of 25 music files after using repetition (after 64 kbps MP3)

5.4.3 Capacity

As for the proposed scheme, embedding four bits per frame with length of 1024 samples, the capacity is about 187 bps. This is a comparatively high capacity when compared to the audio watermarking schemes described in [BSD10, LC06, YK03, KM03, TSKN02, MT03, CS04, LXL06, MT05, XKH08, EB09].The capacity of the scheme can be improved by embedding more than four bits in each frame, that is, modifying more elements in the second column of *U*. However, this will have a negative impact on the imperceptibility.

5.5 Comparison with other algorithms

The performance of the proposed audio watermarking algorithm was compared with other typical SVD based audio watermarking algorithms and the algorithm proposed in

Chapter 4. This comparison is shown in Table 5.2. The ODG score was mapped to MOS score according to Table 2.3. From Table 5.2, it can be found that the proposed algorithm achieved a better overall performance than other SVD watermarking based algorithms. Its robustness becomes extremely high after incorporating the 'repetition' process.

Compared with the algorithm proposed in Chapter 4, this algorithm achieved a much higher capacity. In addition, the robustness of this algorithm is more independent from each specific file. However, the imperceptibility of this proposed algorithm should be further improved.

Table 5.2 Performance comparison of audio watermarking schemes

	Method	**Capacity**	**Blind**	**Robustness to 64 kbps MP3**	**MOS**	**SNR**
Proposed	SVD	187	Yes	Yes	4.36	27.23
[BSD10]	SVD	45.9	Yes	Yes	4.46	24.37
[AA10]	SVD	n/a	No	n/a	5.0	28.55
[OSM05]	SVD	n/a	No	n/a	4.7	n/a
CSPE [Chapter 4]	CSPE	24	Yes	Yes	4.65	36.05

5.6 Summary

The audio watermarking scheme presented in this chapter was based on transforming the spectrum of each frame of the signal using the CSVD. Modifications were then performed on the *U* matrix resulting from the CSVD to embed watermark bits. Experiments were carried out and demonstrated that the proposed blind scheme was imperceptible, robust against MP3 compression and other attacks. Finally, the scheme had a high data capacity of about 187 bps.

Chapter 6 Conclusion and Future Directions

6.1 Introduction

This final chapter is divided into two sections. It offers a set of general conclusions made from the results presented in the earlier chapters and outlines some specific directions for future work.

6.2 Conclusions

The main goal set out at the beginning of this thesis was to develop a new audio watermarking algorithm, which should be imperceptible, robust, computationally efficient and should have a high capacity. As a result of my research, two new audio watermarking algorithms were developed from two different perspectives.

The first algorithm was based on the CSPE, which is a super-resolution spectral analysis tool. The second algorithm was based on the SVD, which is a widely used mathematical tool for matrix decomposition.

The reason for developing a CSPE based algorithm derived from the fact that FFT based watermarking algorithms had achieved good performances. However, there is room for improvement, for example, the imperceptibility can be better, the embedding process could be more secure and the embedding rule could be refined to improve the robustness.

In terms of improving the imperceptibility, one method investigated was the introduction of a super-resolution spectral analysis tool so that spectral components could be manipulated in the time domain. This resulted in reducing the perceptual distortions. The outcome of the authors' experimentation with the CSPE showed it to be highly

accurate and computationally efficient. However, it was found that components appearing over a proportion of a frame could not be identified. This issue had to be addressed before developing a watermarking algorithm based on CSPE. In Chapter 3, an apodization function was employed to solve this problem. Additionally, a set of values for coefficients α_1 and β_1 were derived which led to the uncovering of the previously unidentifiable components appearing over different proportions of a frame. Based on this, a watermarking algorithm was proposed and then applied to synthetic signals.

This algorithm was further applied to the real audio signals and it was found to have an extremely low accuracy. The main reason for this was a *component obscurity* phenomenon, which resulted from magnitude reduction. In order to use this phenomenon to our advantage and thus achieve a high accuracy, a different embedding and detection rule based on the bin locations instead of magnitude values was developed. By utilizing this rule, a shortcoming of the FFT based watermarking algorithms, that is, its vulnerability against any process which could interfere with magnitude, could also be overcome. Furthermore, a component verification process was developed and incorporated to guarantee that the candidate bins identified at the embedding stage can also be identified at the detection stage. A dynamic embedding position generation process was also developed to overcome another shortcoming of the FFT based watermarking algorithms, that is, the embedded watermarks were easy to remove. Finally, perceptual distortions introduced were dealt with by post-processing. The experimental results showed that this algorithm achieved an extremely high imperceptibility with an

ODG score of about 0, and a high accuracy with a *Precision* of around 100%. This makes the algorithm extremely suitable for steganography applications where the imperceptibility of the hidden information is the primary concern.

In Chapter 4, the algorithm was further examined and it was found through experimentation that its robustness was very low. A new observation was discovered relating to the high robustness of the peaks in the CSPE spectrum after attack. Based on this, watermarks were then embedded by manipulating the candidate peaks. A new rule was devised based on the parity of the bin location of the candidate peak. Then, a novel threshold scheme was devised and the impact of threshold values on the watermarking performance was examined. Furthermore, a peak sharpness measure and an error correction scheme were incorporated to improve the robustness. The performance was then verified by using a large random selection of music tracks from different genres. As shown from the experimental results, this algorithm achieved an acceptable imperceptibility, robustness, capacity and computational efficiency.

Another algorithm based on SVD was developed for the following two reasons: firstly, the proposed CSPE based algorithm did not achieve a sufficiently high capacity. Secondly, it was known that SVD was capable of achieving a high robustness. In this algorithm implementation, watermarks were embedded based on creating peaks in the second column of U. Furthermore, a psychoacoustic model was incorporated to suppress the audible distortion. This algorithm achieved a high robustness and capacity, with an acceptable imperceptibility.

In summary, the two algorithms were found to satisfy the general criteria for good watermarking performance, as outlined earlier, and are of use in different applications. Based on this research, it was discovered that a high robustness could be obtained by utilizing data that is known to be inherently robust against certain attacks. This data can be identified through the use of signal processing analysis or mathematical analysis of the original content. The embedding and detection rules should be defined based on those features that are not vulnerable to attacks. Psychoacoustic knowledge can be used so that data can be manipulated in a way that does not affect the perceptual transparency significantly.

It is definite that there is scope for future work on these two algorithms to improve their performance. Some suggestions will be offered in the following section.

6.3 Future work

As for the CSPE based watermarking algorithms proposed in Chapter 4, there is some work that can be carried out in future. For example, the development of a method to determine the reference value *r* dynamically would be beneficial, as it was seen that using a uniform value for *r* does not always achieve a high robustness and imperceptibility for each individual music track. This is due to the different energy distributions across frequency regions. Alongside this, an additional process should be incorporated to detect those frames with a low energy, so that they are not used to embed any watermark bit.

A further improvement would be to use a Just Noticeable Distortion (JND) so that a manipulated peak's magnitude would lie below the hearing threshold. However, this

might have an adverse effect on the computational efficiency of the algorithm. Thus, whether to incorporate this improvement should be dependent on the specific application.

Another improvement would be to embed more bits in one frame to increase the capacity. This would involve selecting more reference values in each frame. However, this improvement can also be used as obfuscation so that it is difficult for an attacker to guess the exact watermarking embedding positions.

As for the SVD based watermarking algorithm, one improvement would be to use a JND to determine the optimum values for the manipulated elements in the second column of U, so that the imperceptibility could be improved. Furthermore, the capacity can be improved by embedding watermarks in S.

For both algorithms, a synchronization process should be added to improve the robustness against time domain attacks [MRF10]. The synchronization process is used to find the start position of watermark bits. In addition, the watermark information can be encrypted before embedding to enhance security [Bla10]. Furthermore, using chaotic functions instead of pseudo random functions for the generation of watermarks has proved advantageous. For example, the watermarks generated are more robust to the common attacks [MK05].

Furthermore, combining the CSPE with SVD can improve the watermarking performance. For example, the peaks in the CSPE spectrum are derived and then organized as a matrix, followed by the decomposition of this matrix using SVD. Then,

the second column of the matrix U and the matrix S can be manipulated to embed watermark bits.

The multi-level watermarking should be further researched. With multi-level watermarking, different watermarking algorithms can be applied to the same host. One reason for this is that each watermarking algorithm has its own advantages and disadvantages, and by combining these watermarking algorithms together, their advantages can be combined, potentially improving the overall watermarking performance. For example, if one watermarking algorithm is robust against time domain attacks, the other watermarking algorithm might be selected to be robust against frequency domain attacks. In this way, if a time domain attack occurs, the first watermarking algorithm can survive, while if only frequency domain attacks occur, the second watermarking algorithm can survive. Additionally, by applying multi-level watermarking algorithms to the same host, different targets can be achieved simultaneously such as tamper-proof and illicit distributor tracking.

The formal security framework for watermarking scheme needs further investigation. The reason for this is that if a watermarking system is to be widely deployed in real world applications, the security of a commercial application should be guaranteed for its users.

Furthermore, a mathematical framework needs to be designed to objectively evaluate the robustness and to calculate the maximum capacity of the watermarking algorithm. With this framework, the robustness can be evaluated objectively which

provides a fairer comparison between different watermarking algorithms, and provides a more useful feedback for improving the robustness.

Finally, benchmarks of watermarking performance for different applications are worth investigating. With these benchmarks, it would be easy to know whether a watermarking algorithm is appropriate for certain application scenarios.

References

[AA10] A.H. Ali and M. Ahmad, "Digital Audio Watermarking Based on the Discrete Wavelets Transform and Singular Value decomposition," *European Journal of Scientific Research*, ISSN 1450-216X vol.39(1), pp.6-21, 2010.

[AAC11] http://manual.audacityteam.org/index.php?title=AAC_Export_Options, 2011.

[AH04] A. ARNAB and A. HUTCHISON, "Digital rights management – a current review," *Departmental technical report*, no. cs04-04-00, University of Cape Town, 2004.

[AP76] H.C. Andrews and C.L. Patterson, "Singular value decomposition (SVD) image coding," *IEEE Transactions on Communications*, vol. 24, pp. 425–432, 1976.

[AP98] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography", *IEEE J. Select. Areas Commun.*, vol. 16, pp. 474–481, 1998.

[Arn00] M. Arnold, "Audio watermarking: Features, applications and algorithms," *IEEE Int. Conf. Multimedia Expo 2000*, vol. 2, pp. 1013–1016, 2000.

[Arn01] M. Arnold, "Attacks on digital audio watermarks and countermeasures," *IEEE Int. Conf. WEB Delivering of Music*, 2003, 1-8.

[Arn02] M. Arnold, etal, "Quality evaluation of watermarked audio tracks," *INTERNATIONAL Society of Photo-Optical Instrumentation Engineers*,

Bellingham, WA, 2002.

[AS02] M.Arnold and K. Schilz, "Quality evaluation of watermarked audio tracks," *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents IV*, San Jose, CA, USA, 2002.

[AS04] M.Abe and J. O. Smith, "Design Criteria for the Quadratically Interpolated FFT Method: Bias due to Interpolation," *Technical Report STAN-M-114*, Dept. of Music, Stanford University, October 2004.

[Aud10] http://audacity.sourceforge.net/download/ , 2010.

[BDG03] E. Boutillon, J. Danger and A. Gazel, "Design of High Speed AWGN Communication Channel Emulator," *Analog Integrated Circuits and Signal Processing*, vol. 34(2), pp. 133-142, 2003.

[BGML96] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35 (3/4), pp. 313–336, 1996.

[BHT63] B.P. Bogert, M.J.R. Healy and J.W. Tukey, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-cepstrum and Saphe Cracking," in Proceedings of the Symposium on Time Series Analysis, by M. Rosenblatt, Wiley N.Y. 1963, pp. 209-243.

[Bla07] J.Blackledge, "Digital Watermarking and Self-Authentication using Chirp Coding," *ISAST Transactions on Electronics and Signal Processing*, ISSN 1797-2329, vol. 1 (1), pp. 61 – 71, 2007.

[Bla10] J.Blackledge, "Information Hiding using Stochastic Diffusion for the

Covert Transmission of Encrypted Images," *ISSC2010*, UCC Cork, Ireland, 2010.

[BO08] J.Blackledge and F.Omar, "Audio Data Verification and Authentication using Frequency Modulation based Watermarking," *International Society for Advanced Science and Technology, Journal of Electronics and Signal Processing* , ISSN 1797-2329, vol. 3(2) , pp. 51 – 63 , 2008.

[Bos86] B. Bose, "Burst Unidirectional Error-Detecting Codes," *IEEE Transactions on Computers*, vol. 35, no.4, pp.350-353, April 1986.

[BP10] J. Blackledge and N. Ptitsyn, "Encryption using Deterministic Chaos," *ISAST Transactions on Electronics and Signal Processing*, vol. 4(1), pp. 6 – 17, 2010.

[BSD10] V. K. Bhat and I. Sengupta, "A. Das, An adaptive audio watermarking based on the singular value decomposition in the wavelet domain," *Digital Signal Processing*, vol. 20(6), pp. 1547-1558, ISSN 1051-2004, December 2010.

[CA07] T. Christos and F. Andreas, "Real Time Spatial Representation of Moving Sound Sources," *The 123rd AES convention*, New York, NY, USA, pp. 72-79, Oct. 2007.

[CAM00] T. Cedric, R. Adi, and I. Mcloughlin, "Data concealment in audio using a non- linear frequency distribution of PRBS coded data and frequency-domain LSB insertion," *IEEE Region 10 International Conference on*

Electrical and Electronic Technology, Kuala Lumpur, Malaysia, pp. 275-278, September 2000.

[CC07] D. Câmpeanu, A. Câmpeanu, "PEAQ – an Objective Method to Assess the Perceptual Quality of Audio Compressed Files", *Proceedings of International Symposium on System Theory*, SINTES 12, Craiova, România, 2005 October.

[CCL07] C. C. Chang, Y. C. Chou and T. C. Lu, "A semi-blind watermarking based on discrete wavelet transform," In *Proceedings of the 9th international conference on Information and communications security*, 2007.

[CH11] J. C. Chuang and Y. C. Hu, "An adaptive image authentication scheme for vector quantization compressed image," *Journal of Visual Communication and Image Representation*, In Press, Corrected Proof, ISSN 1047-3203, Available online 2 April 2011.

[Cha02] D. Chandra, "Digital image watermarking using singular value decomposition," *Proceedings of the IEEE 45th Midwest Symposium on Circuits and Systems*, vol. 3, pp. 264–267, August 2002.

[Che10] X. Chen, "Music and Image applications of Mobile Phone Serious Game," *Journal of Information and Communication Technology*, vol. 3(2), pp. 82-87, 2010.

[CH00] M. Cerna and A.F. Harvey, "The Fundamentals of FFT-Based Signal Analysis and Measurement", *National Instruments Corporation*,

Application Note 041, 2000.

[CHL06] C.C. Chang, Y.S. Hu, and T.C. Lu, "A watermarking-based image ownership and tampering authentication scheme," *Pattern Recognition Letters*, vol. 27 (5), pp. 439–446, 2006.

[CKL+96] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia," *Inform. Hiding Workshop*, Cambridge, England, pp. 147–158, June 1996.

[CKL+97] I. J. Cox, J. Kilian, and T. Leighton, "T. Shammoon, Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Process*, vol. 6(12), pp. 1673-1687, 1997.

[CMB02] I. J. Cox, M. L. Miller, and J. A. Bloom, "Digital Watermarking," *Mo -rgan Kaufmann*, London, 2002.

[CMB+07] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, and T. Kalker, "Digital Watermarking and Steganography," *Morgan Kaufmann Publishers*, San Francisco, 2007.

[Col99] C. COLOMES, et al., "Perceptual Quality Assessment for Digital Audio: PEAQ – the new ITU Standard for Objective Measurement of Perceived Audio Quality," *Proc. Of the AES 17th International Conference*, Florence Italy, pp. 337-351, 1999.

[Cri89] R.J. Crinon, "Sinusoid parameter estimation using the fast Fourier transform," *IEEE International Symposium on Circuits and Systems*, vol. 2,

pp.1033-1036 , 1989.

[CS04] N. Cvejic and T. Seppänen, "Spread spectrum audio watermarking using frequency hopping and attack characterization," *Signal Processing*, vol. 84 (1), pp. 207-213, 2004.

[CS05] N. Cvejic and T. Seppänen, "Increasing robustness of LSB audio steganography by reduced distortion LSB coding," *Journal of Universal Computer Science*, vol. 11(1), pp. 56-65, 2005.

[CT65] J. Cooley and J. Tukey, "An algorithm for the machine calculation of the complex Fourer series," *Math. Comput.*, vol. 19, pp. 297-301, 1965.

[CTL05] C.C. Chang, P. Tsai, and C.C. Lin, "SVD-based digital image watermarking scheme," *Pattern Recognition Letters*, vol. 26, pp. 1577-1586, 2005.

[Cve04] N. Cvejic, "Algorithms for Audio Watermarking and Steganography," *Dissertation*, University of OULU, 2004.

[Cve07] N. Cvejic, "Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks," *IGI Global*, August 7, 2007.

[CW01] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47 (4), pp. 1423–1443, 2001.

[CW09] W.C. Chen and M.S. Wang, "A fuzzy c-means clustering-based fragile

watermarking scheme for image authentication," *Expert Systems with Applications 36 (2 Part 1)*, pp. 1300–1307, 2009.

[CYH+07] K. Chung, W. Yang, Y. Huang, S. Wu, and Y. Hsu, "On SVD-based watermarking algorithm," *Applied Mathematics and Computation*, vol. 188(1), pp. 54-57, May 2007.

[CYH+07] K. Chung, W. Yang, Y. Huang, S. Wu, and Y. Hsu, "On SVD-based watermarking algorithm," *Applied Mathematics and Computation*, vol. 188(1), pp. 54-57, May 2007.

[DD03] G. Doërr and J.L. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Communication*, vol. 18(4), pp. 263-282, Apr. 2003.

[DKK10] P. K. Dhar, M. L. Khan and J. M. Kim, "A New Audio Watermarking System using Discrete Fourier Transform for Copyright Protection," *International Journal of Computer Science and Network Security*, vol.10 (6), June 2010.

[DLS02] J. Dittmann, A. Lang, and M. Steinebach, "Stirmark benchmark: Audio watermarking attacks based on lossy compression," *in Proceedings of SPIE* vol. 4675, 19 - 25 January 2002..

[DML+06] J. Dittmann, D. Megias, A. Lang and J. H. Joancomarti, "Theoretical framework for a practical evaluation and comparison of audio watermarking schemes in the triangle of robustness, transparency and capacity," *Transactions on Data Hiding and Multimedia Security*, Lecture

Notes in Computer Science, Springer, Berlin, vol. 4300, pp. 1–40, 2006.

[DP09] A. Deshpane and K.M.M.Prabhu, "A substitution by interpolation algorithm for watermarking audio," *signal processing*, Elsevier, vol. 89(2), pp. 218-225, 2009.

[DPH93] J. Deller, J. Proakis, and J. Hansen, "Discrete-Time Processing of Speech Signals," *Macmillan*, 1993.

[DSLZ04] J. Dittman, M. Steinebach, A. Lang, and S. Zmudzinski, "Advanced audio watermarking benchmarking," *In Proceedings of the IS&T/SPIE's 16th Annual Symposium on Electronic Imaging*, vol.5306, San Jose, CA, US, January 2004.

[DWW03] S. Dumitrescu, X. Wu and Z. Wang, "Detection of lsb steganography via sample pair analysis," *IEEE Transactions on Signal Processing*, vol. 51(7), pp. 1995–2007, 2003.

[EB09] E. Ercelebi and L. Batakcil, "Audio watermarking scheme based on embedding strategy in low frequency components with a binary image," *Digital Signal Process*, vol. 19 (2), pp. 265–277, 2009.

[Edw92] T. Edwards, "Discrete Wavelet Transforms: Theory and Implementation," *Technical Report*, Stanford University, September 1992.

[EE05] "Watermarking," http://www.elec.qmul.ac.uk/mmv/watermarking.html, Queen Mary University of London Electronic Engineering. 2005.

[ES09] Y. Erfani and S. Siahpoush, "Robust audio watermarking using improved

TS echo hiding," *Digital Signal Processing*, vol. 19, pp. 809-814, 2009.

[FGD01] J. Fridrich, M. Goljan, and R. Du, "Distortion-free Data Embedding," *Lecture Notes in Computer Science*, vol. 2137, pp. 27-41, 2001.

[FGD01a] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE Multimedia*, vol. 8(4), pp. 22–28, 2001.

[FGD02] J. Fridrich, M. Goljan, and R. Du, "Lossless Data Embedding – New Paradigm in Digital Watermarking," *Applied Signal Processing*, vol. 2002(2), pp. 185-196, February 2002.

[FH02] K. Fitz and L. Haken, "On the Use of Time-Frequency Reassignment in Additive Sound Modeling," *Journal of the AES*, vol. 50, no. 11, pp. 879-893, Nov. 2002.

[FIK06] R. Fujimoto, M. Iwaki and T. Kiryu, "A method of high bit-rate data hiding in music using spline interpolation," *Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Pasadena, pp. 11–14, December 2006.

[Fil07] J. Filipe, "Informatics in Control, Automation and Robotics II," *Springer London*, ISBN 978-1-4020-5625-3, 2007.

[FL08] M. Fan and S. Li, "Restudy on SVD-based watermarking scheme," *Applied Mathematics and Computation*, vol. 203(2), pp. 926-930, Sept. 2008.

[FLK06] R. Fujimoto, M. Lwaki and T. Kiryu, "A method of high bit-rate data

hiding in music using spline interpolation," *Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing Pasadena* , pp. 11–14, 2006.

[FM09] M. Fallahpour and D. Megas, "High capacity audio watermarking using FFT amplitude interpolation," *IEICE Electronics Express*, vol. 6 (14), pp. 1057–1063, 2009.

[FM10] M. Fallahpour and D. Megías, "DWT-based high capacity audio watermarking," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, ISSN.0916-8508, pp. 331-335, 2010.

[Gei90] W. A. Geisel, "Tutorial on Reed-Solomon Error Correction Coding," *NASA Technical Memorandum* , No. 102162, August 1990.

[GH99] F. P. Gonzalez and J. R. Hernandez, "A tutorial on digital watermarking Security Technology," *IEEE 33rd Annual 1999 International Carnahan Conference on Security*, pp. 286 – 292, Oct. 1999.

[GLB96] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," *Proceedings of the 1st Information Hiding Workshop LNCS*, vol. 1174, pp. 295–315, 1996.

[Gol11] http://www.goldwave.com last accessed: June 17 2011.

[Gri08] Dawn Griffiths, *Head First Statistics,* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2008.

[GS06] R. Garcia and K. Short, "Signal analysis using the Complex Spectral Phase Evolution (CSPE) method," *AES Convention 120*, Paris, France, May 2006.

[GZE03] E. Ganic, N. Zubair, and A. Eskicioglu, "An optimal watermarking scheme based on singular value decomposition," *Proceedings of the IASTED International Conference on Communication, Network, and Information Security*, pp. 85–90, 2003.

[HC07] Y. Hu and Z. Chen, "An SVD based watermarking method for image authentication," *International Conference on Machine Learning and Cybernetics*, Hong Kong, China, vol. 3, pp. 1723–1728, August 2007.

[Hea10] R. Healy, "Digital audio watermarking for broadcast monitoring and content identification," *Masters thesis*, National University of Ireland, Maynooth, 2010.

[Hel72] R. Hellman, "Asymmetry of Masking Between Noise and Tone," *Percep. and Psychphys.*, vol.11, pp. 241-246, 1972.

[Hem61] E. F. Hembrooke, "Identification of sound and like signals," United States Patent, 3,004,104, 1961.

[HMB11] A. A. Haj, A. Mohammad, and L. Bata, "DWT-Based Audio Watermarking," *The International Arab Journal of Information Technology*, vol. 8(3), July 2011.

[HMW88] L. Holt, B. G. Maufe, and A. Wiener, "Encoded Marking of a Recording

Signal," U.K. Patent, GB 2196167A, 1988.

[HT09] R. Healy and J. Timoney, "Digital Audio Watermarking with Semi-Blind Detection For In-Car Music Identification," *Audio Engineering Society 36th International Conference*, Michigan, USA. June 2-4, 2009.

[Ifp09] IFPI digital Music Report, 16th January 2009.

[ITU96] "Methods for subjective determination of transmission quality," *ITU-T Recommendation P.800,* Aug. 1996

[JA08] S. Jun and M.S. Alam, "Fragility and robustness of binary-phase-only-filter-based fragile/semifragile digital image watermarking," *IEEE Transactions on Instrumentation and Measurement*, vol. 57 (3), pp. 595–606, 2008.

[Jay08] Jayshankar, "Efficient computation of the DFT of a 2N - point real sequence using FFT with CORDIC based butterflies," *TENCON 2008 - 2008 IEEE Region 10 Conference* , pp.1-5, 19-21 Nov. 2008.

[JDJ99] N. F. Johnson, Z. Duric, and S. Jajodia, "Recovery of Watermarks from Distorted Images," *Third Information Hiding Workshop*, Dresden, Germany, 29 September 1999.

[JJ98] N.F. Johnson and S. Jajodia, "Steganalysis of Images Created using Current Steganography Software," *Lecture Notes in Computer Science*, Springer-Verlag, vol. 1525, 1998.

[KAAA09] N. K. Kalantari, M. A. Akhaee, S. M. Ahadi, and H. Amindavar, "Robust

Multiplicative Patchwork Method for Audio Watermarking," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17(6), pp. 1133-1141, Aug. 2009.

[Kab03] P. Kabal, "An Examination and Interpretation of ITU-R BS.1387 : perceptual Evaluation of Audio Quality," *Technical Report*, McGill University, version 2, 2003.

[KAK07] N. Kalantari, S. Ahadi, and A. Kashi, "A robust audio watermarking scheme using mean quantization in the wavelet transform domain," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 198–201, 2007.

[Kal96] D. Kalman, "A Singularly Valuable Decomposition: The SVD of a Matrix," *College Math.J.*, vol. 27(2) , 1996.

[KB11] M. R. Keyvanpour and F. M. Bayat, "Robust dynamic block-based image watermarking in DWT domain," *World Conference on Information Technology* , ISSN 1877-0509, vol. 3, pp. 238-242, 2011.

[KBWN06] W. Kong, B. Bian, D. Wu, and X. Niu, "SVD based blind video watermarking algorithm," *Proceedings of the International Conference on Innovative Computing, Information and Control*, Beijing, China, pp. 265–268, August 2006.

[KBWN06a] W. Kong, B. Bian, D.Wu, and X. Niu, "Additive vs. image dependent DWT-DCT based watermarking," *Proceedings of International*

Workshop, MRCS, Istanbul, Turkey, pp. 98–105, September 2006.

[KCSH03] H. J. Kim, Y. H. Choi, J. W. Seok, and J. W. Hong, "Audio watermarking techniques," *Pacific Rim Workshop on Digital Steganography*, Kyushu Institute of Technology, 2003.

[KH01] D. Kundur and D. Hatzinakos, "Diversity and attack characterization for improved robust watermarking," *IEEE Transactions on Signal Processing*, vol.49(10), pp. 2383-2396, Oct 2001.

[KH99] D. Kundur and D.Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proceedings of the IEEE* , vol.87 (7), pp.1167-1180, Jul 1999.

[KM02] F. Keiler and S. Marchand, "Survey on extraction of sinusoids in stationary sounds," *Proc. Of the 5th Int. Conference on Digital Audio Effects (DAFX-02)*, Hamburg, Germany, September 2002.

[KM03] D. Kirovski and H. S. Malvar, "Spread spectrum watermarking of audio signals," *IEEE Transaction on Signal Process*, vol. 51(4), pp. 1020-1033, 2003.

[KNS02] B. S. Ko, R. Nishimura, and Y. Suzuki, "Time-spread echo method for digital audio watermarking using PN sequences," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, vol. 2, pp. 2001-2004, 2002.

[KNS05] B. S. Ko, R. Nishimura, and Y. Suzuki, "Time-spread Echo Method for

Digital Audio Watermarking," *IEEE Transactions on Multimedia*, vol. 7(2), pp. 212-221, 2005.

[KP00] S. Katzenbeisser and F. A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking," Artech House, London, 2000.

[KS05] T. Kratochv ĺ and M. Ševčík, "Frequency analysis of low bit-rate modern digital audio compression algorithms," 2005.

[KSS06] C. Kim, K.D. Seo, and W.Y. Sung, "A Robust Formant Extraction Algorithm Combining Spectral Peak Picking and Root Polishing," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1-16, 2006.

[KT88] N. Komatsu and H. Tominaga, "Authentication system using concealed images in telematics," *Memoirs of the School of Science and Engineering*, Waseda University, vol. 52, pp. 45 – 60, 1988.

[Kub95] G. Kubin, "What is a chaotic signal?," *IEEE Workshop on Nonlinear Signal and Image Processing*, I. Pitas, Ed., pp. 141–144, 1995.

[Kun01] D. Kundur, "Watermarking with diversity: Insights and implications," *IEEE Multimedia*, vol. 8(4), pp. 46–52, 2001.

[LC00] Y. Lee and L. Chen, "High capacity image steganographic model," *IEEE Proceedings on Vision, Image and Signal Processing*, vol. 147(3), pp. 288 -294, June 2000.

[LC01] C.Y. Lin and S.F. Chang, “A robust image authentication method distinguish JPEG compression from malicious manipulation,” *IEEE Transactions on Circuits and Systems of Video Technology*, vol. 11 (2), pp. 153–168, 2001.

[LC06] W.N. Lie and L.C. Chang, “Robust high-quality time-domain audio watermarking based on low-frequency amplitude modification,” *IEEE Trans. Multimedia*, vol. 8 (1), pp. 46–59, 2006.

[LDSV05] A. Lang, J. Dittmann, R. Spring, and C. Vielhauer, “Audio watermark attacks: from single to profile attacks,” *Multimedia and security Workshop*, New York, USA, August 1-2 2005.

[Lin00] C.Y.Lin, “Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection,” *PhD thesis*, Columbia University, 2000.

[LL01] C.S. Lu and H.Y.M. Laio, “Multipurpose watermarking for image authentication and protection,” *IEEE Transactions on Image Processing*, vol. 10 (10), pp. 435–439, 2001

[LL11] B. Loker and R. Liu, “Approach to Real Time Encoding of Audio Samples: a DSP Realization of the MPEG Algorithm,” available on http://www.mp3-tech.org/programmer/docs/index.php [last accessed: 6 July 2011]

[LLC06] W.N. Lie, G.S.Lin, and S.L.Chen, “Dual protection of JPEG images based

on informed embedding and two-stage watermark extraction techniques," *IEEE Transactions on Information Forensics and Security*, vol. 1 (3), pp. 330–341, 2006.

[LL08] T.Y. Lee and S.D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognition*, vol. 41 (11), pp.3497–3506, 2008.

[LNK06] J. Liu, X. Niu and W. Kong, "Image watermarking based on singular value decomposition," *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Pasadena, CA, USA, pp. 457–460, December 2006.

[LS04] Y. W. Liu and J. O. Smith, "Watermarking Sinusoidal Audio Representations by Quantization Index Modulation in Multiple Frequencies," *Proceedings of ICASSP 2004*, vol. 5, pp. 373-376, 2004.

[LSL00] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: a state of the art overview," *IEEE Signal Processing Magazine*, vol. 17, pp. 20–46, 2000.

[LT02] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. on Multimedia*, vol. 4(1), pp. 121-128, Mar. 2002.

[LXL06] W. Li, X. Xue, and P. Lu, "Localized audio watermarking technique robust against time-scale modification," *IEEE Trans. Multimedia*, vol. 8 (1) , pp. 60–69, 2006.

[MAK08] H. Malik, R. Ansari, and A. Khokhar, "Robust audio watermarking using frequency-selective spread spectrum," *IET Information Security*, vol 2(4), pp. 129–150, 2008.

[Mal99] H. S. Malvar, "A modulated complex lapped transform and its application to audio processing," *Int. Conf. on Acoust., Speech, Signal Proc.*, Phoenix, AZ, pp. 1421–1424, May 1999.

[MAS08] A. A. Mohammad, A. Alhaj, and S. Shaltaf, "An improved SVD based watermarking scheme for protecting rightful ownership," *Signal Processing*, Elsevier North-Holland: Amsterdam, The Netherlands, vol. 88, pp. 2158–2180, September 2008.

[MCL08] V.Martin, M.Chabert, and B. Lacaze, "An interpolation-based watermarking scheme," *Signal Processing*, vol. 88 (3), pp. 539–557, March 2008.

[MF03] H. S. Malvar and D. A. F. Florencio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Transaction on Signal Process*, vol. 51(4), pp. 898-905, 2003.

[MJM03] D. Megas, J. H. Joancomart, and J. Minguillón, "A robust audio watermarking scheme based on MPEG 1 layer 3 compression," *Communications and Multimedia Security—CMS 2003*, Lecture Notes in Computer Science, Berlin Heidelberg, Germany, Springer-Verlag, vol. 2828, pp. 226–238, October 2003.

[MJM05] D. Megías, J. H. Joancomartí, and J. Minguillón, "Total Disclosure of the Embedding and Detection Algorithms for a Secure Digital Watermarking Scheme for Audio," *ICICS 2005*, pp. 427-440, 2005.

[MK05] A. Mooney and J. G. Keating, "Generation and Detection of Watermarks Derived from Chaotic Function," *SPIE*, *Opto-Ireland*, vol. 5615, pp. 58-69, 2005.

[MK11] S. P. Mohanty, E. Kougianos, "Real-time perceptual watermarking architectures for video broadcasting," *Journal of Systems and Software*, vol. 84(5), pp. 724-738, 2011.

[Mob98] B. Mobasseri, "Direct sequence watermarking of digital video using m-frames," *IEEE International Conference on Image Processing*, Chicago, IL, pp. 399–403, 1998.

[Moo91] B. C. J. Moore, "An Introduction to the Psychology of Hearing," *Academic Press*, 1991.

[MRF10] D. Megías, J. S. Ruiz, and M. Fallahpour, "Effcient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification", *Signal Processing*, vol. 90(12), pp. 3078-3092, 2010.

[MT03] M.F. Mansour and A.H.Tewfik, "Time-scale invariant audio data embedding," *EURASIP J. Appl. Signal Process*, vol. 1, pp. 993–1000, 2003.

[MT05] M.F. Mansour and A.H. Tewfik, "Data embedding in audio using time-scale modification," *IEEE Trans. Speech Audio Process*, vol.13 (3) , pp. 432–440, 2005.

[NC06] L. W. Nung and L.C. Chang, "Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification," *IEEE Transactions on Multimedia*, vol.8(1), pp. 46- 59, Feb. 2006.

[Nel01] D. Nelson, "Cross Spectral Methods for Processing Speech," *Journal of the Acoustic Society of America*, vol. 110 (5), pp. 2575-2592, Nov. 2001.

[NH00] C. Neubauer and J. Herre, "Advanced watemarking and Its Applications," *The 109th Convention AES*, California, USA, pp. 22-25, 2000.

[NV10] J. Nikunen and T. Virtanen, "Noise-to-mask ratio minimization by weighted non-negative matrix factorization algorithm," *IEEE Int. Conf. Acoust., Speech, Signal Process*, Dallas, TX, 2010, to be published.

[OB08] F. Omar and J. Blackledge, "Audio Data Verification and Authentication using Frequency Modulation based Watermarking," *International Society for Advanced Science and Technology, Journal of Electronics and Signal Processing* , vol. 3(2), pp. 51 - 63, 2008.

[OSHY01] H. O. Oh, J. W. Seok, J. W. Hong, and D. H. Youn, "New Echo Embedding Technique for Robust and Imperceptible Audio Watermarking," *Proc. ICASSP 2001*, pp. 1341-1344, 2001.

[OSM05] H Ozer, B. Sankur, and N. Memon, "An SVD based audio watermarking

technique," *Proceedings of the 7th ACM Workshop on Multimedia and Security*, pp. 51–56, August 2005.

[PAK98] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on Copyright Marking Systems," *Lecture Notes in Computer Science*, Springer-Verlag, vol. 1525, 1998.

[Pai00] T. Painter, et al, "Perceptual coding of digital audio," New York, NY, ETATS-UNIS, Institute of Electrical and Electronics, 2000.

[Pha08] R.C.W. Phan, "Tampering with a watermarking-based image authentication," *Pattern Recognition* , vol. 41 (11), pp. 2493–3496, 2008.

[PPB10] J.C. Patra, J.E. Phua, and C. Bornand, "A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression," *Digital Signal Processing*, vol. 20 (6), pp. 1597–1611, 2010.

[PW02] A.H. Paquet and R.K. Ward, "Wavelet-based digital watermarking for image authentication," *Proceedings of the IEEE Canadian Conference on Electrical & Computer Engineering*, pp. 879–884, 2002.

[Qua02] T. F. Quatieri, Discrete-Time Speech Signal Processing: Principles and Practice, Prentice-Hall, 2002.

[Que01] M. P. Queluz, "Authentication of digital images and video: Generic models and a new contribution," *Signal Processing: Image Communication*, vol. 16 (5), pp. 461-475, January 2001.

[QX11] X. Qi and X. Xin, "A quantization-based semi-fragile watermarking scheme for image content authentication," *Journal of Visual Communication and Image Representation*, vol. 22 (2), pp. 187–200, 2011.

[QZ04] X. Quan and H. Zhang, "Audio Watermarking based on a Psychoacoustic Model and Adaptive Wavelet Packets," *Proceedings of the 7th International Conference on Signal Processing*, vol. 3, pp. 2518-2521, 2004.

[RLV07] J. Rauhala, H. M. Lehtonen and V. Välimäki, "Fast automatic inharmonicity ation algorithm," *Journal of the Acoustical Society of America*, vol. 121(5), pp. 184-189, 2007.

[RMK07] A. Ranade, S. S.Mahabalaro and S. Kale, "A variation on SVD based image compression," *Image and Vision Computing*, vol. 25(6), pp. 771-777, June 2007.

[RR11] S. Rawat and B. Raman, "A chaotic system based fragile watermarking scheme for image tamper detection," *International Journal of Electronics and Communications*, In Press, Corrected Proof, Available online 17 February 2011.

[RS60] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *J. Soc. Ind.Appl. Math.*, Vol. 8, pp. 300-304, and Math. Rev. vol. 23B, pp. 510, 1960.

[SAH05] J. Synnevag, A. Austeng, S.Holm, "Minimum variance adaptive beamforming applied to medical ultrasound imaging," *Ultrasonics Symposium*, 2005 IEEE , vol. 2, pp. 1199- 1202, 2005.

[SAKS95] Y. Suzuki, F. Asano, H. Y. Kim, and T. Sone, "An optimum computergenerated pulse signal suitable for the measurement of very long impulse response," *J. Acoust. Soc. Amer.*, vol. 97(2), pp. 1119–1123, 1995.

[Sch96] B. Schneier, "Applied Cryptography," 2nd Edition, John Wiley & Sons Inc., New York, 1996.

[SDF95] H. C. Stankwitz, R. J. Dallaire, and J. R. Fienup, "Non-linear Apodization for Sidelobe Control in SAR Imagery," *IEEE Trans. On Aerospace and Elect. Syst.*, vol. 31(1), pp. 267-279, Jan. 1995.

[SGD11] J. Singh, P. Garg, and A. De, "Audio Watermarking based on Quantization Index Modulation using Combined Perceptual Masking," *Multimedia tools and Applications*, Springer, 2011.

[Sho00] Short et al, "Method and apparatus for compressed chaotic music synthesis," United States Patent 6,137,045, October 2000.

[Sin11] V. Singh, "Digital Watermarking: A Tutorial, Journal of Selected Areas in Telecommunications," January Edition, 2011.

[SLDP02] M. Steinebach, A. Lang, J. Dittmann, and F. A. P. Petitcolas, "Stirmark benchmark: audio watermarking attacks based on lossy compression,"

SPIE Security and Watermarking of Multimedia, San Jose, CA, USA, vol. 4675, pp. 79–90, Jan. 2002.

[SPR+01] M. Steinebach, F.A. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, S. Seibel, and N. Fates, "StirMark benchmark: audio watermarking attacks," *International Conference On Information Technology: Coding And Computing*, 2001.

[SS10] P. Singla and T. Singh, "Desired Order Continuous Polynomial Time Window Functions for Harmonic Analysis," *IEEE Transactions on Instrumentation And Measurement*, vol. 59, pp.2475-2481, 2010.

[SS11] M. M. Sathik and S.S. Sujatha, "Application of Toeplitz matrix in watermarking for image authentication," *2011 International Conference on Computer, Communication and Electrical Technology*, pp.55-59, March 2011.

[SSY02] R. Sun, H. Sun, and T. Yao, "A SVD and quantization based semi-fragile watermarking technique for image authentication," *Proceedings of the 6th International Conference on Signal Processing*, pp. 1592–1595, August 2002.

[SY06] S. R. Subramanya and B. K. Yi, "Digital Rights Management," *IEEE Potentials*, vol. 25(2), pp.31-34, 2006.

[Sze79] W. Szepanski, "A signal theoretic method for creating forgery proof documents for automatic verification," *Carnahan Conference on Crime*

Countermeasures, pp. 101–109, 1979.

[SW00] J. Simpson and E. Weiner, "Oxford English Dictionary," Oxford University Press, New York, 2000.

[Swe96] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3 (2), pp. 186–200, 1996.

[SZTB98] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Process*, vol. 66(3), pp. 337-255, 1998.

[TB97] L. Trefethen and D. Bau, "Numerical linear algebra," *SIAM : Society for Industrial and Applied Mathematics*, PA, USA, 1997.

[TDE04] B. Thom, K. Douglas and W. Erling, "Audio Fingerprints: Technology and Applications," *The 117th Audio Engineering Society Convention*, Oct. 2004.

[TEC01] G. Tzanetakis, G. Essl, and P. Cook , "Audio Analysis using the Discrete Wavelet Transform," *Proc. WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001)* , Skiathos, Greece, 2001.

[TFJ00] G. Thomas and B. C. Flores, "S. S. Jae, SAR sidelobe apodization using the Kaiser Window," *Image Processing*, 2000.

[TS01] R. Tachibana, S. Shimizu, T. Nakamura, and S. Kobayashi, "An audio watermarking method robust against time and frequency fluctuation," in

Proc. SPIE Security and Watermarking of Multimedia Contents III, 2001, vol. 4314, pp. 104–115.

[TSKN02] R. Tachibana, S. Shimizu, S. Kobayashi, and T. Nakamura, "An audio watermarking method using a two-dimensional pseudo-random array," *Signal Process*, vol. 82 (10) ,1455–1469,2002.

[VKK09] H. Verma, R. Kaur, and R. Kumar, "Random Sample Audio Watermarking Algorithm for Compressed Wave Files," *International Journal of Computer Science and Network Security*, vol.9 (11), November 2009.

[VTCL05] R. Vieru, R. Tahboub, C. Constantinescu, and V. Lazarescu, "New Results using Audio Watermarking Based on the Wavelet Transform," *International Symposium on Signals, Circuits and Systems*, vol. 2, pp. 441- 444, 2005.

[Wal01] C. Walker, "Personal Communication", Muzak Corporation, June, 2001.

[WCC04] C.T. Wang, T. S. Chen, and W.H. Chao, "A new audio watermarking based on modified discrete cosine transform of MPEG/audio layer III," *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, Taiwan, March 21–23, pp. 984–989, 2004.

[Wei79] C.J. Weinstein, "Programs for Digital Signal Processing," *IEEE Press*, 1979.

[WHHS05] S. Wu, J. Huang, D. Huang, and Y. Q. Shi, "Efficiently Self-Synchronized

Audio Watermarking for Assured Audio Data Transmission," *IEEE Trans. on Broadcasting*, vol. 51(1), pp. 69–76, March 2005.

[WL03] M. Wu and B. Liu, "Multimedia Data Hiding," *Springer Verlag*, New York, NY, 2 003.

[WM01] P.W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10 (10) , pp. 1593–1601, 2001.

[WNY09] X. Y. Wang, P. P. Niu, and H. Y. Yang, "A robust digital audio watermarking based on statistics characteristics," *Pattern Recognition*, vol. 42 (11) , pp. 3057-3064, ISSN 0031-3203, November 2009.

[WPD99] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and videos," *Proc. IEEE*, vol. 87(7), pp. 1108-1126, 1999.

[WSK99] C. P. Wu, P. C. Su, and C. C. Kuo, "Robust audio watermarking for copyright protection," *Proc. SPIE'S 44th Ann.*, vol. 3807, pp. 387-397, July 1999.

[WYWQ10] D. Wu, S. G. Yang, Z.C. Wang, W.H. Qin, "BCH Code-Based Robust Audio Watermarking Algorithm in the DWT Domain," *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on* , vol.1, pp. 878-880, 11-12 May 2010

[WZ06] X. Y. Wang and H. Zhao, "A Novel Synchronization Invariant Audio

Watermarking Scheme Based on DWT and DCT," *IEEE Transactions on Signal Processing*, vol. 54(12), pp. 4835-4840, 2006.

[XKH08] S. J. Xiang, H. J. Kim, and J. W. Huang, "Audio watermarking robust against time-scale modification and MP3 compression," *Signal Processing*, vol.88(10), pp.2372-2387, October 2008.

[XM06] Y. Xiong and Z. X. Ming, "Covert Communication Audio Watermarking Algorithm Based on LSB," *International Conference on Communication Technology*, ICCT 06, pp. 1-4, 2006.

[XWSX99] C. Xu, J.Wu, Q. Sun, and K. Xin, "Applications of digital watermarking technology in audio signals," *J. Audio Eng. Soc.*, vol. 47(10), Oct. 1999.

[YK03] I. K. Yeoand and H. J. Kim, "Modified patch work algorithm: A novel audio watermarking scheme," *IEEE Trans. Speech Audio Process*, vol. 11 (4), pp. 381–386, Jul. 2003.

[YK03+] I. K. Yeo and H. J. Kim, "Generalized patchwork algorithm for image watermarking," *Multimedia Syst.*, vol. 9(3), pp. 261–265, 2003.

[YK99] C. Yeh and C. Kuo, "Digital Watermarking through Quasi mArrays," *Proc. IEEE Workshop on Signal Processing Systems*, Taipei, Taiwan, pp. 456-461, October 1999.

[YKL01] H. Yu, D. Kundur, and C. Lin, "Spies, thieves, and lies: The battle for multimedia in the digital era," *IEEE Multimedia*, vol. 8(3), pp. 8–12, 2001.

[YSZ11] L. B. Ying, I. Y. Soon, and L. Zhen, “Blind and robust audio watermarking scheme based on SVD-DCT,” *Signal Processing*, vol. 91 (8), pp. 1973-1984, August 2011.

[ZF90] E. Zwicker and H. Fastl, “Psychoacoustics, Facts and Models,” *Springer-Verlag* ,1990.

[ZL05] X. Zhang and K. Li, “Comments on an SVD-based watermarking scheme for protecting rightful ownership,” *IEEE Trans. Multimedia*, vol. 7 (3), pp. 593–594, April 2005.

[ZZ91] E. Zwicker and U. Zwicker, “Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System,” *J. Audio Eng. Soc.*, pp. 115-126, Mar. 1991.