

國立中央大學

電機工程研究所

碩士論文

金匙小波域音訊浮水印系統

指導教授：張寶基教授

研究生：黃景明

中華民國九十一年六月



國立中央大學圖書館 碩博士論文授權書

(91 年 5 月最新修正版)

本授權書所授權之論文全文與電子檔，為本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

(☒)同意 (立即開放)

()同意 (一年後開放)，原因是：_____

()同意 (二年後開放)，原因是：_____

()不同意，原因是：_____

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動讀者間「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、光碟、網路或其它各種方法收錄、重製、與發行，或再授權他人以各種方法重製與利用。以提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名：黃景明

論文名稱：金匙小波域音訊浮水印系統

指導教授姓名：張寶基 教授

系所：電機工程研究所 ☐博士 ☒碩士班

學號：89521044

日期：民國91年6月14日

備註：

1. 本授權書請填寫列印兩份紙本，並親筆簽名後（全文電子檔內之授權書可用電腦打字代替），一份請裝訂於紙本論文封面後之次頁，一份於辦理離校時，交圖書館服務台（以統一轉交國家圖書館），未附本授權書，圖書館將不予驗收。
2. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

金匙小波域音訊浮水印系統

摘要

本論文主要針對應用於音訊上的浮水印，提出一套不僅具有良好音訊品質且經音訊攻擊後仍具有高強韌性，且在萃取端並不需要原始音訊的數位浮水印技術。

本文提出的方法，以小波封包 (Wavelet Packet) 分頻方式，將樂音訊號經由濾波器群組分成 29 個次頻帶，其頻寬分布與人類聽覺的 26 個關鍵頻帶 (Critical Band) 相近，藉以找出人耳聲學模型 (Psychoacoustic Model) 中的最小遮蔽臨界值 (Minimum Masking Threshold)，此值將作為嵌入浮水印時更動小波係數的依據。浮水印嵌入時選擇人耳較不敏感的偏低頻係數作為嵌入之用，並且使用區塊分類來過濾不適合嵌入浮水印的係數區塊，將數位浮水印嵌入較不易改變極性之區塊以確保浮水印的強韌性。另一個特點為在浮水印嵌入過程中會產生一組與原始音訊相關的安全鑰匙 (Secure Key)，萃取端藉由此安全鑰匙即可完整取出嵌入的浮水印資訊，所以並無傳統做法中對原始音訊的需求，可以省去原始音訊的儲存空間。此外為了克服音訊同步的問題，也加入了同步碼搜尋的機制。實驗結果顯示對於 MP3 48 Kbps 以上之壓縮，以及 StirMark 各式的攻擊均有足夠的強健性，對於音訊裁切攻擊也能正確的找回同步點位置。

Key-based Audio Watermarking System using Wavelet Packet Decomposition

Abstract

In this thesis, we propose an audio watermarking system based on wavelet packet decomposition and psycho-acoustic modeling. The audio watermarking system can deliver perceptual transparent audio quality, and it is robust against various signal processing or malicious attacks.

The original audio signal is first segmented and divided into 29 subbands via wavelet packet decomposition. The bandwidth allocation of the subband decomposition structure is close to the critical band structure of human auditory system. Middle and middle-low subbands are chosen for watermark embedding. A selective embedding method is used to embed watermark into those coefficient blocks with clear block polarities. Modification of selected coefficients is based on the minimum masking threshold of psycho-acoustic model. Instead of the original Audio signal, a secure key is used in the watermark extraction to indicate the locations where watermark bits are embedded. In addition, a synchronization mechanism is imposed to search for the frame boundary of attacked audio signal.

Simulation results show that the watermarking system is robust against MP3 compression at 48 Kbps and above. It also survives StirMark attacks for audio and cropping attacks.

目錄

摘要.....	I
Abstract.....	II
目錄.....	III
圖目.....	V
表目.....	IX
第一章 緒論.....	1
1.1 簡介.....	1
1.2 研究動機與目的.....	2
1.3 系統架構.....	4
1.4 論文架構.....	5
第二章 音訊浮水印的發展簡介.....	7
2.1 數位浮水印的應用.....	7
2.2 數位浮水印的功能需求.....	10
2.3 音訊浮水印標準評定程序.....	12
2.4 數位音訊浮水印的發展.....	15
第三章 小波分析與人耳聲學模型.....	21
3.1 小波簡介.....	21
3.1.1 小波轉換.....	21
3.1.2 小波濾波器.....	24
3.1.3 小波封包.....	27
3.2 小波分頻架構.....	29
3.3 人耳聲學模型.....	33
3.3.1 基本原理與其應用.....	33

3.3.2 模型公式.....	37
3.3.3 訊號之各頻帶的最小遮蔽臨界值	41
3.4 轉換頻率遮蔽曲線至小波域	44
第四章 小波域數位音訊浮水印	46
4.1 數位浮水印嵌入架構	48
4.1.1 小波次頻帶選擇.....	49
4.1.2 子區塊分類.....	51
4.1.3 私密鑰匙的產生.....	53
4.1.4 子區塊極性強化.....	55
4.2 數位浮水印萃取架構	57
4.3 同步碼嵌入架構.....	59
4.4 同步碼搜尋萃取架構	64
第五章 實驗結果及討論	66
5.1 浮水印容量與音質評估	66
5.1.1 浮水印容量評估.....	66
5.1.2 樂音音質評估.....	68
5.2 音訊壓縮測試.....	71
5.3 StirMark for Audio 測試	76
5.4 裁切測試.....	84
第六章 結論.....	92
參考文獻.....	93
附錄 A.....	96

圖目

圖 1.1 數位音訊浮水印系統嵌入架構圖	5
圖 1.2 數位音訊浮水印系統萃取架構圖	5
圖 2.1 Tewfik method 嵌入浮水印流程圖	15
圖 2.2 PA model masking threshold 輸出	16
圖 2.3 PN sequences 產生圖	16
圖 2.4 頻譜相位修改示意圖	17
圖 2.5 host signal randomize	18
圖 2.6 Wang method 嵌入浮水印流程圖	18
圖 2.7 Impulse response and frequency response of single echo	19
圖 2.8 Frequency response of multiple echo	19
圖 3.1 小波的頻譜與時間分割配置圖	21
圖 3.2 傅立葉轉換的頻譜與時間分割配置圖	21
圖 3.3 小波分解多解析度空間圖	24
圖 3.4 一階小階小波分頻與合成濾波器	25
圖 3.5 Daubechies biorthogonal 6.8 小波濾波器係數	26
圖 3.6 Daubechies Biorthogonal 6.8 小波濾波器之振幅響應	26
圖 3.7 Daubechies Biorthogonal 6.8 小波濾波器的群組延遲	27
圖 3.8 兩個濾波器組的三階小波轉換系統與其頻帶解析	28
圖 3.9 三階小波封包系統的完整二元樹結構與其頻帶解析	28
圖 3.10 小波封包之各次頻帶頻寬分布	29
圖 3.11 小波封包次頻帶與 MPEG 人耳聲學模型關鍵頻帶的關係	30
圖 3.12 Flute 訊號第一、二個音框之小波係數	31
圖 3.13 Organ 訊號第一、二個音框之小波係數	31

圖 3.14 Harpsichord 訊號第一、二個音框之小波係數	32
圖 3.15 Violin 訊號第一、二個音框之小波係數	32
圖 3.16 窄頻雜訊的遮蔽效應	35
圖 3.17 時間軸上的遮蔽效應	36
圖 3.18 MPEG Psychoacoustic Model II 之絕對聽力臨界值 ATH	37
圖 3.19 Pre-Echo 例子	40
圖 3.20 最小遮蔽臨界值取出示意圖	41
圖 3.21 Organ 訊號第 2、42、82 個音框的遮蔽曲線	42
圖 3.22 Organ 訊號第 2、42、82 個音框的最小遮蔽臨界值	42
圖 3.23 Flute 訊號第 2、42、82 個音框的遮蔽曲線	43
圖 3.24 Flute 訊號第 2、42、82 個音框的最小遮蔽臨界值	43
圖 3.25 Organ 訊號第 2、42、82 個音框的最小遮蔽臨界值	45
圖 3.26 Flute 訊號第 2、42、82 個音框的最小遮蔽臨界值	45
圖 4.1 數位音訊浮水印系統嵌入架構圖	47
圖 4.2 數位音訊浮水印系統萃取架構圖	48
圖 4.3 數位音訊浮水印嵌入架構流程圖	48
圖 4.4 嵌入數位浮水印的小波次頻帶	50
圖 4.5 嵌入數位浮水印的頻帶範圍	50
圖 4.6 次頻帶係數子區塊分類	51
圖 4.7 TXOR 運算產生安全鑰匙的例子	54
圖 4.8 子區塊係數修改示意圖	56
圖 4.9 數位音訊浮水印萃取架構流程圖	57
圖 4.10 由數位音訊萃取數位二階浮水印的例子	58
圖 4.11 一組【同步碼 數位浮水印】資料	60
圖 4.12 同步碼與數位浮水印嵌入樂音訊號範例	60

圖 4.13 同步碼嵌入架構流程圖	60
圖 4.14 嵌入 $S_l = 1$ 時更動 E_s 示意圖	63
圖 4.15 強化視窗	64
圖 4.16 抑制視窗	64
圖 4.17 同步碼搜尋萃取架構流程圖	64
圖 5.1 Level 1 及 Level 2 浮水印容量	67
圖 5.2 數位二階浮水印	68
圖 5.3 Level 1 及 Level 2 樂音音質	69
圖 5.4 方法 Level 1 中壓縮位元率與 NC 值的關係	72
圖 5.5 方法 Level 2 中壓縮位元率與 NC 值的關係	73
圖 5.6 方法 Level 1 中，在不同壓縮位元率下所萃取出之浮水印	74
圖 5.7 方法 Level 2 中，在不同壓縮位元率下所萃取出之浮水印	74
圖 5.8 方法 Level 1 中不同音訊攻擊下所萃取出之浮水印 NC 值 (1)	79
圖 5.9 方法 Level 1 中不同音訊攻擊下所萃取出之浮水印 NC 值 (2)	80
圖 5.10 方法 Level 2 中不同音訊攻擊下所萃取出之浮水印 NC 值 (1)	81
圖 5.11 方法 Level 2 中不同音訊攻擊下所萃取出之浮水印 NC 值 (2)	82
圖 5.12 Invert 攻擊後萃取出之數位二階浮水印	83
圖 5.13 Level 1 中取樣點位移後萃取出之數位二階浮水印 NC 值	85
圖 5.14 Level 2 中取樣點位移後萃取出之數位二階浮水印 NC 值	86
圖 5.15 Guitar 樂音經 D/A、A/D 後強度增強	88
圖 5.16 Guitar 樂音測試結果	88
圖 5.17 Violin 樂音經 D/A、A/D 後強度減弱	89

圖 5.18 Violin 樂音測試結果	89
圖 5.19 Mozart : Eine Kleine Nachtmusik, K525-2nd Movement 裁切後 樂音測試	90

表目

表 3.1 關鍵頻帶的中央與邊界頻率	34
表 4.1 TXOR 運算真值表	54
表 5.1 平均每個音框可嵌入的位元數	67
表 5.2 統計 12 種樂器可嵌入位元數的統計值	67
表 5.3 嵌入浮水印之樂音品質	69
表 5.4 統計 12 種樂器音質的統計值	69
表 5.5 Level 1 聽覺喜好比例值	70
表 5.6 Level 2 聽覺喜好比例值	70
表 5.7 不同壓縮位元率下 Level 1 方法的 NC 值	72
表 5.8 不同壓縮位元率下 Level 2 方法的 NC 值	73
表 5.9 Stirmark for Audio 對 Level 1 方法的 NC 值	77
表 5.10 Stirmark for Audio 對 Level 2 方法的 NC 值	78
表 5.11 Resampling 攻擊後萃取出浮水印 NC 值	84

第一章 緒論

網際網路的快速發展與使用多樣性，使得資訊在網際網路上傳遞變得非常的迅速與便利，使用者可以輕易的在網際網路上搜尋取得所想要的資訊，而且隨著多媒體資訊的數位化、音視訊壓縮格式的發展以及網際網路頻寬的增加，多媒體資訊在網際網路上的傳播成為一種流行的趨勢。由於多媒體資訊的取得十分便利，且數位資料容易複製、修改、傳播的特性，使得多媒體資料被任意拷貝、非法使用、非法散佈變成一項非常嚴重的問題。因此對於數位資料的著作權的宣告及保護已是一項重要的研究方向。

1.1 簡介

隨著電腦的普及和網際網路的發展，人們可以快速的在網際網路上搜尋取得各種多媒體資訊，許多的公司行號及個人也常在網頁上放置屬於公司或個人創作的圖像、影片、音樂等多媒體資訊供大家欣賞，或者是經由使用者付費而可以享受到隨選視訊、線上電影院及線上音訊廣播等服務。然而這些多媒體資料通常是可以下載而且也是非常容易複製、修改及傳遞的，為了避免這些創作被一些惡意的使用者任意拷貝、傳播，或宣稱是自己的創作而非法圖利，如何對這些多媒體資訊的著作權做宣告或保護就變得非常重要。而數位浮水印 (Digital Watermarking) 就是為了解決智慧財產權等相關問題而發展出來的一種技術[1][2]。

數位浮水印是類似於資料隱藏 (Data Hiding) 的一種技術，可以將一些關於著作權的訊息，如特殊的商標、序號，或所有權者的相關

資訊，藏入圖像、視訊、音訊等欲保護的數位資料中，以利創作者由數位資料中取出藏在其中的資訊來證明其所有權。而且為了達到保護著作權的目的，數位浮水印必須滿足許多要求，例如它必須要能夠抵抗一般的數位訊號處理以及不能影響到原始媒體的品質等等要求。

一般人可能會認為使用加解密技術的資訊安全系統也同樣可以達到保護多媒體資料的目的。密碼系統是將原始的資料經過加密的技術變成無法辨識的密文，若沒有經過解密的程序，該資料則無法使用。而資料一經解密程序解密後，將與原始資料無異，保護措施已經消失，則該資料將毫無保障。而且加密資料如果被他人蓄意的破解，解密後的資料並無法驗證著作權，這個攻擊的行為在網路發達與電腦運算速度極快速的今天，並沒有所謂絕對安全的加密技術。若是資料是以公開流通性為主，例如網頁上的資料，很明顯的便不適用於加密機制。而數位浮水印則是將一些智慧財產權的訊息，在不嚴重傷害多媒體品質的前提下，小幅度的修改原始資料的某些性質，嵌入隱藏在多媒體資料中，且浮水印並不會改變資料格式，也不會限制資料存取的權限，使用者並不易發覺該資料是否藏有浮水印。若是使用者發覺有浮水印的存在而想攻擊破壞浮水印，但是浮水印具有強健性 (Robustness)，並不容易從數位資料中移除，若是執意要去除浮水印，則會大大的破壞了數位資料的品質，而失去了鑑賞或傳播的價值。而一但數位資料的所有權者發現有惡意使用者非法拷貝和傳播時，可將數位浮水印取出以做為著作所有權的證明。

1.2 研究動機與目的

早期數位浮水印的發展主要針對在靜態影像上的應用，由於靜態

影像資料量較小，且經過影像壓縮（如 JPEG）後資料量更少，經常使用在網頁上，使用者可以方便快速的瀏覽，但由於網頁上的資料可以被完整的傳送與下載，因此對於應用在靜態影像上的數位浮水印技術有其迫切性的需求。而近年來由於寬頻網路建設不斷擴展，如 ISDN、xDSL、纜線數據網路等，加上個人電腦處理器效能的大幅增進與音視訊多媒體壓縮技術的演進，使用者對於音視訊多媒體的需求與日俱增，在寬頻網路上傳播音訊與視訊已是今日常見的一種趨勢。因此應用在音訊及視訊上的數位浮水印技術已是一門值得研究的課題。

應用在靜態影像及視訊上的數位浮水印的研究已經有許多的方法被提出，這些方法最主要也都是集中在研究數位浮水印的不可視性（Invisibility）以及如何抵抗各種各樣的訊號處理或攻擊。這些研究主要可以分類為兩大類：空間域（Spatial Domain）中的浮水印，以及頻率域（Frequency Domain）中的浮水印。而這些研究方法主要趨向於分析人類視覺系統（Human Visual System, HVS），得到人眼可接受的動態誤差範圍等相關資訊，藉由分析這些資訊得到更具強健性的浮水印技術。

相較於靜態影像或視訊上的浮水印，音訊浮水印具有更高的困難度及挑戰性[3]，因為人類聽覺系統（Human Auditory System, HAS）遠比人類視覺系統來的更為敏感。人類聽覺系統可以感知到聲音強度變化的動態範圍非常大，並且對於外加的雜訊非常的敏感。雖然如此，聽覺系統還是有其較不敏感的地方，例如對於聲音的絕對振幅大小以及相位的變化有較低的敏感度，而且強度較強的聲音常會遮蔽強度較弱的聲音，這些特性經常使用在音訊壓縮技術之中，而這也是音訊浮水印系統所必須要深入研究的地方。

由於人耳對於每個頻率聲音的靈敏程度是不一樣的，所謂關鍵頻帶（Critical Band）便是定義在相同的關鍵頻帶內，人耳對於聲音的靈敏度大約是相同的，人耳的每一個關鍵頻帶寬度並不相同，通常是越高頻寬度越寬。要如何利用關鍵頻帶所提供的資訊應用在音訊浮水印系統中是一個重要的關鍵。此外近年來小波轉換成功的運用於音訊的編碼[4][5][6]，因為小波的多解析度，可以有效配合關鍵頻帶非等寬頻帶的特性。所以本篇論文應用小波封包（Wavelet Packet）分頻配合人耳聲學模型，發展一套在小波域中的數位音訊浮水印技術。

1.3 系統架構

本音訊浮水印系統，浮水印嵌入方塊圖如圖 1.1，浮水印萃取方塊圖如圖 1.2。浮水印嵌入方式是以小波封包為基礎，先將樂音訊號分割成 1024 個取樣點的音框，再對音框做小波封包處理，並同時利用人耳聲學模型對樂音做分析，在嵌入浮水印之前，先嵌入特定的同步碼，之後再依據欲嵌入之數位浮水印及小波係數的特性，做數學邏輯運算產生一組二元安全鑰匙，以取代傳統數位浮水印技術對原始音訊的需求，最後將修改過的小波係數還原成時域訊號即為藏有數位浮水印的樂音訊號。而浮水印萃取方式則是先搜尋到正確的同步碼後，再利用安全鑰匙從樂音訊號中萃取出藏在其中的數位浮水印。

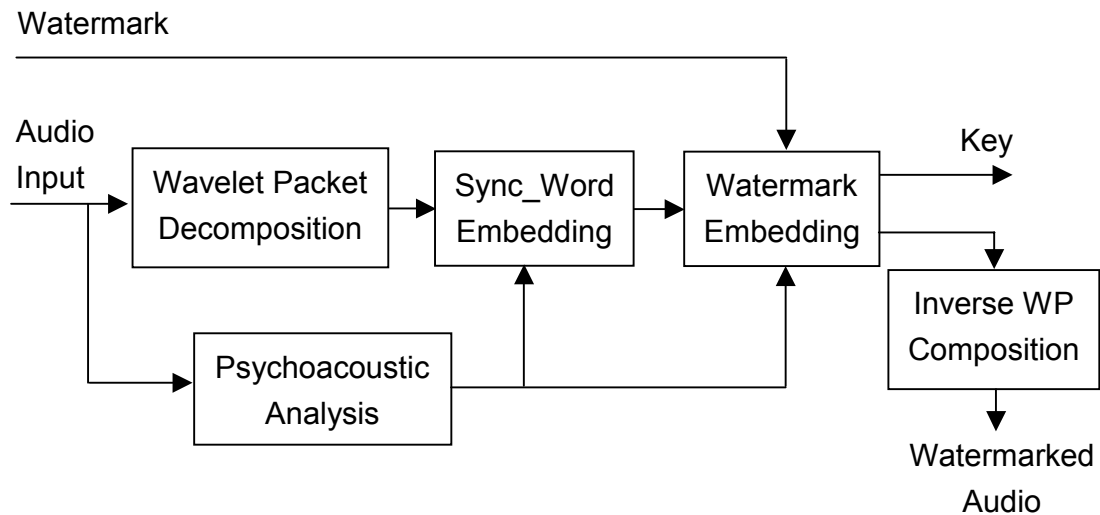


圖 1.1 數位音訊浮水印系統嵌入架構圖

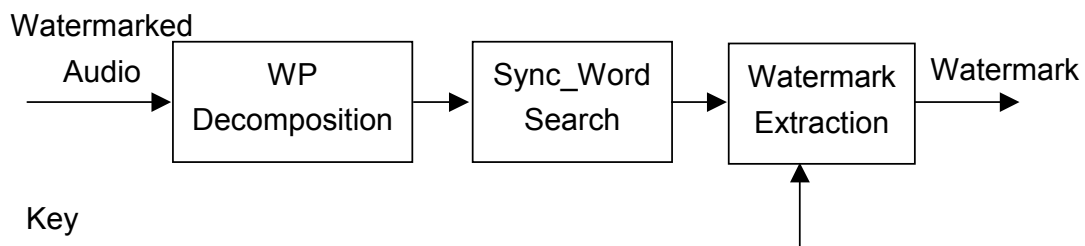


圖 1.2 數位音訊浮水印系統萃取架構圖

1.4 論文架構

本論文的第一章，為簡單的動機概述。第二、三章為基本原理及數位浮水印的介紹。第二章將會先簡單介紹數位浮水印的功能需求及其應用，以及如何客觀的評估數位浮水印的優劣和介紹一套大家熟知的數位音訊浮水印的標準評定程式（StirMark for Audio），並簡介分類近年來音訊浮水印技術的發展。第三章則對小波轉換、小波封包分頻及人耳聲學模型做一概略性的簡介，並提出系統分頻架構。第四章說明本論文所提出的在小波域中的數位音訊浮水印技術，以及同步碼

之嵌入及搜尋萃取的方式。第五章為實驗結果及品質評估討論，檢視音訊浮水印技術對樂音音質的影響，以及針對音訊壓縮、StirMark for Audio、音訊裁切等攻擊做測試，評定音訊浮水印的強健性。第六章則為結論。

第二章 音訊浮水印的發展簡介

近年來所提出的浮水印技術，是一種保護著作權與提倡版權的方式。所謂數位浮水印就是要驗證數位資料的擁有者或是傳播者，藉由加入浮水印，可以宣告或證明著作權。加入浮水印的資料，因為與原始資料幾乎相同，因此在使用上與原始資料並無差異。

在本章中將介紹音訊浮水印的發展。在 2.1 節中將會介紹數位浮水印可應用的範圍；2.2 節討論數位浮水印必須具備哪些特性；2.3 節介紹數位音訊浮水印的標準評定程序 (Benchmark)；2.4 節列舉並概略性的分類一些近年來在音訊發展數位浮水印的技術。

2.1 數位浮水印的應用

不同的數位浮水印技術可適用於不同的應用範圍，而不同的應用範圍其對於數位浮水印的功能需求也不相同，本節將介紹幾種常見的數位浮水印的應用類型[7]。

廣播監控 (Broadcast Monitoring)

在電視或電台廣播中，廣告常是一個電台的重要收入來源，電台藉著將廣告時段提供給廣告商播放廣告來收取費用，而廣告商當然也希望電台確實的將廣告依時段播出。而音樂家或演員也會要求電台在播放他們的作品時能夠付給他們應得的版稅，版權所有者也希望能確實的監控是否有未授權的電台播放他們的作品。

在廣播監控的應用方面，可以將一個獨特的數位浮水印嵌入影片或音樂片段中，並設立一個自動監控的接收站，接收站可以接收監控

電台播放的影片或聲音等媒體，並自動的在媒體中搜尋這個獨特的數位浮水印，如此便可以確切的知道這些媒體被播放的時間、次數等相關的訊息。

版權宣告 (Owner Identification)

傳統的文字版權宣告形式如“©年代日期, 所有者”，我們常可以在書本的封面、CD 的包裝上或是電影的片尾中看到此類的宣告，而這種文字類型版權宣告方式的缺點就是可以很容易的從保護媒體中移除。而數位浮水印是將宣告資訊隱藏在欲保護的媒體中，並無法輕易的移除，若與文字型態的版權宣告配合使用，可以補強版權宣告形式的完整性。

所有權證明 (Proof of Ownership)

數位浮水印除了可以用來宣告版權之外，也必須要有可以證明所有權的功能。理論上著作者可以利用嵌入於媒體中的數位浮水印來證明其所有權，然而這不是一個可以輕易解決的問題[8]。

認證 (Authentication)

認證也是數位浮水印的其中一種重要的應用，在資訊數位化的今天，任何一種多媒體資料的內容皆有可能被竄改，以網路上流傳的照片為例，它的內容有可能被輕易的竄改而成為合成照，但是有時後我們卻無法發覺其真偽，這種問題尤其在醫學影像上或法律訴訟相關問題上更顯得特別重要。

解決的方法可以直接在靜態影像中嵌入一組特別的識別標誌，由於此識別標誌已經變成了影像的一部份，若影像遭到竄改則識別標誌也會受到某種程度的影響，我們可以用此識別標誌來判斷收到的影像

是否為原始正確的影像資料。甚至某些文獻上的方法還可以大略的指出所收到影像被修改的大概位置。

指紋鑑別 (Transactional Watermarks (Fingerprinting))

在上述廣播監控、版權宣告、認證等應用時，我們對於所有多媒體資料的複本拷貝，都是嵌入同一組數位浮水印。然而有時候我們卻需要對每一個複本嵌入不同的浮水印，這種應用我們稱為指紋鑑別，它可以讓多媒體版權擁有者或多媒體傳播者辨識該多媒體資料是不是一個合法的複本。

以另外一種例子來說明，假設某音樂 CD 在正式發售上市前，只有少數幾個複本交給相關的製作參與者，但是在發售前卻發現此音樂 CD 已經在盜版市場流通。由於每一個合法複本皆有一個獨特的浮水印，一但有此情形發生，將可以從盜版的 CD 中萃取出數位浮水印，以此來追查出此非法盜拷的源頭是從哪一個複本流散出去的。

複製管制 (Copy Control)

前面所述的幾種應用，雖然可以對非法使用或非法拷貝造成嚇阻的效果，但是還是無法阻止非法複製的發生，然而若是可以和硬體設備配合，如播放或錄製的機器，還是可以達到複製管制的要求。例如若是錄製的機器偵測到標示禁止錄製的數位浮水印時，便會取消錄製的動作。然而這種應用還是須要所有硬體製造廠商的配合。

祕密通訊 (Covert Communication)

祕密通訊是早期數位浮水印發展的一種重要的應用之一，可以將它視為資訊隱藏的一種應用，主要的目的是要將欲傳送的祕密訊息隱藏在實際傳送的掩護媒體資料中，就算有第三者擷取到此媒體資料，

也只是將其視為一般性質的資料，而無法探知真正欲傳送的祕密訊息。

除此之外，將資訊隱藏配合多媒體壓縮編碼技術，在傳送位元流的同時，也會傳遞隱藏在其中的祕密訊息，可將此祕密訊息視為位元流的附加資訊 (Side Information)，並利用此附加資訊作為錯誤偵測、封包遺失回復[33]等相關技術的應用。

2.2 數位浮水印的功能需求

如同 2.1 節的敘述，針對不同的應用，數位浮水印所需要具備的功能也不盡相同，以一般性的觀點來看數位浮水印技術，它提供了一個將媒體資料與隱藏資訊連結在一起的技術，而這個技術衍生出了許多的應用。一般而言數位浮水印的應用可以約略的分為兩大類[9]：

隱密浮水印 (Secret Watermarks)

此類浮水印主要的目的是用來做驗證、判別或證明之用，如 2.1 節提到的廣播監控、版權宣告等用途，可想而知此類數位浮水印就必須要具有高度的安全性，只有版權所有者才有辦法存取數位浮水印的資料。

公用浮水印 (Public Watermarks)

此類浮水印是用來給多數人使用的一種技術，如 2.1 節提到的祕密通訊等應用，他的特性就是所有相關的人皆可以存取數位浮水印的資料，而它的安全性要求就是不能讓不相關的第三者可以輕易的偵測到或移除數位浮水印。

雖然面對不同的應用，數位浮水印會有不同的功能需求，但是以一般而言，數位浮水印技術還是需要滿足某些基本、訊號處理及安全性上的功能需求：

基本特性 (General Properties)

- 數位浮水印必須要能夠即時處理
- 必須要具有可以調整不同程度的強健性、樂音音質、浮水印容量的機制
- 萃取數位浮水印不需要用到原始的多媒體資料
- 數位浮水印技術可以應用在不同的多媒體資料上
- 要能夠嵌入多個數位浮水印 (Multiple Embedding)

訊號處理特性 (Signal Processing Properties)

- 數位浮水印不能影響媒體的感官品質
- 數位浮水印必須要能夠抵抗一般的數位訊號處理，如壓縮、濾波、量化、裁切等處理

安全性特性 (Security Properties)

- 數位浮水印技術必須依靠安全鑰匙 (Secure Key) 來確保安全性，而不是依靠在對數位浮水印技術方法的保密
- 數位浮水印技術方法必須要公開
- 數位浮水印必須要是統計上無法偵測的
- 對於使用多份複本來攻擊的方式 (Collusion Attacks)，數位浮水印要具備足夠的強健性

近年來在音訊發展數位浮水印的文獻，大多數都著重於強健性以及對於音質感知透明性（Perceptual Transparency）的探討，然而在音訊浮水印中，要滿足什麼樣的條件才算是具有強健性，一般而言，樂音訊號經常依據不同的需求而經過某些種類的數位訊號處理，或者是惡意的第三者欲藉由對樂音訊號的攻擊而破壞隱藏在其中數位浮水印，所以數位浮水印必須要能夠抵抗下列這些形式的攻擊[10]：

- 加入雜訊
- 線性或非線性濾波，如低通濾波器
- 音訊壓縮，如 MP3、AAC、AC-3 等
- 對取樣點重新量化，如 16-bit \rightarrow 8-bit
- 時間刻度的增減，如時間拉長 10 %
- 等化器的處理
- 移除或插入取樣點，或是取樣點次序交換
- D/A 跟 A/D 的轉換
- 混音，如雙聲道混合成單聲道
- 對多個嵌入浮水印的複本做平均處理

2.3 音訊浮水印標準評定程序

一個好的數位浮水印技術，必須要能夠提供足夠的且值得信賴的數據來證明其優異性，但是一般文獻上所列出的數據結果大多是作者自行做的驗證，當然不同的文獻所使用的驗證方式不同，其數據結果的表達方式也各有差異，我們很難從這些數據結果去比較各種數位浮水印技術的優劣。

傳統上我們要信賴浮水印技術所宣稱的可靠性及實用性，必須先

對它的品質、性能做一個評定，而評定的方式可分為三個等級：

1. 信任技術提供者對其技術方法所做的品質保證
2. 自己對浮水印技術做充分的測試
3. 尋求可信賴的第三者，對浮水印技術做性能評定

然而只有第三種方式才能提供我們一個客觀無偏見的評定結果，而這也是浮水印標準評定程序（Benchmark）發展的目標之一。藉由一個好的標準評定程序，研究人員或發展數位浮水印技術的廠商可以提供一個具客觀性且值得信賴的數據結果，使用者可由這些數據結果來評估浮水印技術是否達到他們的基本需求，研究人員也可以藉著比較各種浮水印技術的數據結果來思考如何改進浮水印技術的性能。

針對應用在靜態影像上的數位浮水印技術，有幾套廣為大家所熟知的標準評定程序，如 StirMark、CheckMark、Optimark 等。而由於現今音訊浮水印技術的發展已越趨成熟，所以 StirMark 也開始發展針對數位音訊浮水印技術的標準評定程序（StirMark for Audio）[11]。為了對音訊浮水印技術的性能做一個評判，並了解其弱點所在何處，StirMark for Audio 建立了許多樂音訊號可能會遭受到的攻擊方式，並對這些攻擊方式做一連串的測試。這些攻擊方式大致可分為幾個種類：

- Dynamics : compressor、denoiser...
- Filters : high pass、low pass...
- Ambience : delay、reverb...
- Conversion : resampling、inversion...
- Noise : random noise...
- Modulation : chorus、enhance...

- Time stretch and pitch shift : pitch shifter、time stretch...
- Sample permutations : flip samples、cut samples...

在 StirMark for Audio 的發展測試過程中，有兩個主要的目標必須要確認，其一是測試哪一種音訊攻擊方式可以有效的破壞數位浮水印，另一點就是測試這些音訊攻擊是否會影響樂音訊號的感官聽覺品質，假若某種音訊攻擊方式可以有效的破壞數位浮水印，但是卻也會嚴重的影響樂音訊號的品質，那麼這種音訊攻擊方式就沒有太大的實用性。因此在 StirMark for Audio 的發展測試階段，是採用下述三階段的步驟：

1. Attack identification：先採用音訊編輯軟體如 CoolEdit 等以及前面提到的音訊攻擊方式對樂音訊號做修改，每一種攻擊方式皆有參數可以調整攻擊的強度，並將此參數調到最大且不嚴重影響音質的強度。
2. Watermark attacks：使用步驟 1 所完成的攻擊方式對幾種不同的音訊浮水印技術做測試，每一個測試的樂音訊號都是先嵌入數位浮水印後，再做攻擊測試。
3. Subjective tests：由步驟 1 及步驟 2 選出其中較有實用價值的攻擊方式，並對這些受攻擊後的樂音訊號做主觀的聽覺測試。

以其實驗結果來看[11]，其中最有效的一種攻擊方式是音調變化 (Pitch Shifting)，但是它也會造成樂音訊號品質的低落，所以它無法成為標準評定程序的一部份。在第五章的實驗結果中，我們將採用此套標準評定程序[12]來對本論文的數位音訊浮水印系統做測試。

2.4 數位音訊浮水印的發展

雖然音訊方面的浮水印發展，屬於剛起步的階段，文獻上音訊訊號的數位浮水印方法日漸增多，所使用的轉換領域和方法各不相同，以下將介紹幾種常見的音訊浮水印類型。

時域訊號中嵌入浮水印：

A.H. Tewfik[13] 是主要是利用 MPEG-1 Audio 的人耳聲學模型 (Psychoacoustic Model) 做探討，並利用其特性來做數位浮水印的嵌入，以防止加入的浮水印產生可被人耳察覺之雜訊，其流程如圖 2.1。

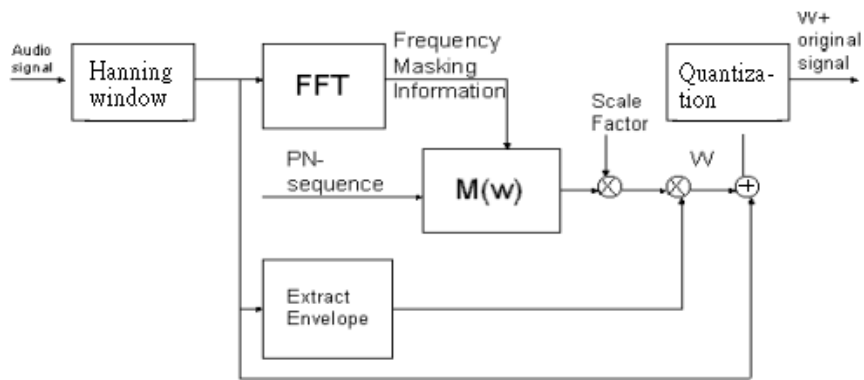


圖 2.1 Tewfik method 嵌入浮水印流程圖

首先將輸入之音訊訊號經過 1024 點之 hanning windows，其音框長度為 1024 個取樣點，再以音框為單位，利用人耳聲學模型算出最小遮蔽臨界值，如圖 2.2。

採用 MPEG-1 之人耳聲學模型，共分成 26 個關鍵頻帶，並利用其特性，將輸出的最小遮蔽臨界曲線當作一濾波器。將數位浮水印以一個位移暫存器 (Shift Register) 轉換成 PN sequences，其結構如圖 2.3，而 PN sequences 經過此濾波器後，PN sequences 的值會低於臨界值以下，再與時域主訊號相加，完成嵌入的程序，於是人耳就不會

察覺所加入 PN sequences，聽不見加入的資料。

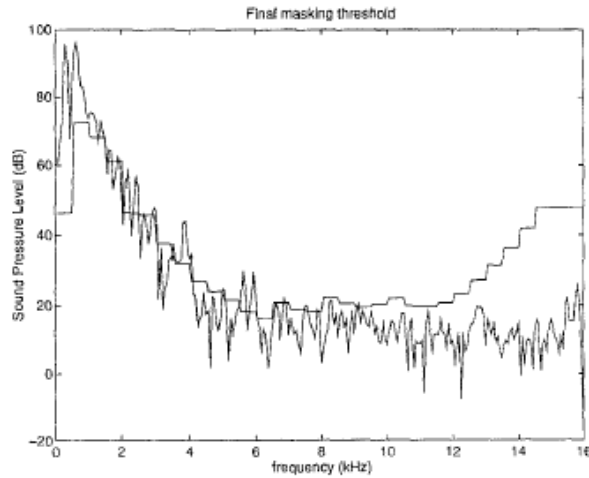


圖 2.2 PA model masking threshold 輸出

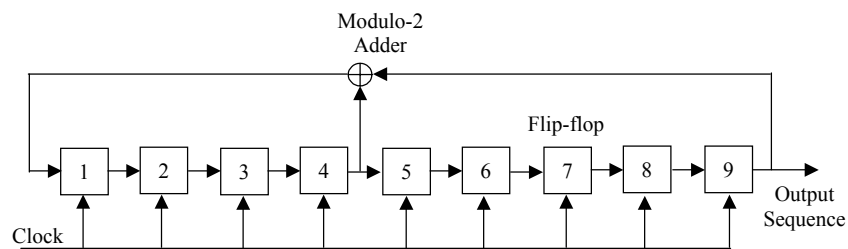


圖 2.3 PN sequences 產生圖

Tewfik 所提出架構之缺點為在萃取端需要原始之音訊訊號，在嵌入端每一個音框需要一濾波器來過濾浮水印，而在萃取端亦需要這個濾波器之反濾波器，複雜度太高，嵌入或萃取時運算速度都會受到影響。

頻域相位中嵌入浮水印：

Tilki[14]利用在訊號頻譜的相位嵌入數位浮水印，因為人耳對於聲音的絕對相位比其連續之相對相位不敏感，並依相位間之相對差值，還原其他音框之相位，使加入浮水印之影響降到最低。Tilki 所

提之架構，使用長度為 2048 點的 FFT 轉換為頻域訊號，並且在頻率高於 2 kHz 以上的部份，每 4 個頻率線就修改一次相位值，修改後的相位為 $\pm \pi/8$ ，如圖 2.4。

Tilki 所提出之方法由於是在固定位置的相位做固定大小的修改，所以較適合用來做複製管制或祕密通訊這類的應用。

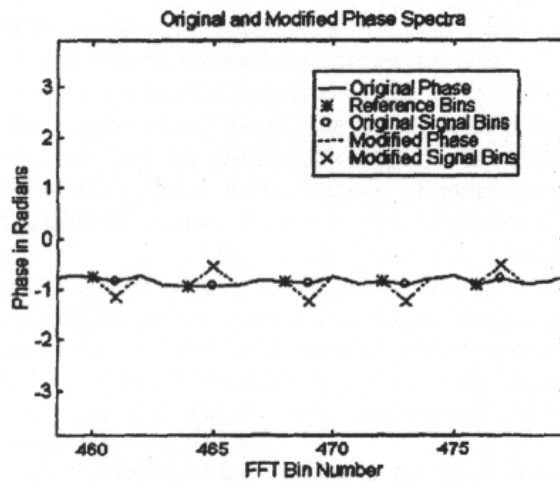


圖 2.4 頻譜相位修改示意圖

頻域中嵌入浮水印：

Wang[15]利用在 MDCT domain 加入數位浮水印，原始樂音先經由一個安全鑰匙做隨機重排 (Randomize)，如圖 2.5，再經過 MDCT、正規化 (Normalization) 與量化後，再加入浮水印，然後再經過反正規化與 IMDCT 和反隨機重排，其流程圖如圖 2.6。Wang 所提之嵌入浮水印架構與採用 MPEG-1 audio layer 3 相同，都是採用 MDCT domain，加入的數位浮水印可能會被遮蔽而不做編碼，故萃取時就會發生錯誤，所以此方法對於 MP3 的壓縮強健性較弱。

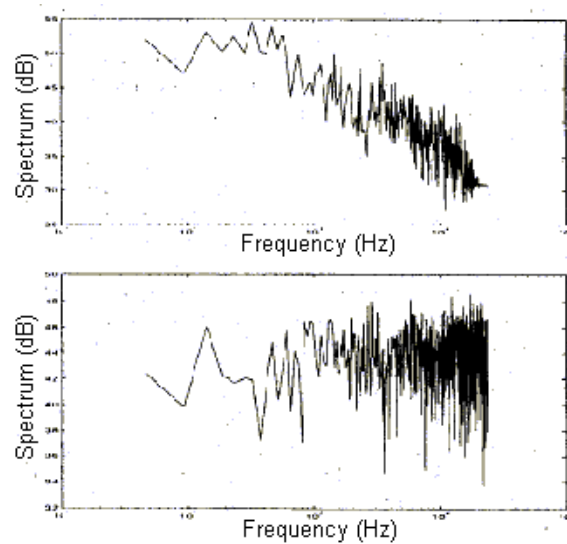


圖 2.5 host signal randomize

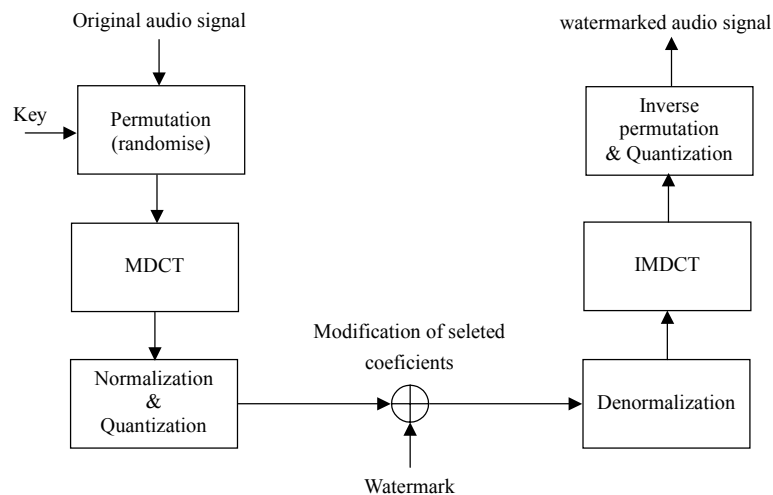


圖 2.6 Wang method 嵌入浮水印流程圖

嵌入迴音中的浮水印：

Druhl[16]提出利用在樂音訊號中加入迴音 (Echo) 的方式來嵌入數位浮水印，迴音的產生主要是因為時間上的延遲，後面的樂音訊號會有前面樂音訊號的分量，然而人耳有時間上的遮蔽效應，只要處理得宜，我們將不會聽到這些迴音訊號，Druhl 就是利用迴音訊號延遲時間的長短來嵌入 ”1”、”0” 的二元資料。

Hyen O Oh[17]改善了迴音訊號對於樂音音質的影響。以單一正

迴音（Positive Single Echo）的效果來看其脈衝響應（Impulse Response）及頻率響應，如圖 2.7 所示，其頻率振幅響應還是會有頗大的變化。

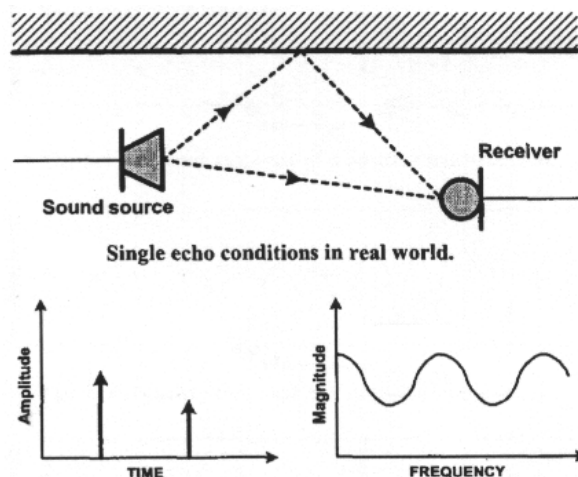


圖 2.7 Impulse response and frequency response of single echo

而 Hyen O Oh 提出多迴音（Multiple Echo）的架構，其中包含了正迴音及負迴音，改善了迴音訊號對於樂音音質的影響，如圖 2.8，其頻率振幅響應在低頻的部分幾乎沒有波動。

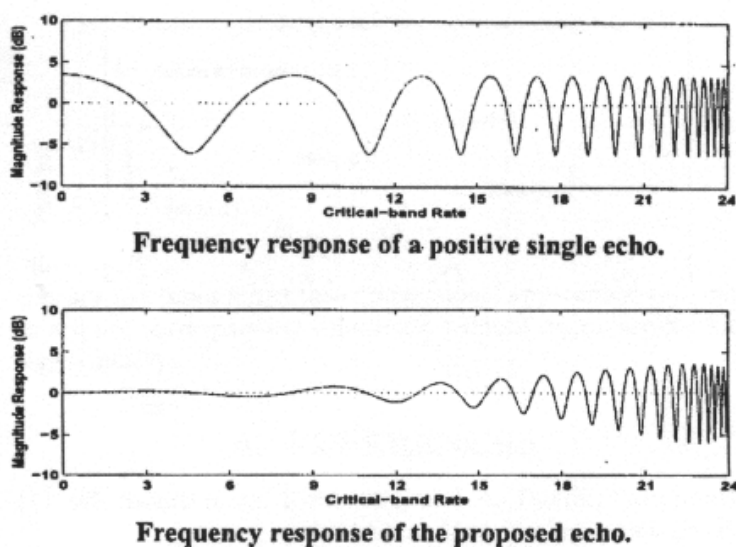


圖 2.8 Frequency response of multiple echo

與壓縮技術結合的浮水印：

C.-C. Tseng[18]提出利用結合線性預測（Linear Prediction）及向量量化（Vector Quantization）的方式來加入數位浮水印，藉由更改量化後的碼字索引（Codeword Index）並依其碼字索引為奇數或偶數來代表 ” 1 ”、” 0 ” 的二元資料。

J. Lacy[19]則是將數位浮水印的嵌入方式與 MPEG-II AAC 的音訊壓縮編碼技術相結合，藉由修改壓縮參數 Scale Factor 來嵌入浮水印。

第三章 小波分析與人耳聲學模型

近年來小波轉換 (Wavelet Transform) 已經被廣泛的使用於數位信號處理領域，包含音訊及視訊的壓縮等應用。本章將介紹小波轉換之基本原理[22]，並提出將應用於音訊浮水印之系統分頻架構。另外也將介紹用來模型化人耳對於聲音感受度的人耳聲學模型。

在 3.1 節中將會介紹小波分析技術; 3.2 節提出小波封包系統分頻架構; 3.3 節介紹人耳聲學模型; 3.4 節討論人耳聲學模型應用於小波係數時所要做的修正。

3.1 小波簡介

3.1.1 小波轉換 (Wavelet Transform)

相對於傅立葉轉換 (Fourier Transform) 針對週期性、非時變 (Time-invariant)、Stationary 的訊號做分析，且其基底波 (Wave or Sinusoids) 能量是無限的；小波分析提供了對暫態 (Transient)、時變 (Time-varying)、Nonstationary 之訊號分析，且小波的能量在時間軸上集中。

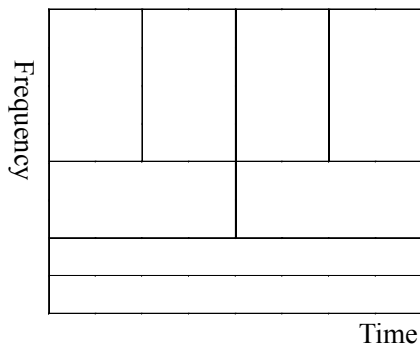


圖 3.1 小波的頻譜與時間
分割配置圖

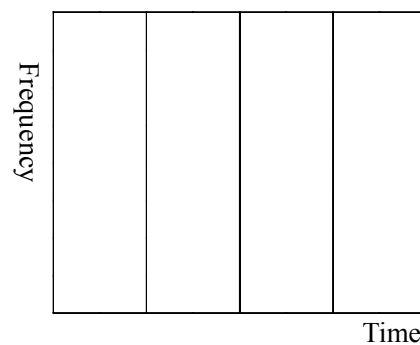


圖 3.2 傅立葉轉換的頻譜與時
間分割配置圖

小波分解 (Wavelet Expansion) 與離散小波轉換 (Discrete Wavelet Transform)

小波系統是一組表示信號或函式的係數方塊 (Building Blocks)。它是二維的分解系統，將信號以時間頻率位置 (Time-Frequency Location) 二維陣列的係數來表示。可分解的信號允許是一維或多維數的，通常只要擁有一組基底 (Basis)。當然，不同的基底，則有不同的分解係數，所以小波分解並不是唯一的。而這些係數組也就定義為該信號的小波轉換。如式 (3.1)， $\varphi_{j,k}(t)$ 是給定的小波基底， j 、 k 為整數， $a_{j,k}$ 為一組線性分解的係數。

$$f(t) = \sum_k \sum_j a_{j,k} \varphi_{j,k}(t) \dots\dots\dots (3.1)$$

所謂的第一代小波系統都是單純地由一個數量函式 (Scaling Function) 做 " Scaling " 和 " Translation " 所生成，如式 (3.2)。 k 提供了時間上的解析，而 j 提供了頻率上的解析。

$$\varphi_{j,k}(t) = 2^{\frac{j}{2}} \varphi(2^j t - k) \quad j, k \in Z \dots\dots\dots (3.2)$$

而離散小波轉換的一般式可寫成式 (3.3)，其中 j_0 可以是零或任何整數。

$$g(t) = \sum_k c_{j_0}(k) 2^{\frac{j_0}{2}} \varphi(2^{j_0} t - k) + \sum_k \sum_{j=j_0}^{\infty} d_j(k) 2^{\frac{j}{2}} \psi(2^j t - k)$$

$\varphi(t)$: scaling function, $c_j(k)$: scaling coefficient(3.3)
 $\psi(t)$: wavelet function, $d_j(k)$: wavelet coefficient

多重解析度 (Multiresolution) 分析

小波具有多重解析特性，適合做漸進式 (Progressive) 訊號壓縮或可調式編碼 (Scalable Coding) [23][24]。何謂多重解析特性呢？舉例來說，如果有一組訊號可以被 $\varphi(t-k)$ 的 Weighted Sum 所表示出來，就會有更多的一組訊號 (包含原來的) 可以被 $\varphi(2t-k)$ 的 Weighted Sum 所表示出來，換句話說，如果這個基底波 $\varphi(t)$ 的寬度變成原來的一半，而且 "Translation Step" 也是原來的一半的話，它就可以表示更多的訊號，或者說它可以將任何訊號表示得更精確。

小波轉換的基底函數 (Basis Function)，有兩種類型，一種是小波函數 (Wavelet Function)，一種是數量函數 (Scaling Function)；以離散小波轉換為例，它的小波函數和數量函數可以用上一精細解析度的 Translation 數量函數的 Weighted Sum 來表示，如式 (3.4)：

$$\varphi(2^j t) = \sum_n h(n) 2^{\frac{j+1}{2}} \varphi(2^{j+1} t - n), n \in Z \dots\dots\dots (3.4)$$

以 Wavelet 和 Scaling Function 為基底，便可以組成不同解析度的向量空間，如式 (3.5) 與 (3.6)，在 V_{j+1} 中 V_j 的正交補空間 (Orthogonal Complement) 定義為 W_j ，以 $V_j \perp W_j$ 表示，而 $V_\infty = L^2$ 包含了全部訊號空間， $V_{-\infty} = \{0\}$ 。向量空間之間的關係為： $V_{j+1} = V_j \oplus W_j$ 和 $V_{j+1} \supset V_j$ 對於所有屬於整數的 j 而言。

$$W_j = \text{span}\{\psi(2^j t - k), k \in Z\} \dots\dots\dots (3.5)$$

$$V_j = \text{span}\{\varphi(2^j t - k), k \in Z\} \dots\dots\dots (3.6)$$

圖 3.3 以 $j=0$ 為例的向量空間圖，式 (3.7) 為小波函數表示式，各向量空間的關係如式 (3.8)。

$$\psi(t) = \sum_n h_1(n) \sqrt{2} \varphi(2t - n), \quad n \in \mathbb{Z} \dots\dots\dots (3.7)$$

$$\begin{aligned} V_1 &= V_0 \oplus W_0, & V_0 &\perp W_0 \\ V_2 &= V_0 \oplus W_0 \oplus W_1, & V_1 &\perp W_1 \end{aligned} \dots\dots\dots (3.8)$$

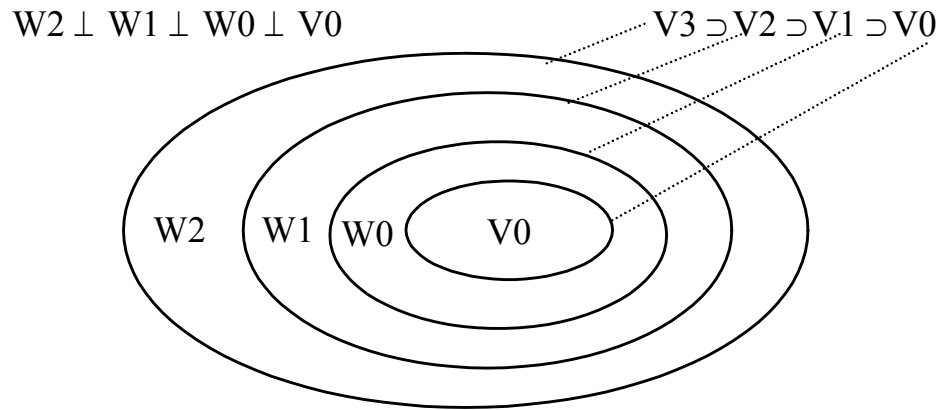


圖 3.3 小波分解多解析度空間圖

3.1.2 小波濾波器

離散小波轉換中，較低解析度（Lower Resolution）的係數可由較高解析度（Higher Resolution）的係數經由一樹狀結構的濾波器群組（Filter Banks）計算出，這個非常有效率的方法，使得小波轉換廣泛的應用在數位訊號處理上。

如圖 3.4 為一階小波分頻與合成濾波器。 $h_0(-n)$ 與 $h_1(-n)$ 為分析濾波器係數，較高規模係數經過與濾波器係數做疊合後，再經過次

取樣 (Down-Sampling or Decimating) 可得到較低規模係數。這樣的步驟可重複下去，形成數階 (Stage) 的濾波器群組架構。注意到一點，由於經過兩個濾波器，輸入訊號的個數增加為兩倍，經過次取樣，所以又回到原來的個數，這樣的意義在於沒有任何資訊的失去，有可能完全的回復至原來的訊號，這也是在濾波器群組理論裏完美重建 (Perfect Reconstruction) 的概念。而在重建時先做反次取樣 (Up-Sampling or Stretching)，意思是間隔地插入零值，使訊號長度加倍，再經過 $g_0(n)$ 與 $g_1(n)$ 的合成濾波器合成。

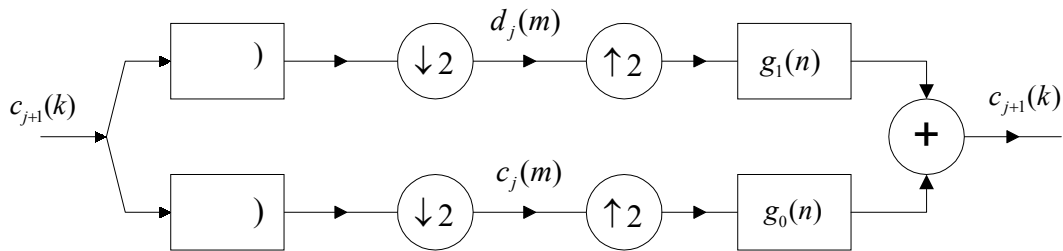


圖 3.4 一階小波分頻與合成濾波器

數量函數在規模夠高的時候會形同脈衝函數 (Delta Function)，當我們求其分解係數時就如同做取樣一樣，所以若信號的取樣頻率大於 Nyquist Rate 也就是頻寬的兩倍，它的取樣點就類似是很好的 Scaling Coefficients，這也是我們可利用小波濾波器來作訊號處理的原因之一。

在小波濾波器的選擇上，根據研究[25][26]，發現 Biorthogonal 的小波濾波器特性較適用於樂音分析，因其小波係數的能量較集中且為 Linear Phase，因此我們採用 Daubechies Biorthogonal 小波濾波器 [27]，係數長度為 18，圖 3.5 分別為分解與合成的高通低通濾波器係數值，其振幅響應特性曲線如圖 3.6 所示。而各濾波器的群組延遲 (Group Delay) 如圖 3.7。

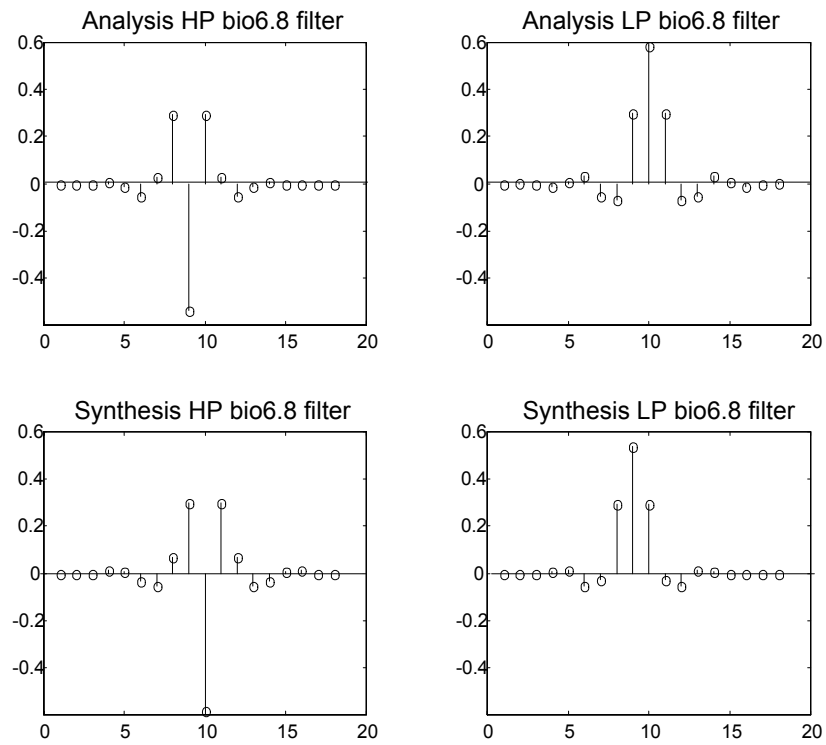


圖 3.5 Daubechies Biorthogonal 6.8 小波濾波器係數

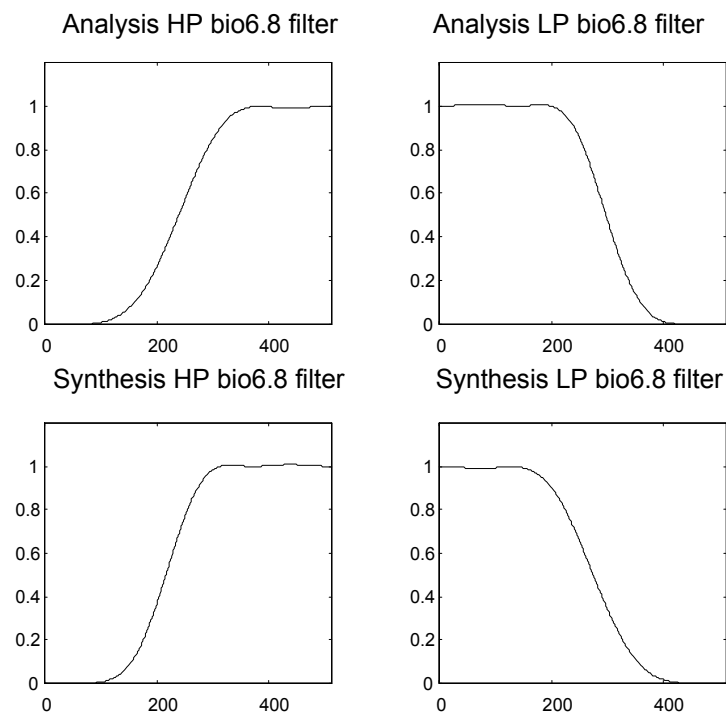


圖 3.6 Daubechies Biorthogonal 6.8 小波濾波器之振幅響應

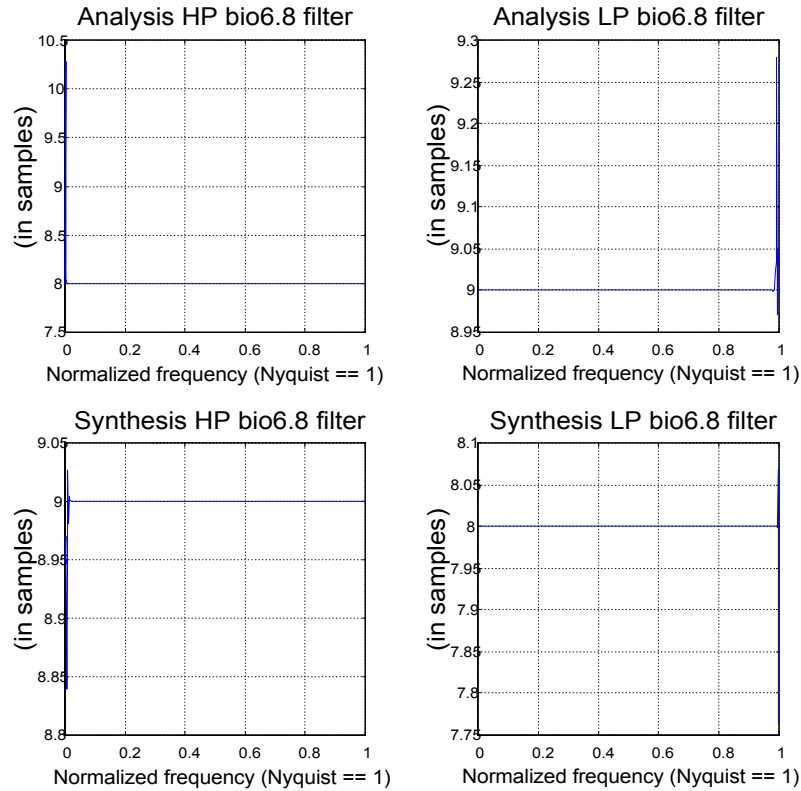


圖 3.7 Daubechies Biorthogonal 6.8 小波濾波器的群組延遲

3.1.3 小波封包 (Wavelet Packet)

小波封包系統是由 Ronald Coifman 所提出[28]，其允許在高頻帶的部分繼續做更細微的切分，與傳統只在低頻帶部分做進一步切分的小波轉換系統不同。它提供了更豐富的頻帶樹狀結構，可適用於更多種類的訊號分析。但是它所付出的代價是運算複雜度的增加，由原本小波轉換的 $O(N)$ 提高為 $O(N\log(N))$ ，複雜度類似 FFT (FFT 之複雜度為 $N\log(N)$)，其中 N 為濾波器的係數長度，定義為係數個數。

以下圖 3.8 與 3.9 可以比較出小波轉換與小波封包之不同處。

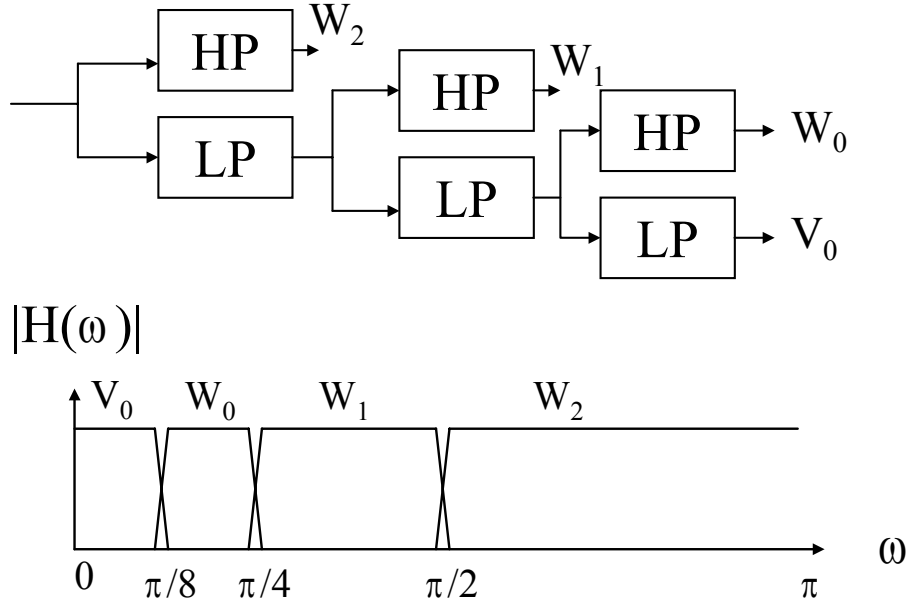


圖 3.8 兩個濾波器組的三階小波轉換系統與其 “Constant-Q” 頻帶解析

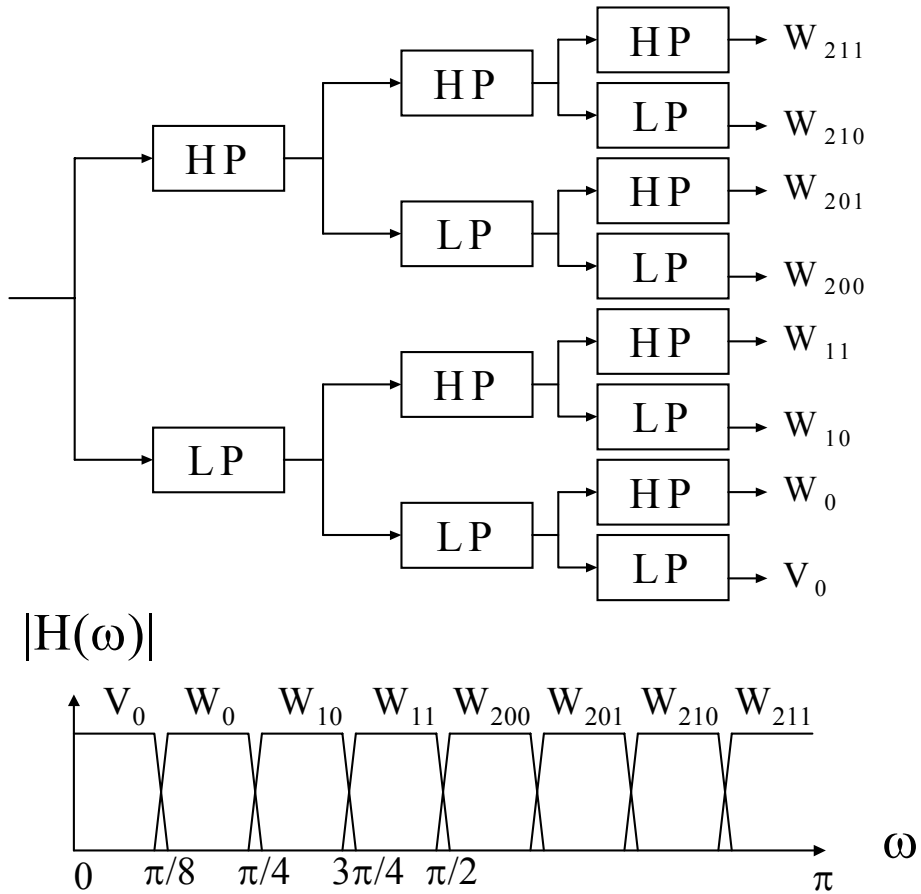


圖 3.9 三階小波封包系統的完整二元樹結構 (Full Binary Tree)

與其頻帶解析

3.2 小波分頻架構

為了配合人耳聲學模型（見 3.3 節）做數位浮水印嵌入，分次頻帶時愈接近關鍵頻帶愈好，舉例來說明，若某一個次頻帶頻寬同時包含好幾個關鍵頻帶在內，它所取的最小遮蔽臨界值是它所包含的關鍵頻帶中比較小的值，如此便犧牲掉了比較大的值，這樣一來是沒有好處的，因為較大的遮蔽效應可以增進浮水印的強健性。

所以我們採用的分頻架構如圖 3.10。採用 $M=2$ 的小波濾波器群組，共 8 階 29 個次頻帶的小波封包結構。

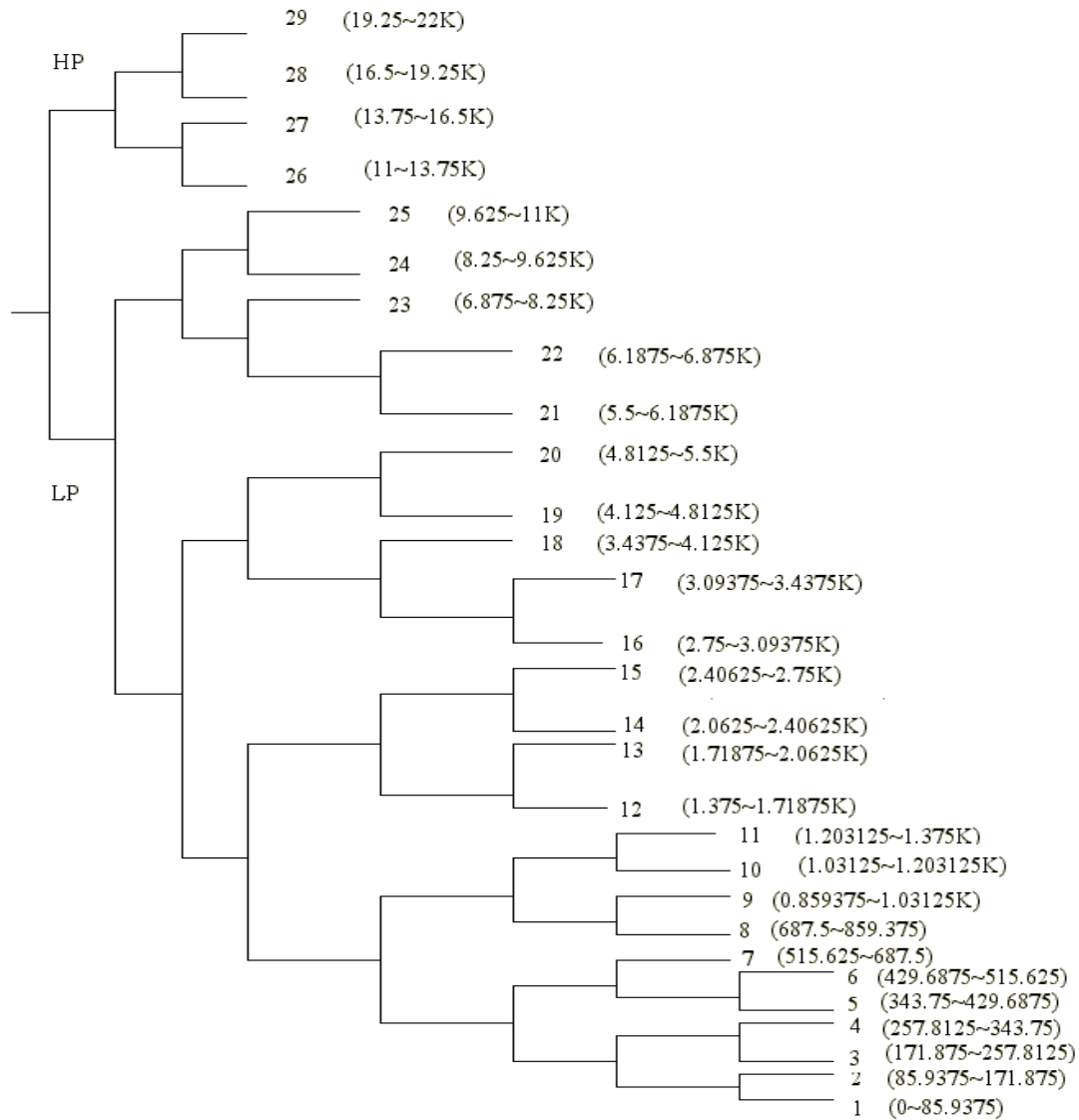


圖 3.10 小波封包之各次頻帶頻寬分布(單位 Hz)

將一般樂音的頻帶範圍 0~22kHz 分成頻寬不等的 29 個次頻帶,其與 MPEG 標準中人耳聲學模型的 26 個關鍵頻帶 (Critical Bands) (見 3.3 節) 的相對關係如圖 3.11 所示。

圖 3.12 至 3.15 將實際樂音訊號的小波係數值列出，可觀察出係數能量的分布情形都非常的集中。

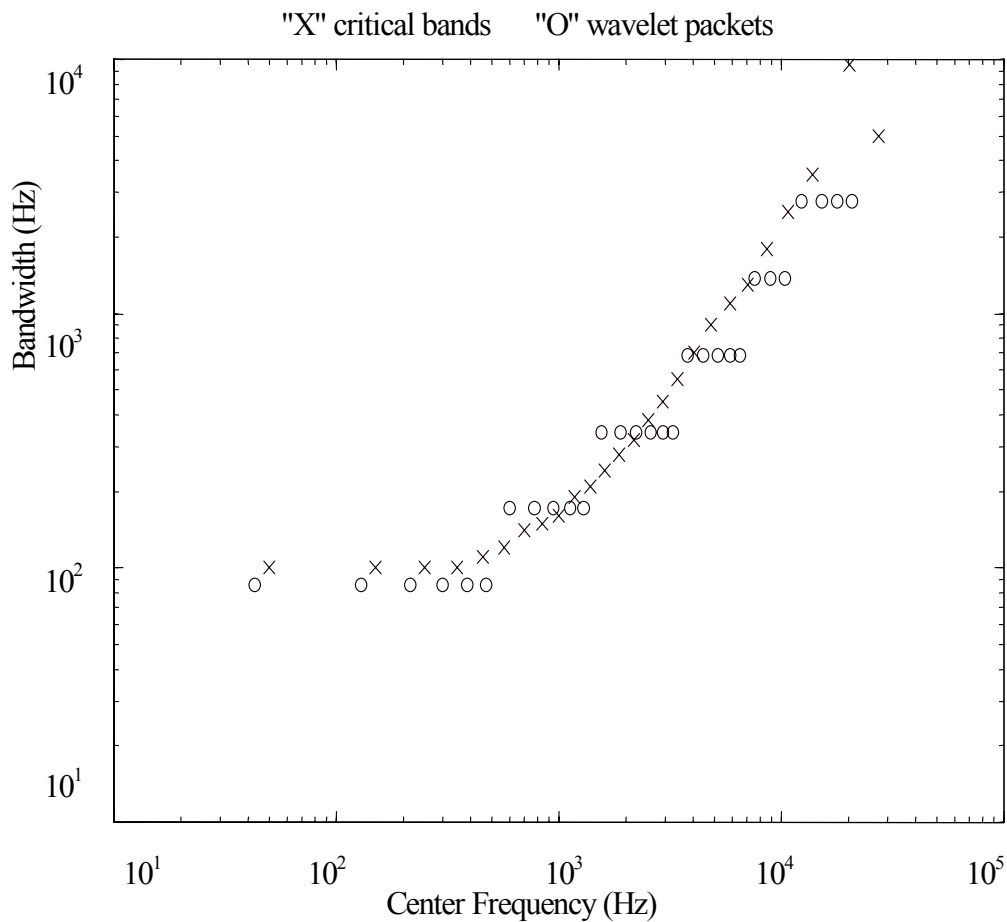


圖 3.11 小波封包次頻帶與 MPEG 人耳聲學模型關鍵頻帶的關係

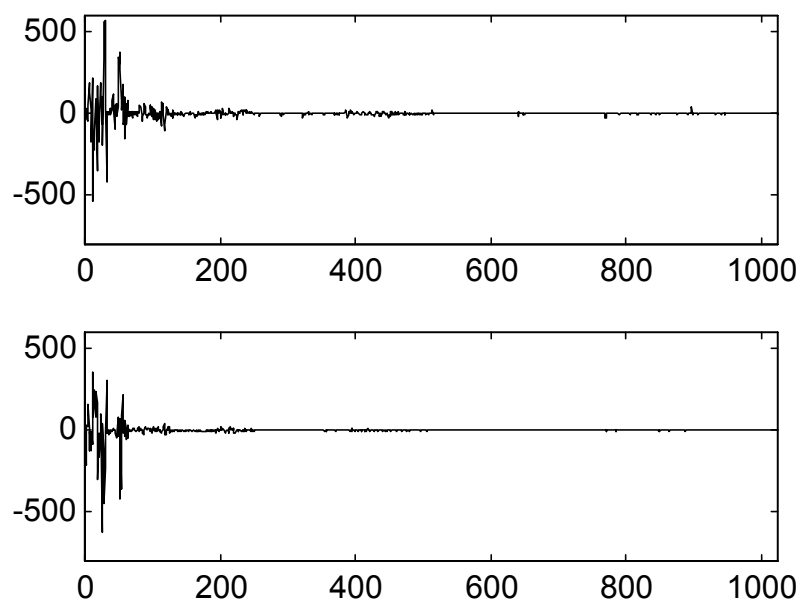


圖 3.12 Flute 訊號第一、二個音框之小波係數

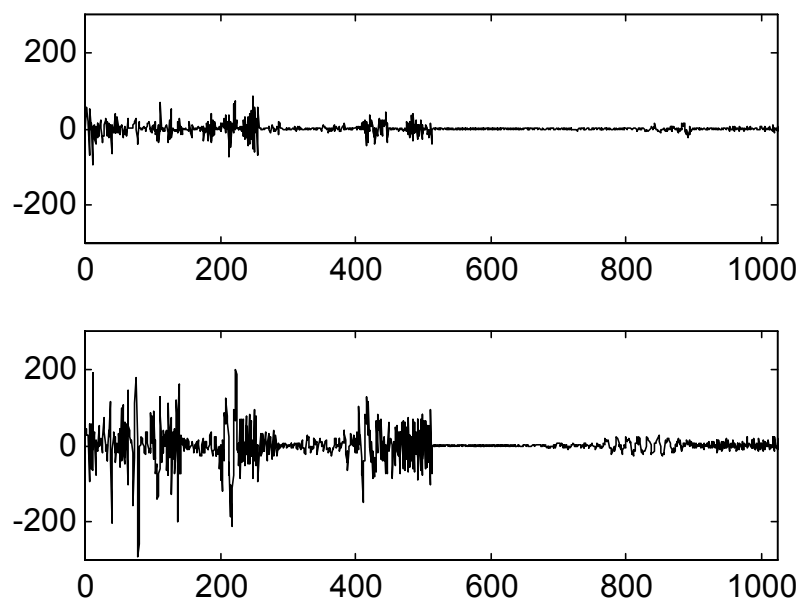


圖 3.13 Organ 訊號第一、二個音框之小波係數

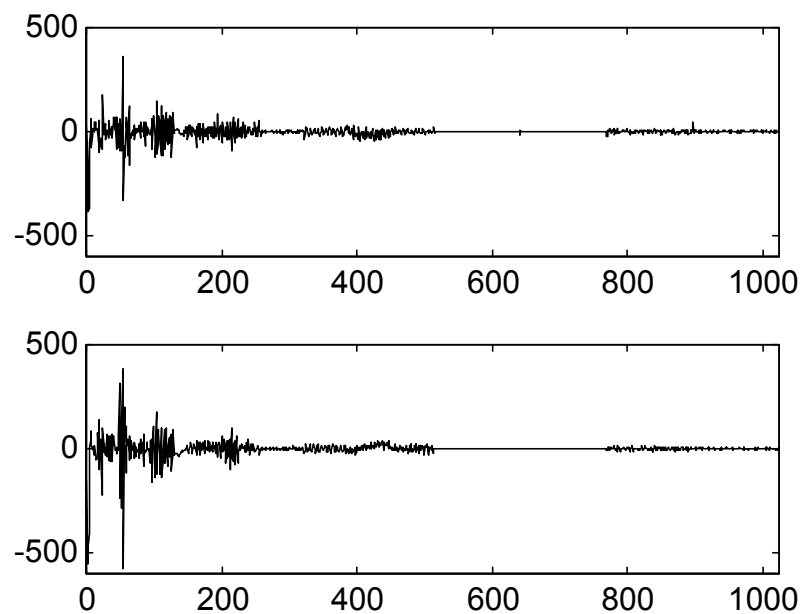


圖 3.14 Harpsichord 訊號第一、二個音框之小波係數

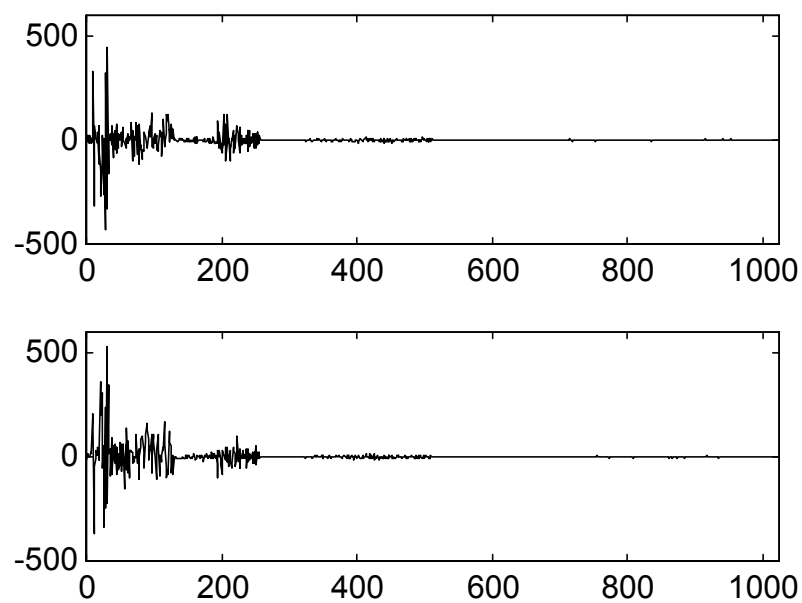


圖 3.15 Violin 訊號第一、二個音框之小波係數

3.3 人耳聲學模型

多年以來人類聽覺系統 (Human Auditory System) 已被深入的研究，由於人耳構造的緣故，使得人類所能聽見的聲音頻率範圍在 20 Hz ~ 20 kHz 之間。而人耳對聲有非線性反應的特性，對於每個頻率的敏感度不同，因此就發展出人耳聲學模型來模型化人耳對聲音的反應感受度。

3.3.1 基本原理與其應用

人類所能聽見的聲音頻率範圍在 20 Hz ~ 20 kHz 之間，而對於中低頻聲音的敏感度遠大於高頻聲音，除此之外，還有一種有趣的性質存在，就是聲音的遮蔽效應 (Masking Effect)，可分為兩方面，一為頻率軸上，一為時間軸上，也就是說當某遮蔽源 (Masker) 存在時，我們將聽不見它所遮蔽住的特定頻率範圍 (或特定時間內)、特定強度下的任何聲響；而最值得注意的是關鍵頻帶 (Critical Band)，其意義是，在某些固定的頻寬範圍之內遮蔽效應的模式是幾乎不變的，其頻寬範圍如式 (3.9) 所示，而在頻率軸上對應一個關鍵頻帶的單位刻度被定義為巴克 (Bark)，近似的轉換公式如式 (3.10)，式中 f 表示欲轉換之頻率值 (單位 Hertz)， B 表示轉換後之巴克值；表 3.1 說明關鍵頻帶的範圍，以標準頻率刻度赫茲 (Hertz) 表示。

$$BW(f) = \begin{cases} 100 \text{ Hz} & f < 500 \text{ Hz} \\ 0.2 f \text{ Hz} & f \geq 500 \text{ Hz} \end{cases} \dots\dots\dots (3.9)$$

$$B = 13.0 \tan^{-1} \left(\frac{0.76f}{1000} \right) + 3.5 \tan^{-1} \left(\left(\frac{f}{7500} \right)^2 \right) \dots\dots\dots (\text{Barks}) \dots\dots\dots (3.10)$$

表 3.1 關鍵頻帶的中央與邊界頻率

Band number	Lower edge (Hz)	Center (Hz)	Upper edge (Hz)
1	0	50	100
2	100	150	200
3	200	250	300
4	300	350	400
5	400	450	510
6	510	570	630
7	630	700	770
8	770	840	920
9	920	1000	1080
10	1080	1170	1270
11	1270	1370	1480
12	1480	1600	1720
13	1720	1850	2000
14	2000	2150	2320
15	2320	2500	2700
16	2700	2900	3150
17	3150	3400	3700
18	3700	4000	4400
19	4400	4800	5300
20	5300	5800	6400
21	6400	7000	7700
22	7700	8500	9500
23	9500	10500	12000
24	12000	13500	15500
25	15500	19500	25000
26	25000	27500	30000

頻率軸上的遮蔽效應

其遮蔽源有很多種類，大致上以頻率範圍來區分，包括廣頻雜訊、窄頻雜訊及單頻音等等，而被遮蔽音（The Maskee, or Probe）通常以單頻音來作測試，方便測出頻率分布範圍及聲音強度大小，如圖 3.16 為窄頻雜訊遮蔽效應的例子[29]。

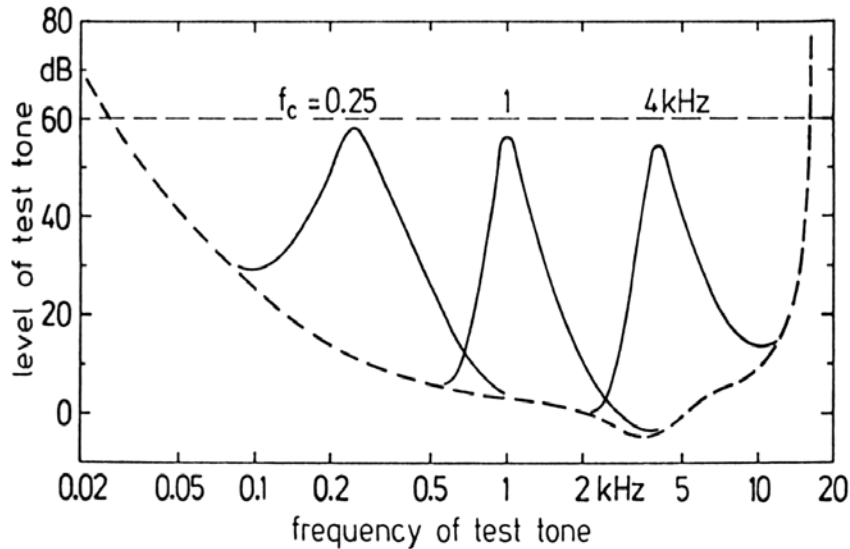


圖 3.16 窄頻雜訊的遮蔽效應

由上圖可看出強度為 60dB 的窄頻雜訊對不同頻率的單頻音所產生的遮蔽效應。圖中曲線各別表示被中央頻率 $f_c = 0.25$ 、1、4 kHz，強度 60dB 的關鍵頻寬雜訊恰好遮蔽住的單頻音強度，而橫軸表示被測試的單頻音頻率。中央頻率愈高的窄頻雜訊所能遮蔽住的單頻音的頻率分佈範圍愈廣。

而圖中下方的虛線表示完全安靜時的最小能聽見度 (Threshold In Quiet or Absolute Threshold of Hearing (ATH))，意指人耳處於安靜狀態時所能聽到的最小聲音強度。可發現人類聽覺最敏銳的頻帶範圍在 2 ~ 5 kHz 之間。

時間軸上的遮蔽效應

圖 3.17 顯示出時間軸上的遮蔽效應。分為前遮蔽(Pre-Masking)，同步遮蔽(Simultaneous Masking) 與後遮蔽(Post-Masking) 三種效應，注意後遮蔽在橫軸上使用不同的起點。前遮蔽的發生時間於遮蔽源產生前大約 50 ms，所能遮蔽住的強度隨著愈接近遮蔽源而提高；

後遮蔽延續的時間為遮蔽源消失後的 150 ms 左右，所能遮蔽住的強度隨著時間減低。

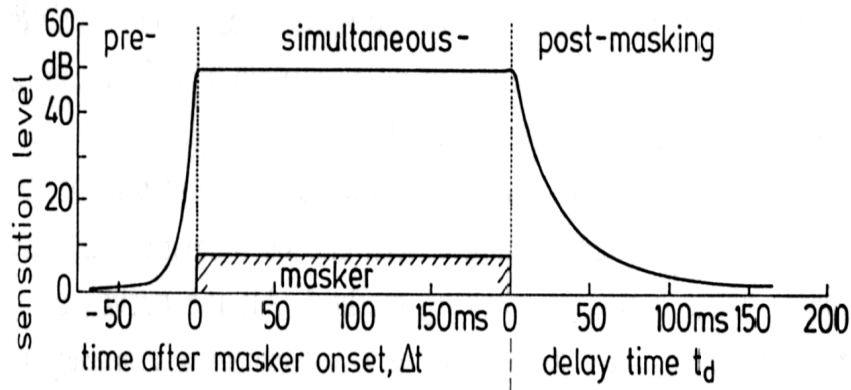


圖 3.17 時間軸上的遮蔽效應

對於遮蔽效應有了瞭解，可以將它應用在聲音編碼或音訊浮水印技術上，只要樂音的變化小於最小遮蔽臨界值，則可以達到聽覺上不失真的效果，這也就是人耳聲學模型的精神所在，它用數學模型模擬出各種遮蔽效應的總和，得到最小遮蔽臨界值，供編碼器作為位元配置的依據、或是浮水印技術更動樂音訊號的依據。當然，有各種複雜度不同的模型架構被提出，通常愈複雜的模型所得到的結果也愈精細縝密。而 MPEG Audio Psychoacoustic Model II [30]可謂是相當複雜的一個模型。

3.3.2 模型公式

人耳聲學模型中最簡單的就是絕對聽力臨界值 ATH (Absolute Threshold of Hearing)了，人類總是聽不見太微弱的聲音，式 (3.11) 模擬出此曲線，式中 f 單位 Hz，ATH 以 dB 表示。

$$ATH(f) = 3.64 \left(\frac{f}{1000} \right)^{-0.8} - 6.5 e^{-0.6 \left(\frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left(\frac{f}{1000} \right)^4 \dots\dots\dots (3.11)$$

而 MPEG Psychoacoustic Model II 以查表方式，讀進 513 個 ATH 值，如圖 3.18 所示。

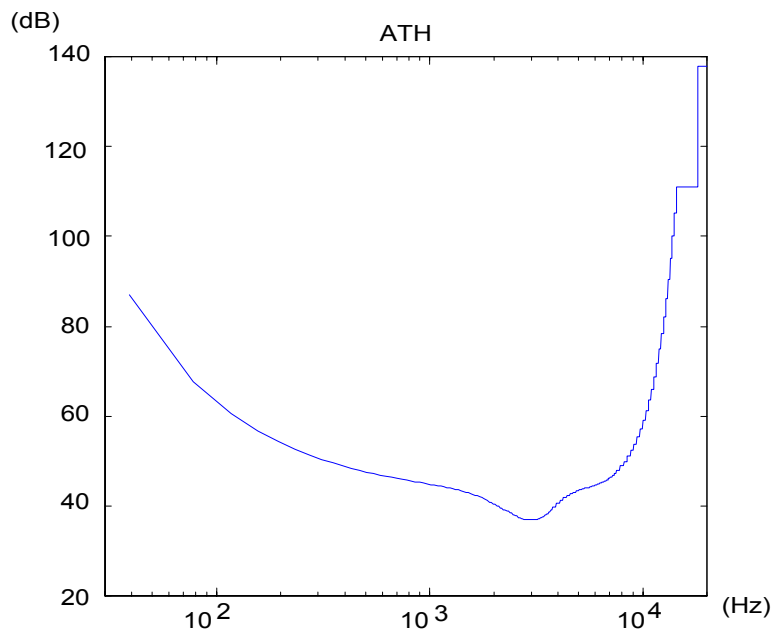


圖 3.18 MPEG Psychoacoustic Model II 之絕對聽力臨界值 ATH

遮蔽效應的影響並不只是侷限在單一的關鍵頻帶中，他也會對鄰近的頻帶產生影響，延展函式 (Spreading Function) 考慮了本身次頻帶 (Inner-Band) 及相互次頻帶 (Inter-Band) 的遮蔽效應，如式

(3.12)，式中 z 代表頻率刻度，以巴克為單位。若將較高遮蔽源的漸減的斜率也考慮進去，則可將上式修改為式 (3.13)，式中 $F(f)$ 為訊號之傅立葉轉換。

$$sf(z) = 15.81 + 7.5(z + 0.474) - 17.5\sqrt{1 + (z + 0.474)^2} \dots\dots\dots(3.12)$$

$$SF(z, i) = (15.81 - i) + 7.5(z + 0.474) - (17.5 - i)\sqrt{1 + (z + 0.474)^2} \dots\dots\dots(3.13)$$

where $i = \min(5 \cdot |F(f)| \cdot BW(f), 2.0)$

然而我們知道雜訊遮蔽源與單頻音遮蔽源的遮蔽效應是大大的不同的，而在對訊號編碼時，所產生的量化誤差被視為雜訊，所以我們只需預測遮蔽源的 " Tonality " 來找出遮蔽臨界值。而 $F(f)$ 的 " Tonality " 須靠預測得到，相位預測器如式 (3.14)，由前兩個時間點的係數相位得到；振幅預測器如式 (3.15)，就簡單的當作跟上一個相同，最後 " Tonality " 由前兩個相位預測誤差 (式 (3.16)) 與前一個振幅預測誤差 (式 (3.17)) 的最大值估算出，如式 (3.18) 所示， $t(f)$ 的值介於 0 到 1 之間。

$$\hat{\varphi}_f = 2 \cdot \arg(F_{f-1}(f)) - \arg(F_{f-2}(f)) \dots\dots\dots(3.14)$$

$$\hat{M}_f = |F_{f-1}(f)| \dots\dots\dots(3.15)$$

$$\varphi_{err_i} = \hat{\varphi}_f - \arg(F_f(f)) / \pi \dots\dots\dots(3.16)$$

$$M_{err_i} = (\hat{M}_f - |F_f(f)|) / \max(F_f(f), \hat{M}_f) \dots\dots\dots(3.17)$$

$$t(f) = 1 - \max(\varphi_{err_i}, \varphi_{err_{i-1}}, M_{err_i}) \dots\dots\dots(3.18)$$

有了延展函數與 " Tonality "，可得到遮蔽臨界值如式 (3.19)，式中 z_m 為遮蔽源的頻率， z 為被遮蔽的頻率，以巴克為單位， $P(z_m)$ 為遮蔽源的能量，單位是 dB，而 $k(t)$ 與 $l(t)$ 是以純雜訊(Pure Noise) 與單頻

音(Pure Tone) 實驗得來的函數，如式 (3.20) 與 (3.21)。

$$M(z) = P(z_m) + SF(z - z_m) - k(t(z))z_m - l(t(z)) \text{ (db)} \dots\dots\dots(3.19)$$

$$k(t) = 0.3t + 0.5(1 - t) \dots\dots\dots(3.20)$$

$$l(t) = 34t + 20(1 - t) \dots\dots\dots(3.21)$$

由不同遮蔽源而來的遮蔽效應必須總和起來，到底是加成能量或絕對振幅值呢？答案是將振幅絕對值的平方根加起來如式 (3.22)，因為若將其直接相加，有可能造成比遮蔽源還大的值。但假使兩者差距很大的話，則可以直接相加，如式 (3.23)。

$$SUM(M_1, M_2) = 40 \log \left(10^{M_1/40} + 10^{M_2/40} \right) \text{ (db)} \dots\dots\dots(3.22)$$

$$SUM(M_1, M_2) = 20 \log \left(10^{M_1/20} + 10^{M_2/20} \right) \text{ (db)} \dots\dots\dots(3.23)$$

以上所述為同步遮蔽 (Simultaneous Masking) 的模型，接著談實現後遮蔽 (Post-Masking) 的模型，以一個單極點 (One-Pole) 的濾波器達成，如式 (3.24)。

$$M'_t = 0.85M'_{t-1} + 0.15M_t \dots\dots\dots(3.24)$$

最後的遮蔽效應輸出值為 $\max(M'_t, M_t)$ ，所以後遮蔽要有維持較長的訊號才能發揮作用。

由於前遮蔽所發生的時間較短，無法像前述之後遮蔽這樣運用，但它還是非常重要的，尤其是對 " Pre-Echo " 效應的遮蔽，所謂 " Pre-Echo " 效應來自於對轉換係數的量化在時間軸上所產生的雜

訊，如下圖所示，這樣的雜訊對於暫態的訊號（如鼓聲）尤其明顯，是我們耳朵可聽得見的，若轉換的 Block 夠小，這種雜訊的效應是可以被前遮蔽所隱藏住的。

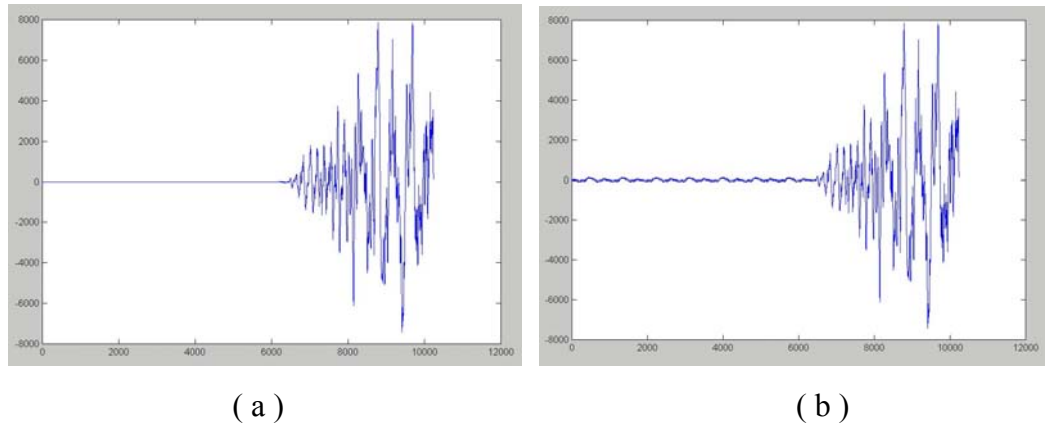


圖 3.19 Pre-Echo 例子，(a) 為未編碼前，(b) 時域/頻域轉換編碼後，在本來無訊號的時間有雜訊產生

3.3.3 訊號之各頻帶的最小遮蔽臨界值

本節將訊號的最小遮蔽臨界值列出以茲參考，其乃根據 MPEG Psychoacoustic Model II，將訊號的每個音框的 512 個遮蔽臨界值算出，再依據系統分頻架構，將 512 條頻率線分成 29 組，每一組中取出最小值，也就是將 29 個次頻帶的最小遮蔽臨界值取出，舉例來說，第一個次頻帶包含 2 個頻率線，第七個次頻帶包含 4 個頻率線，由其中取出最小值，如圖 3.20 所示，其他的次頻帶以此類推。

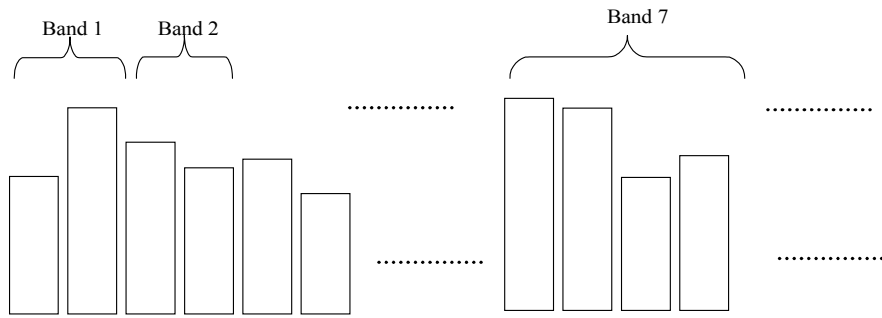


圖 3.20 最小遮蔽臨界值取出示意圖

每個音框的輸出值不一，但其趨勢是相同的。圖 3.21 為管風琴樂音訊號的第 2、42、82 個音框的 512 個臨界值，圖 3.22 為由其中取出的 29 個次頻帶的最小遮蔽臨界值；而圖 3.23 為長笛樂音訊號的第 2、42、82 個音框的 512 個臨界值，圖 3.24 為由其中取出的 29 個次頻帶的最小遮蔽臨界值。

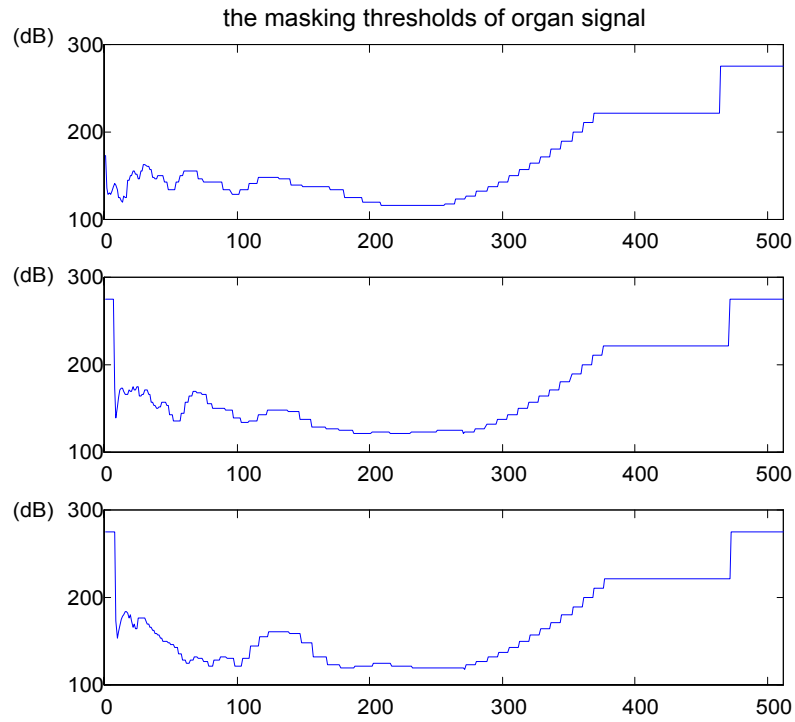


圖 3.21 Organ 訊號第 2、42、82 個音框在頻率上的遮蔽曲線

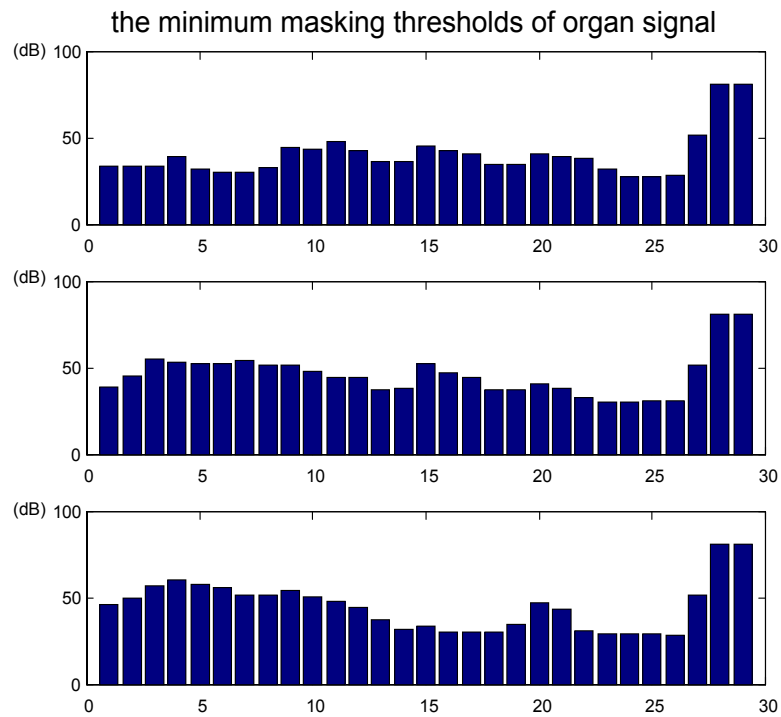


圖 3.22 Organ 訊號第 2、42、82 個音框每個次頻帶的最小遮蔽臨界值

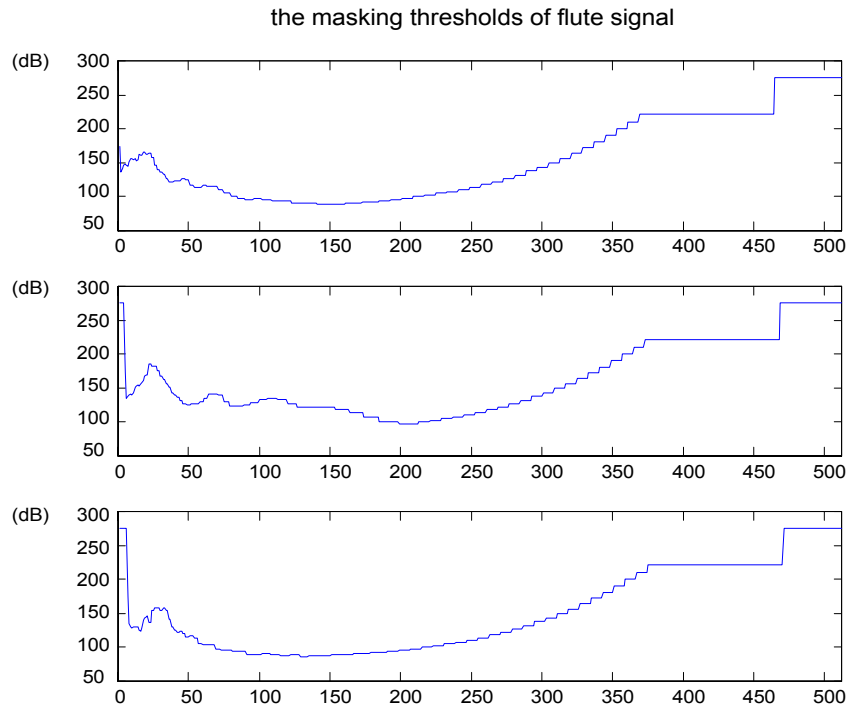


圖 3.23 Flute 訊號第 2、42、82 個音框在頻率上的遮蔽曲線

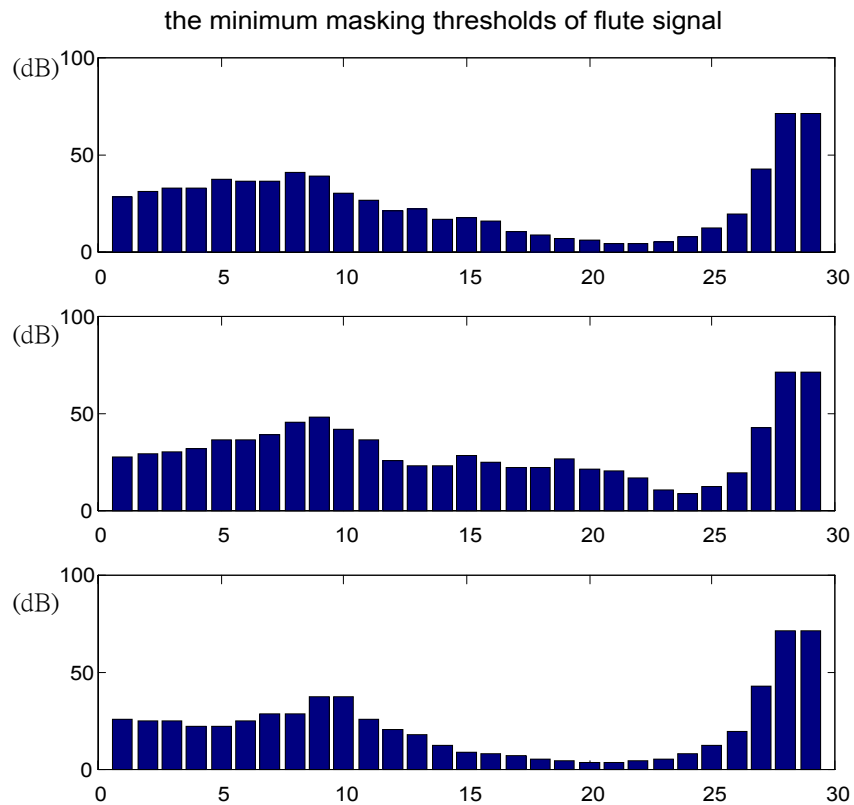


圖 3.24 Flute 訊號第 2、42、82 個音框每個次頻帶的最小遮蔽臨界值

3.4 轉換頻率遮蔽曲線至小波域

在 3.3 節中所討論的聲學模型，其乃根據 MPEG Audio Psychoacoustic Model II 所推算而得到。但是由於小波濾波器組的特性，其小波係數的能量較為集中，所以由 MPEG Audio Psychoacoustic Model II 所推算出的最小遮蔽臨界值，並不適合直接的使用在小波係數上，它必須先經過一個由頻率域轉換到小波域的步驟以決定每個小波次頻帶所能接受的雜訊功率（Noise Power）大小。

要將頻率域的遮蔽臨界值轉換至小波係數使用，在[31]中提出了下列的方法，如式（3.25）所示。

$$\sigma_i^2 = \min_{\omega \in B_i} \left(\frac{T(e^{j\omega})}{\sum_{k=1}^M |F_k(e^{j\omega})|^2} \right) \dots\dots\dots (3.25)$$

式中 σ_i^2 表示第 i 個次頻帶所可以接受的雜訊功率， B_i 表示位於第 i 個次頻帶內的所有頻線集合， $T(e^{j\omega})$ 為頻率域中的遮蔽臨界值， $|F_k(e^{j\omega})|$ 表示在小波分頻樹狀結構中第 k 個分枝的合成濾波器的振幅響應， M 表示次頻帶個數。

藉由式（3.25）對由 MPEG Audio Psychoacoustic Model II 所推算出的遮蔽臨界值做一修正，可使得此修正過後的最小遮蔽臨界值更適合應用在小波係數上。將修正後的最小遮蔽臨界值列出以茲參考，圖 3.25 為管風琴樂音訊號的第 2、42、82 個音框的 29 個次頻帶的最小遮蔽臨界值；而圖 3.26 為長笛樂音訊號的第 2、42、82 個音框的 29 個次頻帶的最小遮蔽臨界值。

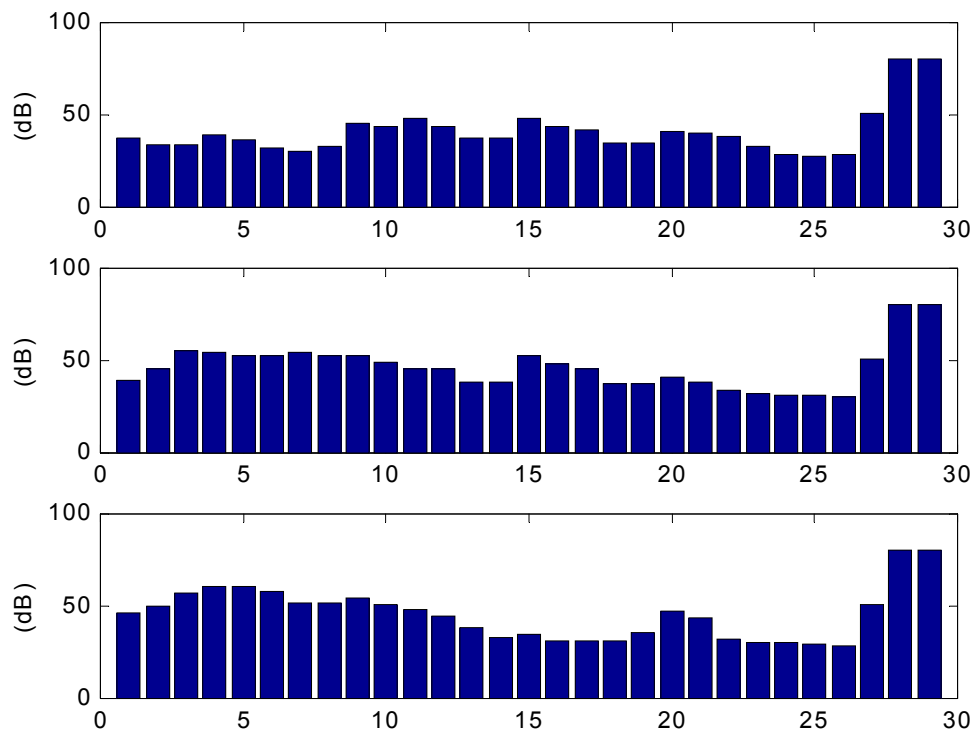


圖 3.25 Organ 訊號第 2、42、82 個音框每個次頻帶的最小遮蔽臨界值

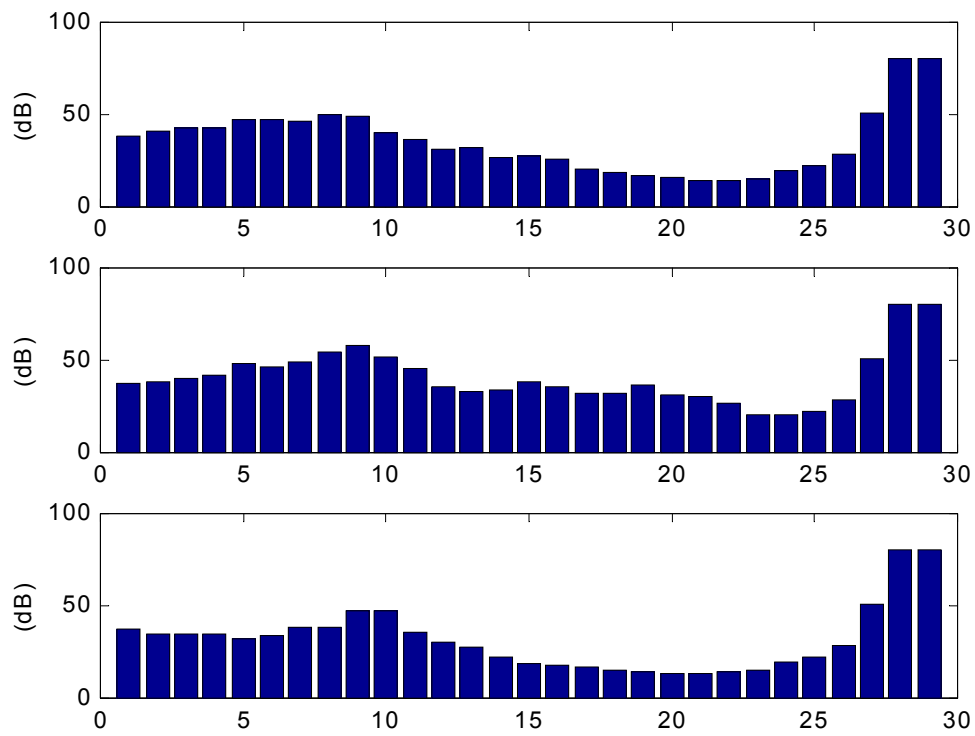


圖 3.26 Flute 訊號第 2、42、82 個音框每個次頻帶的最小遮蔽臨界值

第四章 小波域數位音訊浮水印

近年來隨著小波的快速發展，運用小波分頻與人耳聲學模型在樂音系統上的研究，已有重大的發展，可達成低位元率的傳輸。1999 年，M.-T. Chou 提出以小波封包分頻與零樹編碼配合人耳聲學模型的壓縮系統[32]，其音質相當於 MPEG -1 layer 2 之品質。2000 年，R.-S. Lin 提出多層漸進式零樹小波分頻音訊技術[33]，可針對網路之不同傳輸頻寬來選擇其適當之位元率來傳輸，其位元率分別為 16 kbps、32 kbps、64 kbps。2001 年 D.-K. Lien 提出資料隱藏應用於零樹小波分頻壓縮系統的技術[34]，並將其應用在音訊封包遺失回復技術上，能有效改善封包遺失對樂音所造成的影響。小波分析除了可以有有效的應用在音訊、視訊、影像的壓縮編碼之外，其在數位浮水印的應用上也可以達到非常好的效果。2001 年，L.-L. Lee 提出小波及離散餘弦域上使用子區塊分類之金匙數位影像浮水印系統[35]，其對於影像壓縮（JPEG、SPIHT、JPEG2000）以及影像裁切攻擊都有不錯的強健性。

因此本論文基於小波轉換在數位影像浮水印上的研究[35]，將其應用在數位音訊浮水印的技術上。在數位音訊浮水印系統中最重要的一項莫過於如何能將數位浮水印隱藏入樂音訊號中而又不影響人耳對於樂音訊號品質的感受度。如 3.3 節所述，人類聽覺系統畢竟不是一個完美的接收器，人耳對於樂音訊號在不同的頻率有不同的感受度，因此就如同大多數的音訊壓縮技術一樣，在頻域上或小波域上發展的音訊浮水印也勢必要將此一現象做為一個重要依據，如此才能兼顧數位浮水印的強健性與樂音訊號的品質。

本論文的想法是以小波封包分頻配合人耳聲學模型發展一套可

兼顧數位浮水印強健性與樂音訊號品質的音訊浮水印技術，並且為了抵抗音訊裁切攻擊造成浮水印萃取時音框起點無法對正（Alignment）的問題，加入同步碼搜尋對正音框起點的機制。浮水印嵌入方塊圖如圖 4.1，浮水印萃取方塊圖如圖 4.2。浮水印嵌入方式是先將樂音訊號分割成 1024 個取樣點的音框，再對音框做小波封包處理，並同時利用人耳聲學模型對樂音做分析，在嵌入浮水印之前，先嵌入特定的同步碼，之後再依據欲嵌入之數位浮水印及小波係數的特性，做數學邏輯運算產生一組二元安全鑰匙，最後將修改過的小波係數還原成時域訊號即為藏有浮水印的樂音訊號。而浮水印萃取方式則是先搜尋到正確的同步碼後，再利用安全鑰匙從樂音訊號中萃取出藏在其中的數位浮水印。

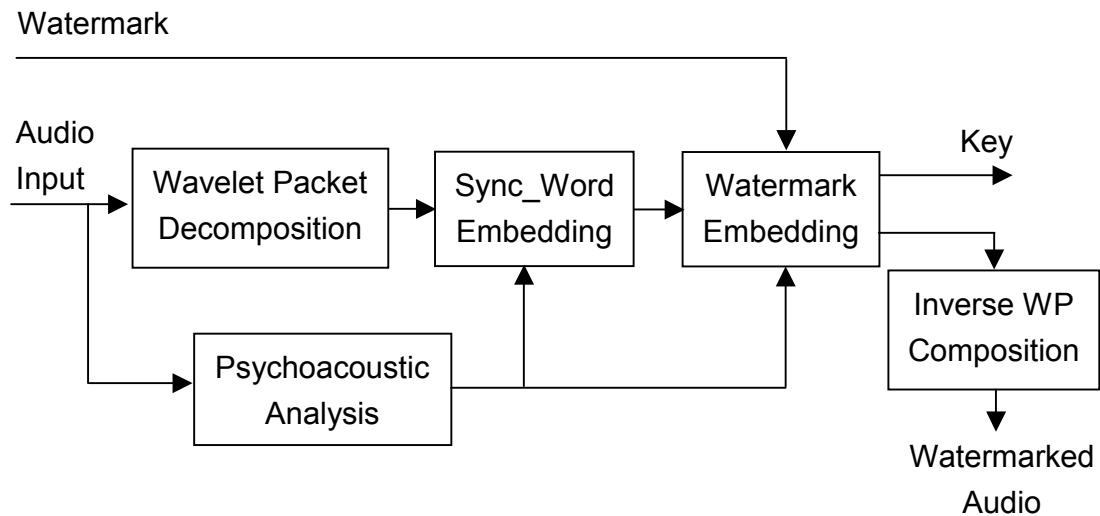


圖 4.1 數位音訊浮水印系統嵌入架構圖

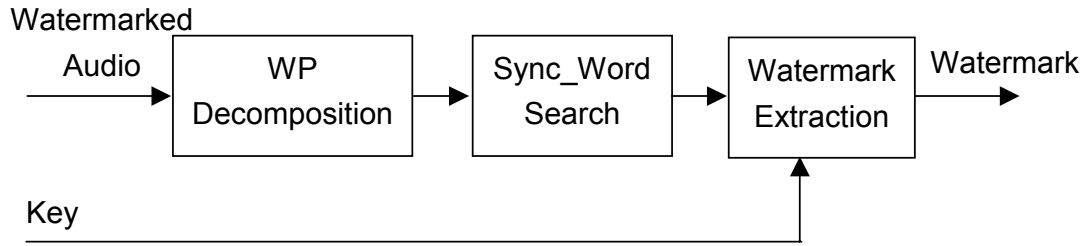


圖 4.2 數位音訊浮水印系統萃取架構圖

在 4.1 節中將會介紹嵌入浮水印的方法；4.2 節說明萃取浮水印的方法；4.3 節將介紹嵌入同步碼的方法；4.4 節說明如何萃取及搜尋同步碼。

4.1 數位浮水印嵌入架構

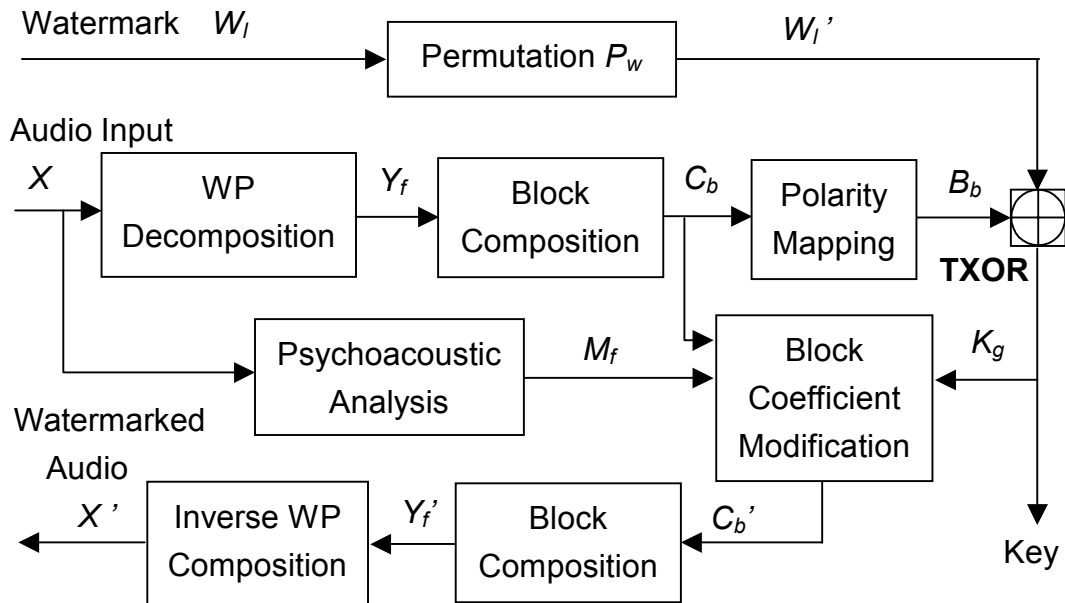


圖 4.3 數位音訊浮水印嵌入架構流程圖

如圖 4.3 為本論文所提出的小波封包分頻配合人耳聲學模型的金

匙音訊浮水印嵌入架構流程圖，以下各小節將詳細說明此架構的處理流程。

4.1.1 小波次頻帶選擇

如同 3.2 節小波分頻架構所述，將樂音訊號分割成 1024 個取樣點的音框後，再對音框做小波封包處理，共 8 階 29 個次頻帶的小波封包結構。以人類聽覺感官而言，人耳對於中頻帶訊號是較為敏感的，以最小能聽見度 (ATH) 來看 (詳見 3.3 節)，在 500 Hz ~ 10 kHz 聽覺較為敏感，尤以 2 ~ 5 kHz 之間聽覺最為敏銳。所以中頻帶係數並不適合用來隱藏數位浮水印。而高頻帶訊號雖然具有較低的敏感性，將數位浮水印藏在較高頻的次頻帶，將對音質的影響會較低，但是高頻帶的係數值通常較小，且經過大多數的音訊壓縮技術 (如: MP3、AC-3、AAC) 壓縮後，高頻帶係數可能會被捨去，而在萃取端造成數位浮水印嚴重失真的現象。所以我們選擇嵌入數位浮水印的小波係數為較低頻次頻帶的係數。以小波封包分頻架構來看，選擇第 1 次頻帶至第 15 次頻帶之間的係數，如圖 4.4 中虛線方框所涵蓋的範圍，並將其分為兩個頻段，次頻帶 1 ~ 11 為頻段 I，次頻帶 12 ~ 15 為頻段 II，如圖 4.4 所示。以頻率範圍配合最小能聽見度來看，選擇的頻率範圍約為 0 ~ 2.75 kHz 左右，其中頻段 I 的頻率範圍約為 0 ~ 1.375 kHz 左右，頻段 II 的頻率範圍約為 1.375 ~ 2.75 kHz，如圖 4.5 所示。

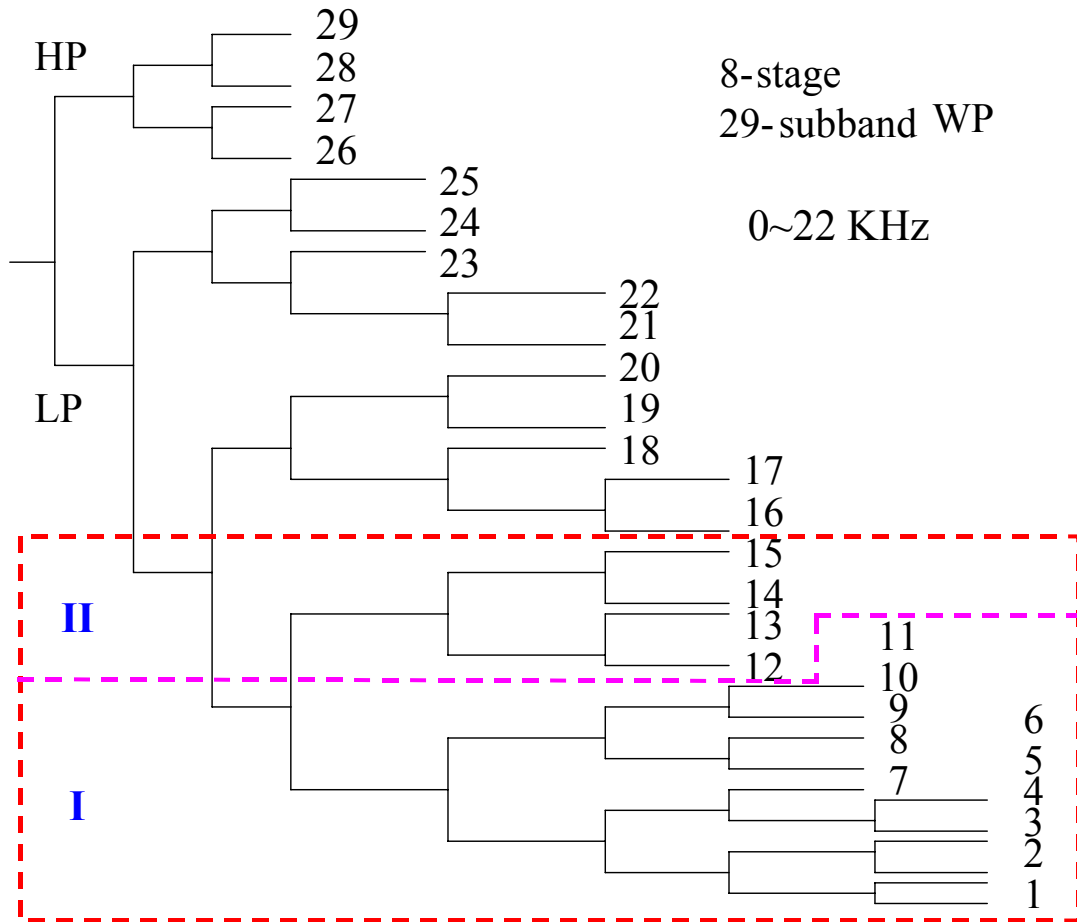


圖 4.4 嵌入數位浮水印的小波次頻帶

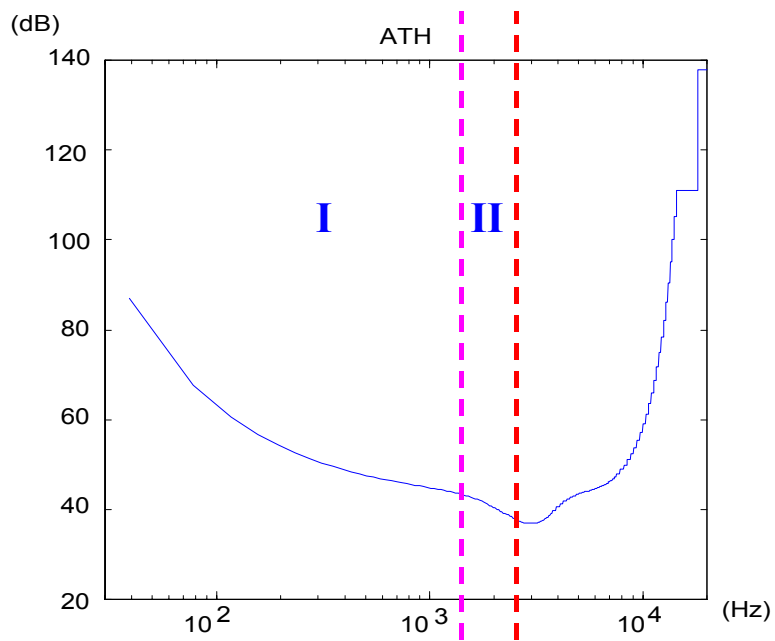


圖 4.5 嵌入數位浮水印的頻帶範圍

4.1.2 區塊分類

以圖 4.4 所示的小波封包分類架構，每一音框（1024 samples）的樂音訊號會產生 29 個次頻帶的小波係數，共 1024 個小波係數點；其中由上一小節中所選擇的小波次頻帶範圍內，次頻帶 1 ~ 6 各包含 4 個小波係數，次頻帶 7 ~ 11 各包含 8 個小波係數，次頻帶 12 ~ 15 各包含 16 個小波係數。每一音框皆會有 128 個小波係數點被選取。

將此 128 個小波係數點依序分為每 4 個係數為一組的區塊 C_b ， b 為區塊序列號碼， $b = 1, 2, \dots, 32$ ， C_b^i 為區塊內的係數， i 為係數編號， $i = 1, 2, 3, 4$ 。把 128 個小波係數分為 32 個區塊後，將依序對區塊的極性作定義與判別。

區塊極性分類

首先定義一個計數係數 N_b ， b 為區塊序列號碼， N_b 主要用來統計區塊中係數大於 0 的個數， N_b 的值介於 0 ~ 4，如圖 4.6 中的例子所示。

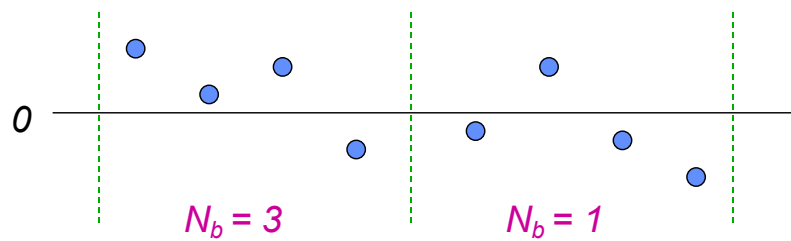


圖 4.6 次頻帶係數區塊分類

另外定義區塊極性 B_b ， b 為區塊序列號碼，依區塊內係數大於 0 的個數將區塊極性分為三類 ”0”、”1”、”U (Undefined)”。由於每個區塊位於不同的訊號頻帶，所以對於不同頻帶內的區塊其極性分類方式也不同，其分類法則如下：

(A) 位於頻段 I 內的區塊

A. If $N_b = 0、1$, $B_b = 0$ 。B. If $N_b = 3、4$, $B_b = 1$ 。C. If $N_b = 2$, $B_b = U$ 。

(B) 位於頻段 II 內的區塊

A. If $N_b = 0$, $B_b = 0$ 。B. If $N_b = 4$, $B_b = 1$ 。C. If $N_b = 1、2、3$, $B_b = U$ 。

對於頻段 I 內的區塊，由於其位於較低頻的次頻帶，在經過音訊壓縮或其他類型的攻擊時，區塊內係數比較不會有太大的變化，所以若區塊內的 4 個係數中有超過一半的係數大於 0，則此區塊的極性 B_b 偏向 ”正”，因此就將區塊的極性 B_b 定義為 ”1”；若區塊內的 4 個係數中有超過一半的係數小於 0，則此區塊的極性 B_b 偏向 ”負”，因此就將區塊的極性 B_b 定義為 ”0”；而當區塊內 4 個的係數中有一半大於 0 一半小於 0 時，此類區塊在受到音訊壓縮等攻擊後，很容易會改變它的極性，所以我們將這類型的區塊極性 B_b 定義為 “U”，代表此類區塊不適合用來嵌入數位浮水印。

而對於頻段 II 內的區塊，由於其位於偏中頻的次頻帶，在經過音訊壓縮或其他類型的攻擊時，區塊內係數容易會有大幅度的變化，所以我們採用較為嚴苛的條件來為其區塊分類，以確保數位浮水印的強健性。只有當區塊內 4 個係數都大於 0 時，才將區塊極性 B_b 定義為 “1”；只有當區塊內 4 個係數都小於 0 時，才將區塊極性 B_b 定義為 “0”；其餘的所有情況都將區塊極性 B_b 定義為 “U”。

4.1.3 安全鑰匙的產生

在將數位浮水印嵌入樂音訊號前，會先對數位浮水印做一些前置的處理。我們把一張 $M \times M$ 大小的數位二階浮水印影像視為一連串由 “1”、“0” 二元資料所組成的序列 W_l ， l 為數位二階浮水印的序列號碼， $l = 1, 2, \dots, M \times M$ 。為了安全起見，會將數位二階浮水印 W_l 經過一個重排 (Permutaton) 的程序而成為 W_l' 。

將重排後的數位二階浮水印序列 W_l' 根據上一小節所描述的區塊極性，選擇性的依序嵌入區塊中，一個區塊只嵌入一位元的資料。嵌入數位浮水印時，我們選擇將數位二階浮水印 “1” 嵌入極性為 “0” 的區塊；將數位二階浮水印 “0” 嵌入極性為 “1” 的區塊，如此在數位浮水印的萃取端便可利用區塊的極性，將嵌入的數位浮水印正確的萃取出來。而我們選擇略過極性為 “U” 的區塊，因為這類型的區塊容易受到音訊處理的影響而改變其極性，所以並不適合用來嵌入數位浮水印。而安全鑰匙 K_g 則是用來指示該區塊是否用來嵌入數位浮水印，當該區塊用來嵌入數位浮水印時，便將其對應的安全鑰匙 K_g 設為 “1”，反之則設為 “0”。

我們在這裡定義一 tri-state XOR (TXOR) 的運算，其運算真值表如表 4.1。上述數位浮水印的嵌入步驟，簡單的說，就是將數位二階浮水印 W_l' 和區塊極性 B_b 做 TXOR 的運算，其運算法則如式 (4.1) 所示。

$$K_g = \begin{cases} W_l' \oplus B_b' & , B_b' = 0, 1 \\ 0 & , B_b' = U \end{cases} \dots\dots\dots (4.1)$$

\oplus : XOR

運算結果將產生一串二元的的安全鑰匙 K_g ，而當運算結果 $K_g = 0$ 的時候，該數位浮水印位元 W_l' 將保留下來與下一個區塊極性 B_b 做 TXOR 運算，直至運算結果 $K_g = 1$ 時，才會進入下一個浮水印位元。而此安全鑰匙只有數位浮水印嵌入者才會擁有，如此便能對數位浮水印的安全性多一層的保護，且在數位浮水印萃取端只需利用此安全鑰匙，便可將數位浮水印萃取出來，並不需要原始的樂音訊號。詳細的 TXOR 運算流程圖，如圖 4.7 所示。

表 4.1 TXOR 運算真值表

W_l'	B_b	K_g
1	0	1
1	1	0
0	0	0
0	1	1
1	U	0
0	U	0

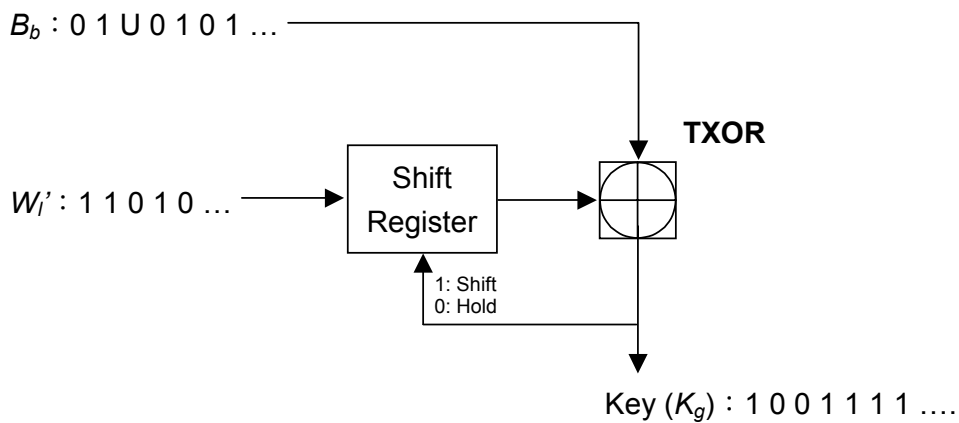


圖 4.7 TXOR 運算產生安全鑰匙的例子

假設 $W'_l[5]: 11010$, $B_b[7]: 01U0101$

● W'_l TXOR B_b :

$$K_g[0] = W'_l[0] \boxplus B_b[0] = 1 \boxplus 0 = 1。$$

$$K_g[1] = W'_l[1] \boxplus B_b[1] = 1 \boxplus 1 = 0, W'_l[1] \text{ Hold。}$$

$$K_g[2] = W'_l[1] \boxplus B_b[2] = 1 \boxplus U = 0, W'_l[1] \text{ Hold。}$$

$$K_g[3] = W'_l[1] \boxplus B_b[3] = 1 \boxplus 0 = 1。$$

$$K_g[4] = W'_l[2] \boxplus B_b[4] = 0 \boxplus 1 = 1。$$

$$K_g[5] = W'_l[3] \boxplus B_b[5] = 1 \boxplus 0 = 1。$$

$$K_g[6] = W'_l[4] \boxplus B_b[6] = 0 \boxplus 1 = 1。$$

4.1.4 區塊極性強化

經過 4.1.3 節所述的 TXOR 的運算之後，我們得到一串二元的安
全鑰匙以及其相對應欲嵌入數位浮水印的區塊。而為了加強數位浮水
印的強健性，我們必須對區塊內的係數 C_b^i 做些修改，並且要考慮到
人耳聽覺系統對修改這些係數的敏感度，所以必須根據人耳聲學模型
計算出的最小遮蔽臨界值來更動這些係數。

詳細方法為：根據安全鑰匙 $K_g = 1$ 時，將其對應到的區塊內係數
值落在介於 0 到門檻值 T 之間的係數，依其區塊極性 B_b ，正向或逆
向的修改這些係數值。而門檻值 T 的選取是根據人耳聲學模型計算出
的最小遮蔽臨界值 M_f 乘上一強度因數 S_f (Scale Factor) 得到。如式
(4.2)。

$$T = S_f * M_f \dots\dots\dots(4.2)$$

區塊係數修改依其區塊極性分為兩類：

(A) $K_g = 1$ 且 $B_b = 1$

If $0 \leq C_b^i \leq T \Rightarrow C_b^i = T$ ，如圖 4.8(a)。

(B) $K_g = 1$ 且 $B_b = 0$

If $0 \geq C_b^i \geq -T \Rightarrow C_b^i = -T$ ，如圖 4.8(b)。

最後將修改後的小波係數依其所屬次頻帶，經小波封包合成濾波器組還原成時域訊號，即為藏有數位二階浮水印的樂音訊號。

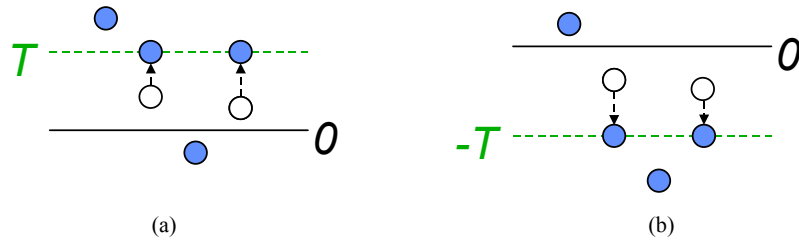


圖 4.8 區塊係數修改示意圖

(a) 區塊極性為 1

(b) 區塊極性為 0

4.2 數位浮水印萃取架構

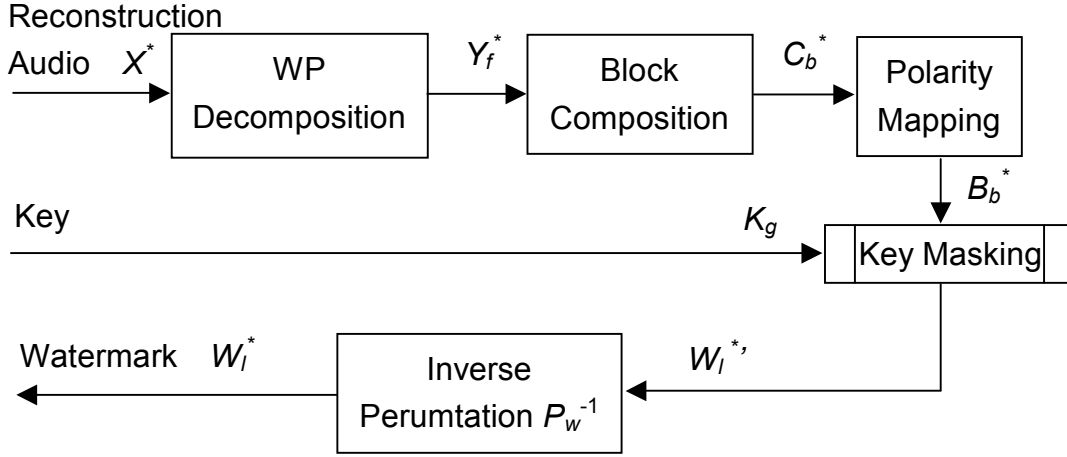


圖 4.9 數位音訊浮水印萃取架構流程圖

數位浮水印的萃取架構流程如圖 4.9 所示。嵌入數位浮水印的樂音訊號經常會因為傳輸系統或儲存媒體的需要，必須經過音訊壓縮以降低位元率，或是惡意使用者欲破壞數位浮水印，而對樂音訊號做濾波、裁切、加雜訊等音訊處理的攻擊。我們通稱這些經過數位音訊處理的樂音訊號為重建音訊 X^* (Reconstruction Audio)，對於重建音訊 X^* 的數位浮水印萃取過程，將如下列步驟所述：

小波分頻及區塊分類

如同 4.1.1 及 4.1.2 節的敘述，對於樂音訊號先分割成 1024 個取樣點的音框後，再對音框做小波封包處理，然後將小波係數分成 4 個係數一組的區塊，再依序對區塊做極性的判別，在萃取端的極性判別方式只採用一種判別方法：

A. If $N_b^* = 0、1$, $B_b^* = 0$ 。

B. If $N_b^* = 3、4$, $B_b^* = 1$ 。

C. If $N_b^* = 2$, $B_b^* = U$ 。

安全鑰匙還原數位浮水印

我們將安全鑰匙 K_g 視為一個遮罩 (Mask)，將區塊極性 B_b^* 通過這個遮罩，也就是利用安全鑰匙 K_g 將有嵌入數位浮水印的區塊挑選出來，並利用下列三個判別法則將數位浮水印還原回來：

(A) $K_g = 1$ 且 $B_b^* = 1$

數位浮水印 $W_l^{*'} = 0$ 。

(B) $K_g = 1$ 且 $B_b^* = 0$

數位浮水印 $W_l^{*'} = 1$ 。

(C) $K_g = 0$

該區塊並未嵌入數位浮水印。

詳細的流程如圖 4.10 所示。依此流程還原回來的即是經過重排程序的數位二階浮水印 $W_l^{*'}$ 。最後只要再將 $W_l^{*'}$ 做一反重排 (Inverse Permutation) 的程序，即可還原為 W_l^* 。

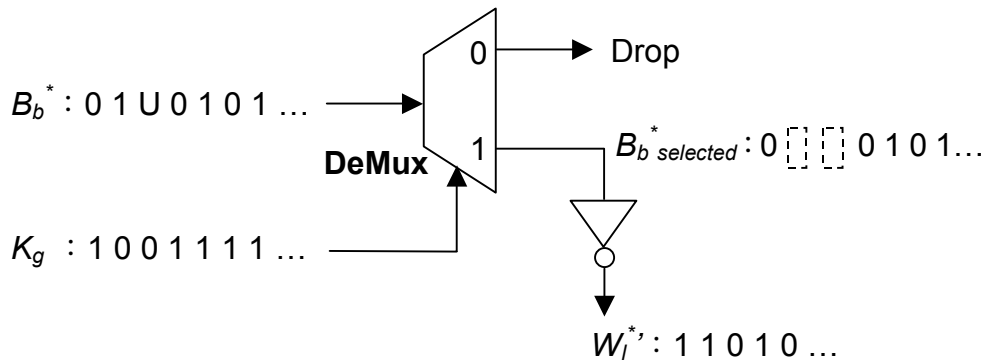


圖 4.10 由重建音訊萃取數位二階浮水印的例子

4.3 同步碼嵌入架構

在數位音訊浮水印的應用中，一個具強健性的數位浮水印除了要能抵抗音訊壓縮、濾波、加雜訊等常見的攻擊外，對於數位音訊處理之後常會產生的音訊延遲 (Delay)，以及裁切或從樂音中擷取出其中一小段等攻擊，也必須要能夠克服。音訊延遲或裁切攻擊對於數位音訊浮水印的影響，是會導致音框平移或無法確切得知音框分割邊界位置等效應，而造成音框起點無法對正 (Alignment) 的問題；音框無法對正，則數位浮水印無法萃取或萃取出來的數位浮水印會嚴重失真。對於裁切或擷取這類簡單有效又不會嚴重影響音質的攻擊，必須採用其他的方法來克服。目前有一些已知的技術可以用來克服此類問題[20][21]，不同的方法配合不同的音訊浮水印技術可以有效解決同步的問題，但是這些方法的缺點是無法精確的搜尋到樂音訊號的同步點，大約會有正負 10 個取樣點[20]及正負 50 個取樣點[21]的誤差，而這樣的效能還不足以達到本論文提出之數位音訊浮水印系統的要求 (詳見 5.4 節)，所以無法應用在我們的系統中。因此我們在此提出一同步碼嵌入技術，可以有效的搜尋對正音框的起始點，使數位浮水印可以正確的萃取出來，以增加數位音訊浮水印技術應用的範圍。

結合同步碼與數位浮水印的技術，嵌入樂音訊號的資料將是一組【同步碼 | 數位浮水印】的二元資料，若將此組資料重複的嵌入樂音訊號中，不論是對樂音訊號做裁切，或是擷取樂音訊號其中的某一段，只要受攻擊後的樂音訊號包含一組完整的【同步碼 | 數位浮水印】的二元資料，就可以在搜尋到同步碼後將數位浮水印萃取出來。圖 4.11 與圖 4.12 為同步碼與數位浮水印配合的範例。

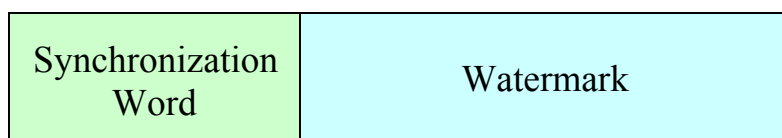


圖 4.11 一組【同步碼 | 數位浮水印】資料

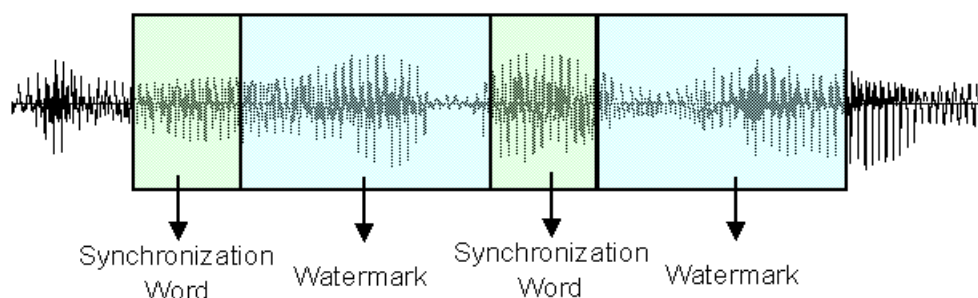


圖 4.12 同步碼與數位浮水印嵌入樂音訊號範例

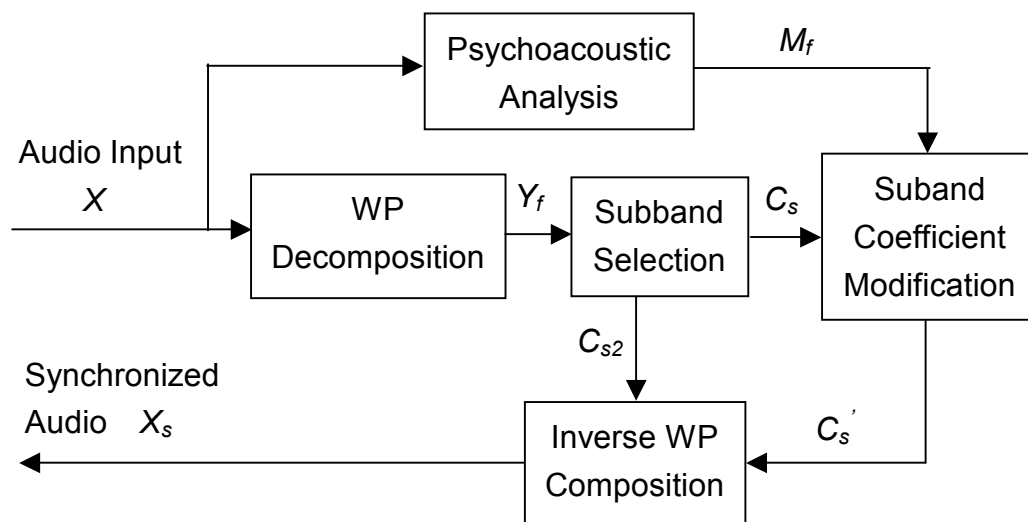


圖 4.13 同步碼嵌入架構流程圖

如圖 4.13 為本論文所提出的同步碼嵌入架構流程圖，以下將詳細說明此架構的處理流程。

小波次頻帶選擇

如同 4.1 節所提出的數位浮水印嵌入架構，同步碼同樣也是嵌入在小波係數中。將樂音訊號分割成音框及小波封包處理成 29 個次頻帶係數後，選擇其中 3 個次頻帶的係數作為嵌入同步碼之用。因為同步碼必須要對取樣點位移敏感，若選擇較低頻次頻帶係數，由於低頻次頻帶係數點較少，對時間的解析度不夠，無法對取樣點位移敏感，而且低頻係數經過如高通濾波器容易失真，不適合用來嵌入同步碼；而若選擇較高頻次頻帶係數，雖然次頻帶係數點較多，但是對音訊攻擊的抵抗力較差，同步碼容易被破壞。所以在此我們選擇如圖 4.4 中位於頻段 II 中的第 12、13、14 次頻帶來嵌入同步碼。

次頻帶係數絕對平均值

在每個音框中，位於頻段 II 中的第 12、13、14 次頻帶，每個次頻帶包含有 16 個係數點 C_s^i ， s 表示係數位於哪個次頻帶， $s = 12、13、14$ ， i 表示次頻帶內係數編號， $i = 1、2、\dots、16$ 。首先先將次頻帶內係數取絕對值為 $|C_s^i|$ ，接著再求此 16 個 $|C_s^i|$ 的平均值 E_s ，並且令 E_{max} 等於這 3 個次頻帶中最大的 E_s ， E_{mid} 等於次大的 E_s ， E_{min} 等於最小的 E_s 。

令欲嵌入的同步碼為 S_l ， l 表示同步碼的長度，而嵌入同步碼的法則，就是利用 E_{max} 、 E_{mid} 、 E_{min} 三者之間的差異，來表示藏入其中的二元資料，如下列步驟：

$$(1) \text{ 令 } E_a = E_{max} - E_{mid} \dots\dots\dots(4.3)$$

$$\text{令 } E_b = E_{mid} - E_{min} \dots\dots\dots(4.4)$$

$$(2) \text{ 若 } S_l = 1$$

$$\text{使 } E_a - E_b \geq T_s \dots\dots\dots(4.5)$$

(3) 若 $S_l = 0$

$$\text{使 } E_b - E_a \geq T_s \dots\dots\dots(4.6)$$

其中門檻值 T_s 的選取必須考慮到人耳聲學模型計算出的最小遮蔽臨界值，由於同步碼的嵌入涵蓋了三個次頻帶的係數，因此決定門檻值 T_s 時，選擇此三個次頻帶中最小的那一個最小遮蔽臨界值 M_{f_min} ，再乘上一強度因數 S_{f_min} 而得到，如式 (4.7)。

$$T_s = S_{f_min} * M_{f_min} \dots\dots\dots(4.7)$$

藉由上述三個步驟，每個音框可以嵌入 1 個 bit 的同步碼，若同步碼的長度 $l = 16$ ，則連續 16 個音框就可以嵌入一個完整的同步碼。然而同步碼的嵌入是以式 (4.5) 及式 (4.6) 來判斷，若原始樂音訊號的次頻帶係數即滿足如此的關係式，則不必更動次頻帶係數；若否，則必須修改次頻帶係數直至滿足式 (4.5) 或式 (4.6) 為止。

強化或抑制次頻帶係數

當次頻帶係數不滿足式 (4.4) 或式 (4.5) 時，必須要修改次頻帶的係數，修改的方式是藉由漸漸增大或減小次頻帶中的係數去更動 E_{max} 、 E_{mid} 、 E_{min} 這三個值的大小，進而去修正 E_a 及 E_b 之間的關係。詳細方法如下列步驟：

(A) 當 $S_l = 1$

(1) If $E_a - E_b \geq T_s$, goto (3)

Else goto (2)

(2) Amplify E_{max} and E_{min} by G_a , Attenuate E_{mid} by G_c , goto (1)

(3) End

(B) 當 $S_l = 0$

(1) If $E_b - E_a \geq T_s$, goto (3)

Else goto (2)

(2) Amplify E_{mid} by G_a , Attenuate E_{max} and E_{min} by G_c , goto (1)

(3) End

以上更動 E_s 的方法如圖 4.14 所示。增大或減小次頻帶係數的方法，就是藉由將次頻帶係數乘上一個大於 1 的常數 G_a 或小於 1 的常數 G_c ，慢慢的修正該次頻帶的 E_s ，直至滿足條件為止。然而為了避免改變係數將對於音框邊緣造成不連續性的影響，增大係數時是將 C_s^i 乘上一個長度為 16 的 Extended-Hanning window (視窗的前 1/4 和後 1/4 組成一個完整的 Hanning window)，稱為強化視窗 (Enhancement Window)，如圖 4.15。減小係數時是將 C_s^i 乘上一個長度為 16 的收縮視窗 (Contraction Window)，如圖 4.16。

最後再將修改後的小波係數依其所屬次頻帶，經小波封包合成濾波器組還原成時域訊號，即為含有同步碼之樂音訊號。

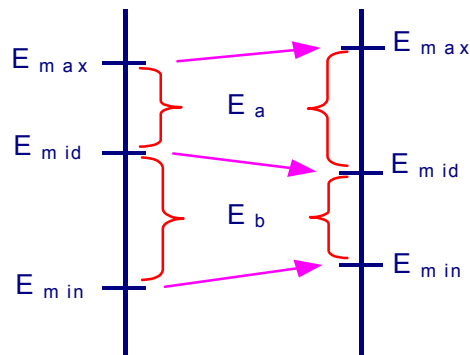


圖 4.14 嵌入 $S_l = 1$ 時更動 E_s 示意圖

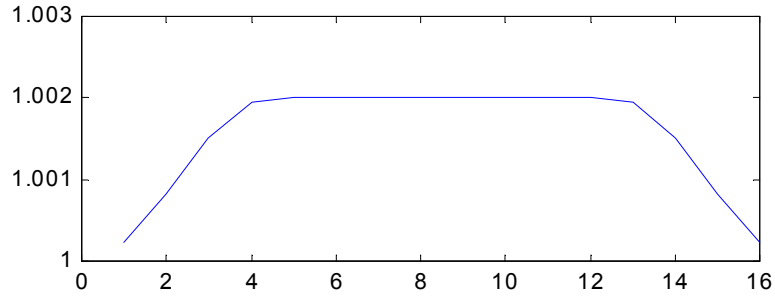


圖 4.15 強化視窗

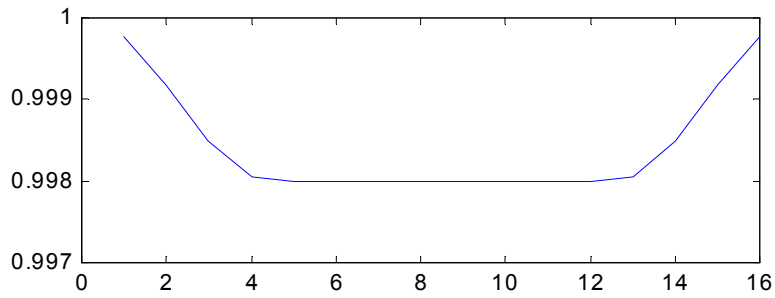


圖 4.16 抑制視窗

4.4 同步碼搜尋萃取架構

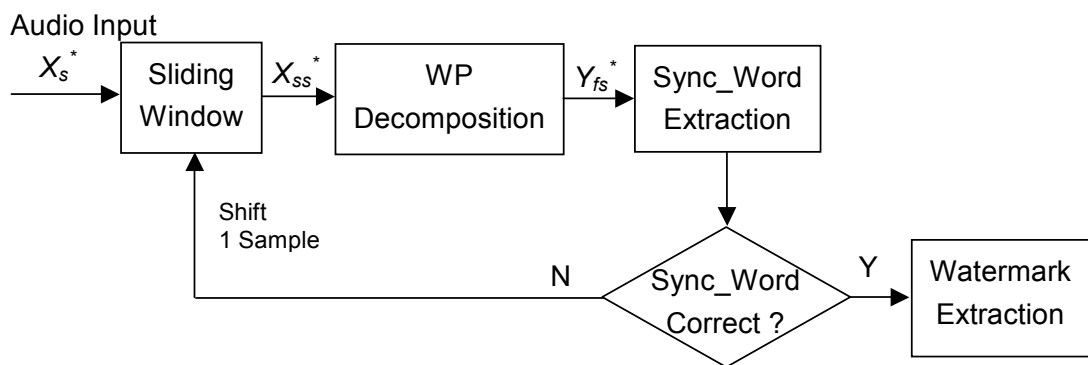


圖 4.17 同步碼搜尋萃取架構流程圖

如圖 4.17 為搜尋及萃取同步碼的架構流程圖，經過音訊延遲或裁切、擷取等攻擊的重建訊號 X_s^* ，我們已無法判斷其音框分割點位

置，所以在萃取數位浮水印之前，必須先透過同步碼的搜尋萃取程序，將音框的起點對正。

由於我們無法判斷重建訊號的音框起點位於何處，所以必須一次位移一個取樣點來搜尋同步碼，首先將重建訊號 X_s^* 輸入一個長度為 L 的滑動視窗 (Sliding Window)，視窗長度 L 的單位為音框個數，選擇 L 等於一個完整同步碼嵌入的音框長度，滑動視窗初始位移取樣點數為 0，經過滑動視窗選取出 L 個音框長度的重建訊號後，將其經過小波封包處理成 29 個次頻帶係數，挑選出位於頻段 II 中的第 12、13、14 次頻帶係數 C_{ss}^* ，用來偵測同步碼是否存在。

萃取同步碼的前置處理如同嵌入同步碼時一樣，首先將次頻帶內係數取絕對值為 $|C_{ss}^{i*}|$ 後，再求此 16 個 $|C_{ss}^{i*}|$ 的平均值 E_{ss}^* ，令 E_{max} 等於這 3 個次頻帶中最大的 E_{ss}^* ， E_{mid} 等於次大的 E_{ss}^* ， E_{min} 等於最小的 E_{ss}^* ，再令 $E_a = E_{max} - E_{mid}$ ， $E_b = E_{mid} - E_{min}$ 。

做完前置處理後，只要再利用 E_a 及 E_b 的關係，即可將同步碼 S_l^* 萃取出來，如下列步驟所述：

(A) 若 $E_a - E_b > 0$

則同步碼 $S_l^* = 1$ 。

(B) 若 $E_b - E_a > 0$

則同步碼 $S_l^* = 0$ 。

若同步碼的長度 $l = 16$ ，則連續處理 16 個音框後就可以得到一組 16 個位元的資料，將此萃取出的二元資料與同步碼做比對，若與同步碼不相同，則滑動視窗向右滑動一個取樣點，再重複上述的搜尋萃取步驟。當萃取出的二元資料與同步碼相同時，表示已經搜尋到同步點，音框的起點已經對正，如此再利用 4.2 節所述的數位浮水印萃取方式，即可順利的將數位二階浮水印萃取出來。

第五章 實驗結果與討論

本論文的實驗模擬環境，是以單聲道 (Monophonic)、CD 品質 (取樣頻率 44.1 kHz，每取樣點以 16 位元表示) 之樂音來作為數位浮水印嵌入的對象。而樂音包含各種樂器。模擬環境主要以 C 程式語言為主，在個人電腦 PC Pentium III 800 進行樂音之小波分析、合成技術和數位浮水印的嵌入、萃取動作。並以 MatLab、Cool-Editor 等資料處理分析軟體進行樂音之分析與播放。測試的樂音有：大提琴 (cello)，合唱曲 (chorus)，長笛 (flute)，吉他 (guitar)，大鍵琴 (harpsichord)，號角 (horn)，魯特琴 (lute)，樂團 (orchestra)，管風琴 (organ)，鋼琴 (piano)，喇叭 (trumpet)，小提琴 (violin)。

5.1 浮水印容量與音質評估

5.1.1 浮水印容量評估

首先我們先測試本論文之音訊浮水印技術所能提供的浮水印容量 (Capacity)，也就是能提供給數位二階浮水印嵌入的位元數，測試的方法為將隨機二位元亂數 ”1”、”0” 嵌入樂音訊號中，直至樂音訊號所提供的可嵌入空間用完為止，統計所嵌入的位元數。每段樂音的長度為 200 個音框，共 204800 個取樣點。

我們將測試兩種浮水印嵌入方法，稱為 Level 1 及 Level 2，兩種方法的差別在於嵌入浮水印的頻帶範圍，如 4.1.1 節所述的小波次頻帶選擇，共分為頻段 I 及頻段 II。而 Level 1 及 Level 2 的分別在於：

- Level 1：只有頻段 I 內的小波次頻帶係數做為數位浮水印嵌入之用。

- Level 2：頻段 I 及頻段 II 內的小波次頻帶係數皆作為數位浮水印嵌入之用。

表 5.1 為實驗數據結果，表 5.2 為 12 種樂器的統計數值，結果顯示在 12 種測試樂音中，Level 1 的方法平均每個音框可以嵌入 4.7 個位元，Level 2 的方法平均每個音框可以嵌入 5.9 個位元。Level 1 及 Level 2 浮水印容量的差異如圖 5.1 所示。

表 5.1 平均每個音框可嵌入的位元數 (Bits/Frame)

樂器名	cello	chorus	flute	guitar	harpsichord	horn
Level 1	5.171717	4.717172	4.540404	4.292929	4.535354	4.272727
Level 2	6.550505	6.207071	5.59596	5.40404	5.959596	5.863636
樂器名	lute	orchestra	organ	piano	trumpet	violin
Level 1	5.005051	5.020202	4.792929	4.939394	4.893939	4.085859
Level 2	5.515152	6.368687	5.828283	6.136364	6.530303	5.439394

表 5.2 統計 12 種樂器可嵌入位元數的統計值 (Bits/Frame)

統計值	Average	Max	Min
Level 1	4.688973	5.171717	4.085859
Level 2	5.949916	6.550505	5.40404

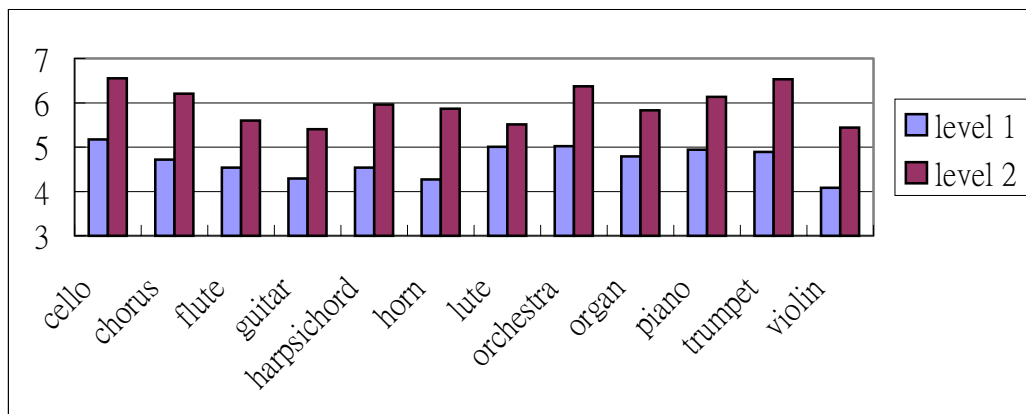


圖 5.1 Level 1 及 Level 2 浮水印容量 (Bits/Frame)

5.1.2 樂音音質評估

在樂音品質的比較上，分別以主觀方法和客觀方法來做比較，主觀方式是以聽覺感受來做比較，而客觀方法以 SSNR (Segment Signal to Noise Ratio) 來做比較。

由 5.1.1 節可知 Level 1 及 Level 2 的浮水印容量不同，因此我們採用兩個不同大小的數位二階浮水印分別用於 Level 1 及 Level 2 的方法。Level 1 嵌入 28 x 28 個像素點的浮水印，如圖 5.2 (a)，Level 2 嵌入 32 x 32 個像素點的浮水印，如圖 5.2 (b)。



(a)



(b)

圖 5.2 數位二階浮水印 (a) Level 1 (28 x 28) (b) Level 2 (32 x 32)

客觀音質評估

我們分別以 Level 1 及 Level 2 的方法將上述數位二階浮水印嵌入樂音訊號中，並且對嵌入浮水印後的樂音訊號以客觀方法 SSNR 來做音質的評估，實驗結果如表 5.3 及表 5.4。結果顯示 Level 1 及 Level 2 的樂音音質 SSNR 平均都有 40 dB 以上，而 Level 1 及 Level 2 樂音音質的差異如圖 5.3 所示，在本實驗中 Level 2 由於嵌入的數位二階浮水印較大，資料量較多，因此 Level 2 的 SSNR 會比 Level 1 稍差，但差別不會太大，約為 1dB 左右。

表 5.3 嵌入浮水印之樂音品質 (dB)

樂器名	cello	chorus	flute	guitar	harpsichord	horn
level 1	47.2851	44.80901	38.69367	43.7297	42.64123	42.75827
level 2	46.63657	43.6281	37.31249	41.16777	42.40325	43.58095
樂器名	lute	orchestra	organ	piano	trumpet	violin
level 1	45.77076	45.63687	48.32864	47.7714	39.97727	44.67887
level 2	45.14637	44.181	46.26271	45.66009	37.45922	43.84455

表 5.4 統計 12 種樂器音質的統計值 (dB)

統計值	Average	Max	Min
level 1	44.34007	48.32864	38.69367
level 2	43.10692	46.63657	37.31249

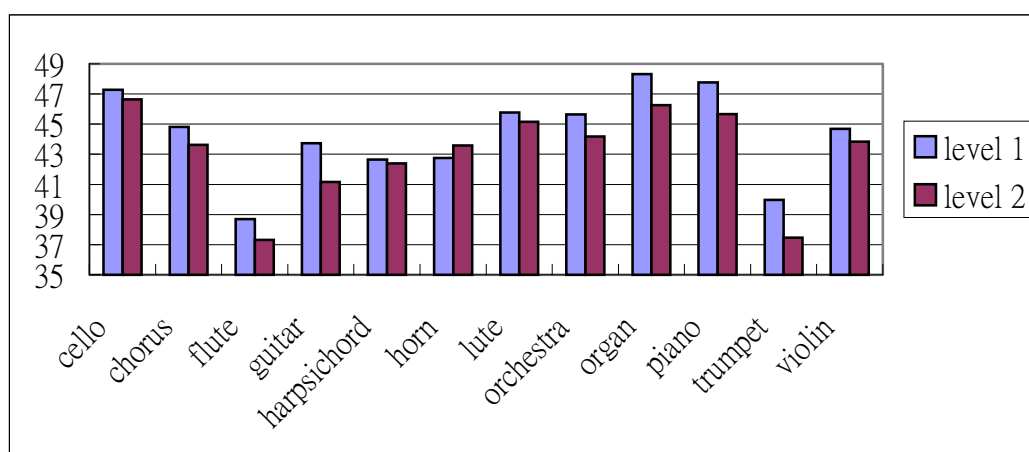


圖 5.3 Level 1 及 Level 2 樂音音質 (dB)

主觀音質評估

在主觀品質評估方面，評估方法採用 Preference Ratio 作為評估的標準，我們請到 20 位受測者，測試方式為每種測試樂音播放兩次，其中一次為原始樂音，另一次為嵌入浮水印的樂音，在不告知播放次序的情況下，受測者選擇其認為音質較佳的樂音，測試結果如表 5.5 及表 5.6。

表 5.5 Level 1 聽覺喜好比例值

Instrument	Original	Watermarked
Cello	47.5 %	52.5 %
Chorus	60.0 %	40.0 %
Flute	47.5 %	52.5 %
Guitar	45.0 %	55.0 %
Harpsichord	52.5 %	47.5 %
Horn	50.0 %	50.0 %
Lute	47.5 %	52.5 %
Orchestra	52.5 %	47.5 %
Organ	52.5 %	47.5 %
Piano	47.5 %	52.5 %
Trumpet	37.5 %	62.5 %
Violin	50.0 %	50.0 %

表 5.6 Level 2 聽覺喜好比例值

Instrument	Original	Watermarked
Cello	45.0 %	55.5 %
Chorus	62.5 %	37.5 %
Flute	55.0 %	45.0 %
Guitar	52.5 %	47.5 %
Harpsichord	52.5 %	47.5 %
Horn	47.5 %	52.5 %
Lute	50.0 %	50.0 %
Orchestra	65.0 %	35.0 %
Organ	60.0 %	40.0 %
Piano	37.5 %	62.5 %
Trumpet	45.0 %	55.0 %
Violin	47.5 %	52.5 %

由測試結果可以看出加入浮水印的樂音的 Preference Ratio 在 50%左右，也就是說本論文的音訊浮水印系統並不會影響樂音訊號在聽覺上的品質。

5.2 音訊壓縮測試

在評估浮水印技術的強健性時，我們採用常用的音訊壓縮編碼技術 MPEG Audio Layer-3 來對嵌入浮水印後的樂音訊號做壓縮攻擊，分別測試 32 ~ 256 Kbps 不同壓縮位元率時，浮水印技術對 MP3 重建訊號萃取浮水印的能力。而強健性的評估方式是採用 Normalized Correlation (NC) 來作為測量的標準，如式 (5.1)。

$$\text{Normalized Correlation (NC)} = \frac{\sum_i \sum_j X(i, j) X^*(i, j)}{\sum_i \sum_j [X(i, j)]^2} \dots\dots\dots (5.1)$$

x：原始浮水印

x*：代表經音訊處理後擷取出的浮水印

NC 值會介於-1 ~ 1 之間，越接近 1 表示數位浮水印的相似度越高，越接近 0 表示數位浮水印被破壞的越嚴重，而-1 表示萃取出的浮水印與原始的浮水印完全相反，相當於 BER = 1 的情況。表 5.7 及表 5.8 為不同壓縮位元率下 Level 1 及 Level 2 方法的 NC 值。圖 5.4 及圖 5.5 分別為 Level 1 及 Level 2 方法其壓縮位元率與萃取出的數位浮水印 NC 值（統計值）的關係。圖 5.7 及圖 5.8 為列舉三種樂音訊號在不同壓縮位元率下所萃取出來的數位二階浮水印。

表 5.7 不同壓縮位元率下 Level 1 方法的 NC 值

	32 Kbps	48 Kbps	64 Kbps	96 Kbps	128 Kbps	256 Kbps
cello	0.831633	0.969388	0.994898	1	1	1
chorus	0.808673	0.969388	1	1	1	1
flute	0.905612	0.977041	1	1	1	1
guitar	0.767857	0.964286	0.987245	0.997449	1	1
harpsichord	0.704082	0.915816	0.982143	1	1	1
horn	0.923469	0.984694	1	1	1	1
lute	0.788265	0.931122	0.989796	1	1	1
orchestra	0.783163	0.959184	0.989796	1	1	1
organ	0.857143	0.987245	1	1	1	1
piano	0.92602	0.979592	0.994898	1	1	1
trumpet	0.971939	0.969388	0.989796	1	1	1
violin	0.831633	0.903061	0.954082	1	1	1
Average	0.841624	0.959184	0.990221	0.999787	1	1
Max	0.971939	0.987245	1	1	1	1
Min	0.704082	0.903061	0.954082	0.997449	1	1

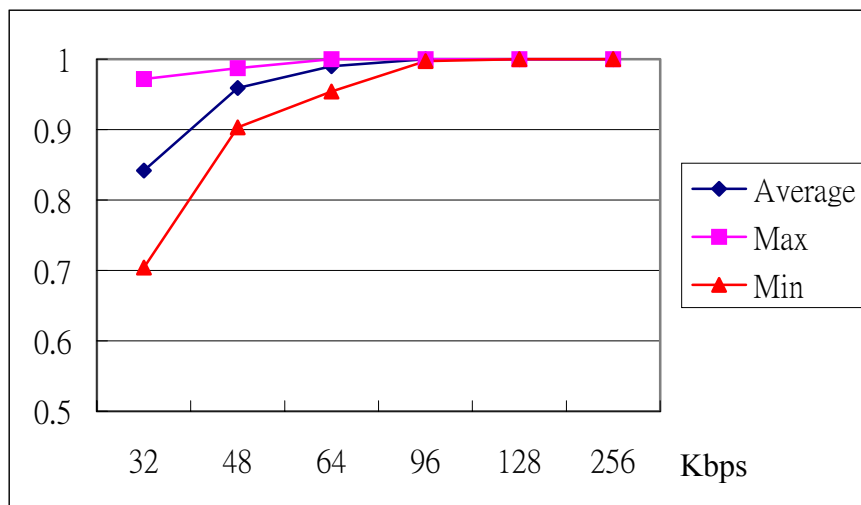


圖 5.4 方法 Level 1 中壓縮位元率與 NC 值的關係

表 5.8 不同壓縮位元率下 Level 2 方法的 NC 值

	32 Kbps	48 Kbps	64 Kbps	96 Kbps	128 Kbps	256 Kbps
cello	0.787109	0.947266	0.996094	1	1	1
chorus	0.744141	0.955078	0.996094	1	1	1
flute	0.822266	0.958984	0.990234	1	1	1
guitar	0.695313	0.917969	0.980469	0.998047	1	1
harpsichord	0.691406	0.919922	0.986328	0.998047	1	1
horn	0.902344	0.996094	0.998047	1	1	1
lute	0.771484	0.884766	0.974609	1	1	1
orchestra	0.757813	0.957031	0.992188	1	1	1
organ	0.818359	0.978516	1	1	1	1
piano	0.927734	0.992188	1	1	1	1
trumpet	0.951172	0.978516	0.988281	1	1	1
violin	0.832031	0.953125	0.982422	1	1	1
Average	0.808431	0.953288	0.990397	0.999675	1	1
Max	0.951172	0.996094	1	1	1	1
Min	0.691406	0.884766	0.974609	0.998047	1	1

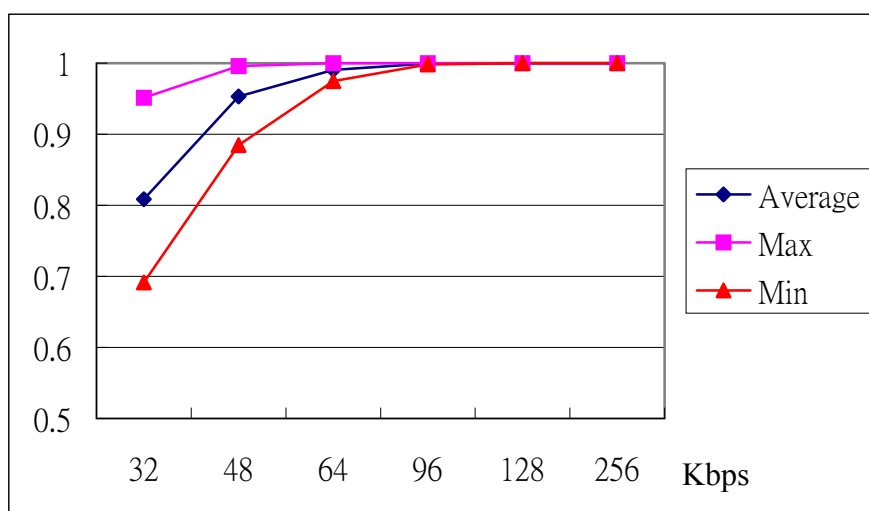


圖 5.5 方法 Level 2 中壓縮位元率與 NC 值的關係

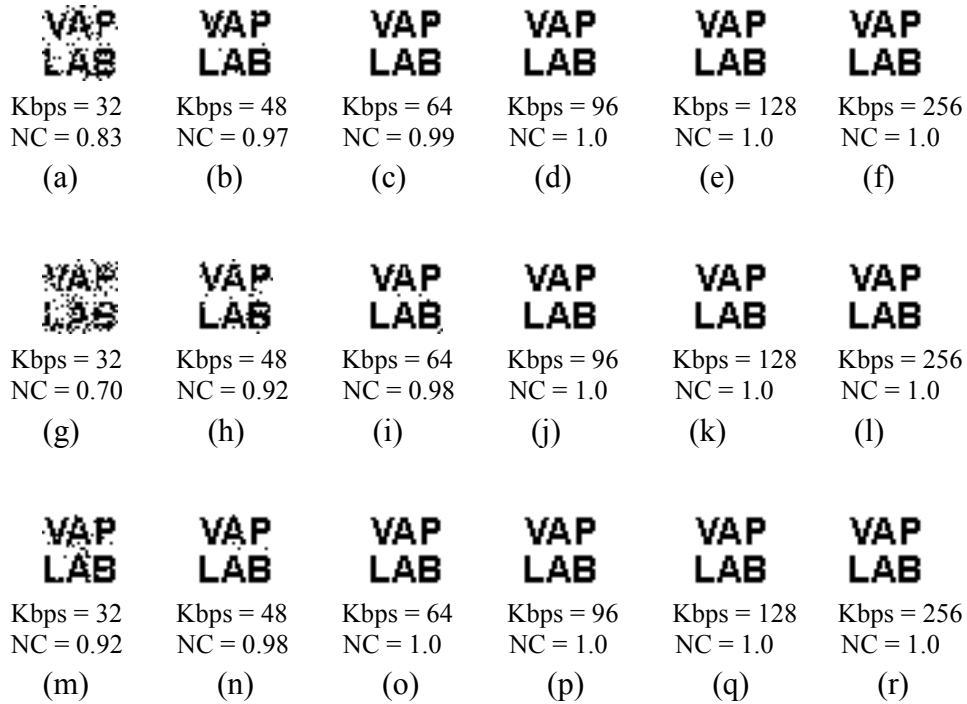


圖 5.6 方法 Level 1 中，在不同壓縮位元率下所萃取出之浮水印

(a)~(f)測試樂音為 cello, (g)~(l)測試樂音為 harpsichord, (m)~(r)測試樂音為 horn

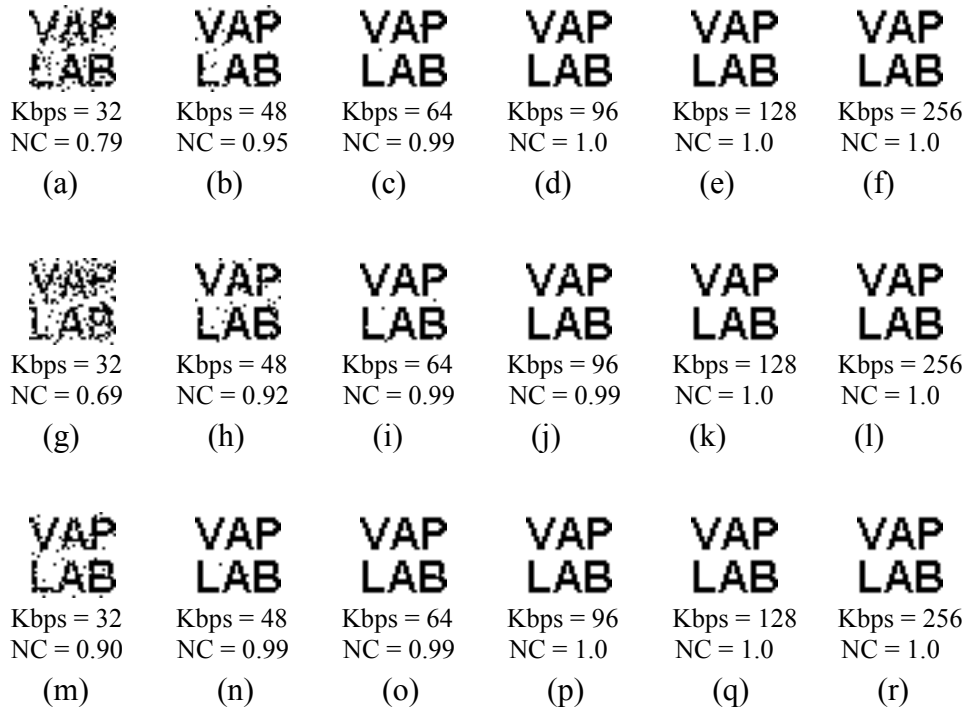


圖 5.7 方法 Level 2 中，在不同壓縮位元率下所萃取出之浮水印

(a)~(f)測試樂音為 cello, (g)~(l)測試樂音為 harpsichord, (m)~(r)測試樂音為 horn

由表 5.7、表 5.8 以及圖 5.4 ~ 圖 5.7，可以看出數位浮水印對於 MP3 壓縮的強健性，壓縮位元率在 48 Kbps 以上時，萃取出數位二階浮水印 NC 值大多都在 0.9 以上，與原始嵌入的浮水印非常相近。而 Level 1 與 Level 2 的差別在於，Level 1 由於只嵌入位於頻段 I 內的小波次頻帶係數，能嵌入的浮水印位元數較少，而由於其在頻率分布上屬於低頻的部份，所以較不容易被音訊壓縮攻擊所破壞，所以 NC 值較高。而 Level 2 除了嵌入頻段 I 的小波係數外，也會嵌入頻段 II 的小波係數，所以能提供的浮水印容量較高，但由於頻段 II 涵蓋的頻率範圍比頻段 I 在頻率上較高，所以 Level 2 對音訊壓縮攻擊的強健性比 Level 1 稍微差了一點點。

5.3 Stirmark for Audio 測試

接下來的測試是採用 2.3 節中介紹的音訊浮水印標準評定程序 Stirmark for Audio 來做音訊浮水印的強健性測試，使用的版本是 v0.2 版[12]，其中包含的所有攻擊方式將列於附錄 A 中。由於此標準評定程序目前尚在發展階段，所以程式中還有一些臭蟲 (Bugs) 存在，某些樂音訊號無法做測試，而且某些種類的攻擊方式也有一些問題存在，我們將排除這些有問題的測試結果。所以在此階段的測試樂音只使用了大提琴 (cello)，長笛 (flute)，大鍵琴 (harpsichord)，號角 (horn)，魯特琴 (lute)，樂團 (orchestra)，管風琴 (organ)，鋼琴 (piano)，喇叭 (trumpet)這九種樂音訊號。

在不考慮 Stirmark for Audio 的攻擊對樂音訊號音質影響的情況下，先列出數位浮水印對所有種類攻擊的強健性測試結果 (其中不包含 voice_remove、add_fft_noise、resampling、extra_stereo 等不適用或有問題的攻擊種類)，對音質的影響將在之後討論。Stirmark 的所有參數均為初使設定值並未更改，表 5.9 為 Level 1 的測試結果，表 5.10 為 Level 2 的測試結果，圖 5.8、圖 5.9 為 Level 1 中九種測試樂音的統計結果，圖 5.10、圖 5.11 為 Level 1 中九種測試樂音的統計結果。

表 5.9 Stirmark for Audio 對 Level 1 方法的 NC 值

	cello	flute	harpichord	horn	lute	orches2	organ	piano	trumpet	Average	Max	Min
rc_highpass	0.762755	0.670918	0.642857	0.57398	0.683673	0.729592	0.755102	0.696429	0.721939	0.693027	0.762755	0.57398
rc_lowpass	0.997449	1	0.994898	0.977041	1	0.997449	0.997449	1	0.997449	0.995748	1	0.977041
addnoise_100	0.956633	0.961735	0.982143	0.846939	0.880102	0.987245	0.997449	0.97449	0.959184	0.949547	0.997449	0.846939
addnoise_300	0.905612	0.918367	0.918367	0.767857	0.658163	0.931122	0.987245	0.905612	0.65051	0.849206	0.987245	0.65051
addnoise_500	0.857143	0.877551	0.82398	0.755102	0.609694	0.928571	0.977041	0.882653	0.602041	0.812642	0.977041	0.602041
addnoise_700	0.82398	0.834184	0.711735	0.734694	0.545918	0.910714	0.979592	0.859694	0.548469	0.772109	0.979592	0.545918
addnoise_900	0.798469	0.82398	0.642857	0.727041	0.487245	0.90051	0.977041	0.852041	0.505102	0.746032	0.977041	0.487245
addbrumm_100	0.984694	0.964286	0.997449	0.931122	0.961735	0.992347	0.992347	0.982143	0.943878	0.972222	0.997449	0.931122
addbrumm_1100	0.966837	0.895408	0.880102	0.852041	0.862245	0.959184	0.951531	0.915816	0.829082	0.901361	0.966837	0.829082
addbrumm_2100	0.954082	0.890306	0.816327	0.834184	0.869898	0.943878	0.956633	0.895408	0.808673	0.885488	0.956633	0.808673
addbrumm_3100	0.94898	0.872449	0.839286	0.813776	0.836735	0.946429	0.931122	0.908163	0.811224	0.878685	0.94898	0.811224
addbrumm_4100	0.951531	0.877551	0.808673	0.813776	0.844388	0.928571	0.94898	0.903061	0.803571	0.875567	0.951531	0.803571
addbrumm_5100	0.946429	0.887755	0.803571	0.831633	0.841837	0.938776	0.928571	0.90051	0.818878	0.877551	0.946429	0.803571
addbrumm_6100	0.938776	0.877551	0.795918	0.831633	0.862245	0.941327	0.943878	0.892857	0.788265	0.874717	0.943878	0.788265
addbrumm_7100	0.923469	0.869898	0.803571	0.829082	0.844388	0.918367	0.933673	0.895408	0.77551	0.86593	0.933673	0.77551
addbrumm_8100	0.918367	0.844388	0.798469	0.82398	0.846939	0.918367	0.92602	0.895408	0.811224	0.864796	0.92602	0.798469
addbrumm_9100	0.92602	0.880102	0.793367	0.821429	0.836735	0.915816	0.920918	0.885204	0.77551	0.861678	0.92602	0.77551
addbrumm_10100	0.908163	0.869898	0.783163	0.816327	0.818878	0.897959	0.923469	0.885204	0.790816	0.854875	0.923469	0.783163
normalize	1	1	1	1	1	1	1	1	1	1	1	1
amplify	1	1	1	1	1	1	1	1	1	1	1	1
addsinus	0.589286	0.57398	0.665816	0.596939	0.428571	0.854592	0.742347	0.813776	0.502551	0.640873	0.854592	0.428571
compressor	1	1	1	1	1	1	1	1	1	1	1	1
cutsamples	0.002551	-0.08929	0.206633	0.005102	0.09949	0.015306	0.071429	0.102041	0.196429	0.067744	0.206633	-0.08929
zerocross	0.408163	0.316327	0.556122	0.479592	0.334184	0.992347	0.854592	0.903061	0.367347	0.579082	0.992347	0.316327
zerolength	-0.01786	0.048469	0.117347	0.020408	0.140306	0.017857	0.005102	0.102041	0.15051	0.064909	0.15051	-0.01786
zeroremove	-0.09694	0.07398	0.084184	0.076531	0.112245	0.045918	-0.09184	-0.0051	0.214286	0.045918	0.214286	-0.09694
fft_hlpass	0.783163	0.826531	0.808673	0.788265	0.772959	0.831633	0.895408	0.836735	0.82398	0.818594	0.895408	0.772959
flipsample	0.673469	0.602041	0.614796	0.497449	0.553571	0.579082	0.576531	0.579082	0.670918	0.594104	0.673469	0.497449
copysample	0.040816	-0.01786	0.135204	0.119898	0.094388	0.022959	-0.05102	0	0.188776	0.05924	0.188776	-0.05102
stat1	0.997449	0.992347	0.984694	0.97449	1	0.997449	0.997449	1	0.994898	0.993197	1	0.97449
stat2	1	1	1	1	1	1	1	1	1	1	1	1
fft_stat1	0.492347	0.57398	0.589286	0.446429	0.535714	0.512755	0.392857	0.528061	0.538265	0.512188	0.589286	0.392857
echo	0.576531	0.543367	0.596939	0.410714	0.584184	0.459184	0.428571	0.505102	0.619898	0.524943	0.619898	0.410714
fft_test	0.459184	0.561224	0.423469	0.382653	0.456633	0.522959	0.382653	0.484694	0.403061	0.452948	0.561224	0.382653
exchange	1	1	1	1	1	1	1	1	1	1	1	1
smooth	1	1	1	1	1	1	1	1	1	1	1	1
smooth2	1	1	0.992347	1	1	1	1	1	1	0.99915	1	0.992347
lsbzero	1	1	1	1	1	1	1	1	1	1	1	1
dynnoise	0.951531	0.941327	0.959184	0.793367	0.864796	0.915816	0.97449	0.90051	0.946429	0.916383	0.97449	0.793367
fft_real_reverse	1	1	1	1	1	1	1	1	1	1	1	1
invert	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
fft_invert	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

表 5.10 Stirmark for Audio 對 Level 2 方法的 NC 值

	cello	flute	harpichord	horn	lute	orches2	organ	piano	trumpet	Average	Max	Min
rc_highpass	0.810547	0.730469	0.712891	0.71875	0.6875	0.785156	0.800781	0.78125	0.792969	0.757813	0.810547	0.6875
rc_lowpass	0.998047	1	0.992188	0.982422	1	0.998047	0.998047	0.998047	0.994141	0.99566	1	0.982422
addnoise_100	0.958984	0.960938	0.994141	0.908203	0.896484	0.996094	0.994141	0.980469	0.958984	0.960938	0.996094	0.896484
addnoise_300	0.923828	0.888672	0.957031	0.857422	0.654297	0.958984	0.988281	0.912109	0.730469	0.874566	0.988281	0.654297
addnoise_500	0.875	0.837891	0.861328	0.806641	0.583984	0.9375	0.980469	0.888672	0.693359	0.829427	0.980469	0.583984
addnoise_700	0.837891	0.800781	0.75	0.783203	0.498047	0.9375	0.972656	0.871094	0.650391	0.789063	0.972656	0.498047
addnoise_900	0.791016	0.763672	0.703125	0.783203	0.486328	0.925781	0.96875	0.871094	0.583984	0.764106	0.96875	0.486328
addbrumm_100	0.986328	0.976563	0.996094	0.960938	0.962891	0.988281	0.986328	0.990234	0.947266	0.977214	0.996094	0.947266
addbrumm_1100	0.958984	0.929688	0.894531	0.898438	0.884766	0.966797	0.96875	0.935547	0.853516	0.921224	0.96875	0.853516
addbrumm_2100	0.958984	0.916016	0.857422	0.900391	0.878906	0.970703	0.964844	0.923828	0.828125	0.911024	0.970703	0.828125
addbrumm_3100	0.955078	0.914063	0.84375	0.888672	0.871094	0.957031	0.951172	0.921875	0.824219	0.902995	0.957031	0.824219
addbrumm_4100	0.947266	0.902344	0.839844	0.894531	0.871094	0.949219	0.953125	0.898438	0.837891	0.899306	0.953125	0.837891
addbrumm_5100	0.929688	0.896484	0.833984	0.892578	0.873047	0.945313	0.949219	0.904297	0.826172	0.894531	0.949219	0.826172
addbrumm_6100	0.933594	0.896484	0.826172	0.902344	0.863281	0.957031	0.960938	0.902344	0.820313	0.895833	0.960938	0.820313
addbrumm_7100	0.929688	0.900391	0.84375	0.902344	0.871094	0.9375	0.931641	0.904297	0.826172	0.894097	0.9375	0.826172
addbrumm_8100	0.929688	0.898438	0.820313	0.892578	0.863281	0.943359	0.933594	0.898438	0.822266	0.889106	0.943359	0.820313
addbrumm_9100	0.921875	0.894531	0.822266	0.894531	0.853516	0.933594	0.923828	0.896484	0.828125	0.885417	0.933594	0.822266
addbrumm_10100	0.923828	0.898438	0.826172	0.882813	0.865234	0.931641	0.941406	0.90625	0.832031	0.889757	0.941406	0.826172
normalize	1	1	1	1	1	1	1	1	1	1	1	1
amplify	1	1	1	1	1	1	1	1	1	1	1	1
addsinus	0.652344	0.605469	0.707031	0.660156	0.478516	0.898438	0.775391	0.841797	0.632813	0.694662	0.898438	0.478516
compressor	1	1	1	1	1	1	1	1	1	1	1	1
cutsamples	0.019531	-0.05078	0.115234	-0.05664	0.103516	0.011719	0.041016	0.019531	0.164063	0.040799	0.164063	-0.05664
zerocross	0.382813	0.353516	0.517578	0.480469	0.349609	0.998047	0.830078	0.876953	0.408203	0.577474	0.998047	0.349609
zerolength	-0.00586	0.035156	0.115234	0.029297	0.103516	0.070313	0.023438	0.068359	0.103516	0.06033	0.115234	-0.00586
zeroremove	-0.01367	0.023438	0.085938	0.164063	0.072266	0.054688	-0.09375	0.023438	0.080078	0.044054	0.164063	-0.09375
fft_hlpass	0.835938	0.867188	0.794922	0.878906	0.800781	0.859375	0.912109	0.873047	0.880859	0.855903	0.912109	0.794922
flipsample	0.626953	0.554688	0.605469	0.583984	0.613281	0.539063	0.572266	0.603516	0.650391	0.594401	0.650391	0.539063
copysample	0.035156	-0.04883	0.097656	0.087891	0.066406	0.052734	0	0.021484	0.132813	0.049479	0.132813	-0.04883
stat1	1	0.996094	0.992188	0.974609	0.998047	0.998047	0.996094	1	0.996094	0.994575	1	0.974609
stat2	1	1	1	1	1	1	1	1	1	1	1	1
fft_stat1	0.484375	0.546875	0.552734	0.509766	0.533203	0.552734	0.457031	0.521484	0.566406	0.524956	0.566406	0.457031
echo	0.566406	0.541016	0.597656	0.503906	0.601563	0.523438	0.433594	0.556641	0.642578	0.551866	0.642578	0.433594
fft_test	0.488281	0.558594	0.447266	0.501953	0.421875	0.550781	0.472656	0.503906	0.449219	0.488281	0.558594	0.421875
exchange	1	1	1	1	1	1	1	1	1	1	1	1
smooth	1	1	1	1	1	1	1	1	1	1	1	1
smooth2	1	1	0.990234	1	1	1	0.996094	1	1	0.998481	1	0.990234
lsbzero	1	1	1	1	1	1	1	1	1	1	1	1
dynnoise	0.951172	0.947266	0.976563	0.876953	0.861328	0.9375	0.984375	0.912109	0.966797	0.934896	0.984375	0.861328
fft_real_reverse	1	1	1	1	0.998047	1	1	1	1	0.999783	1	0.998047
invert	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
fft_invert	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

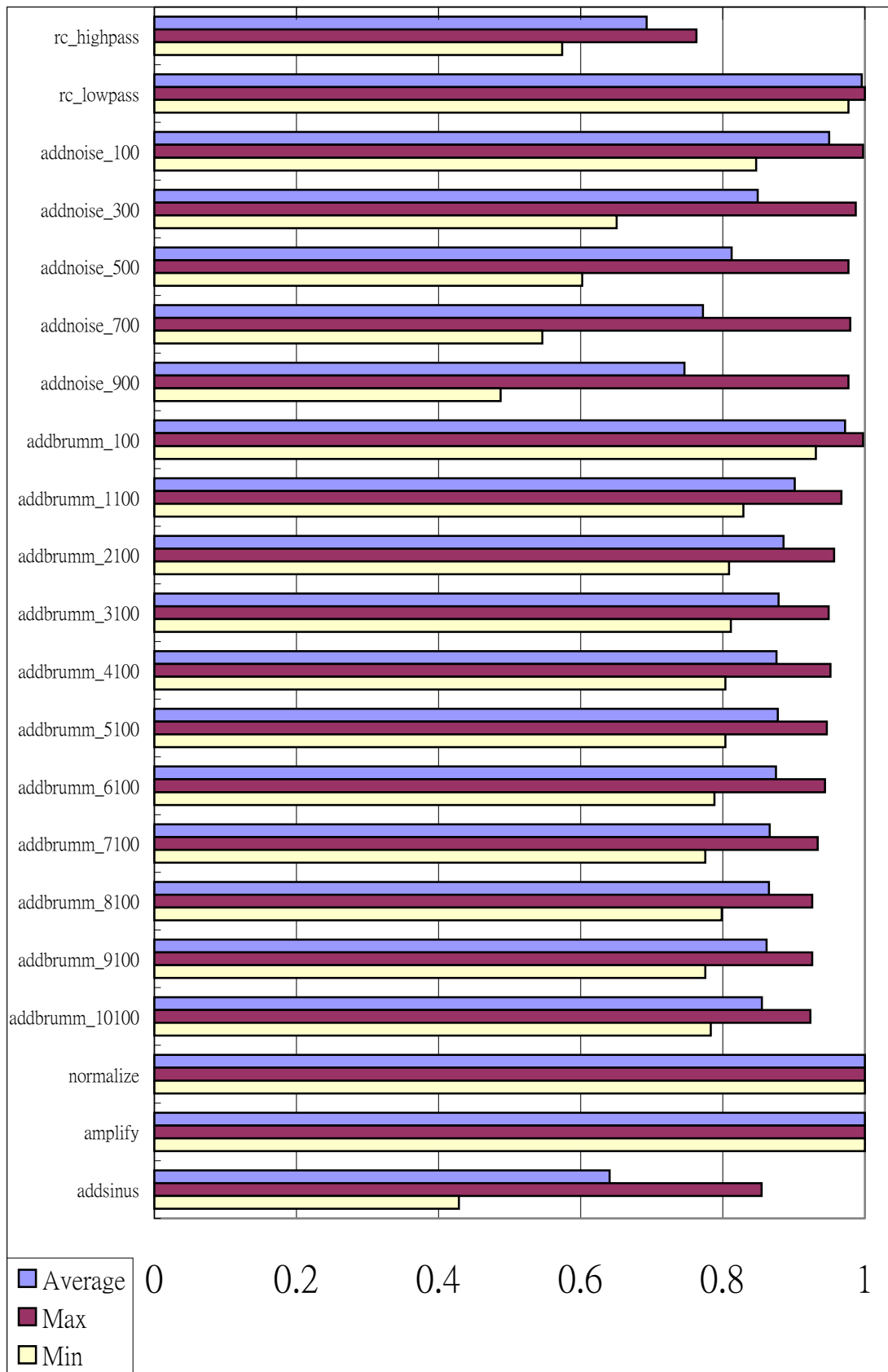


圖 5.8 方法 Level 1 中不同音訊攻擊下所萃取出之浮水印 NC 值 (1)

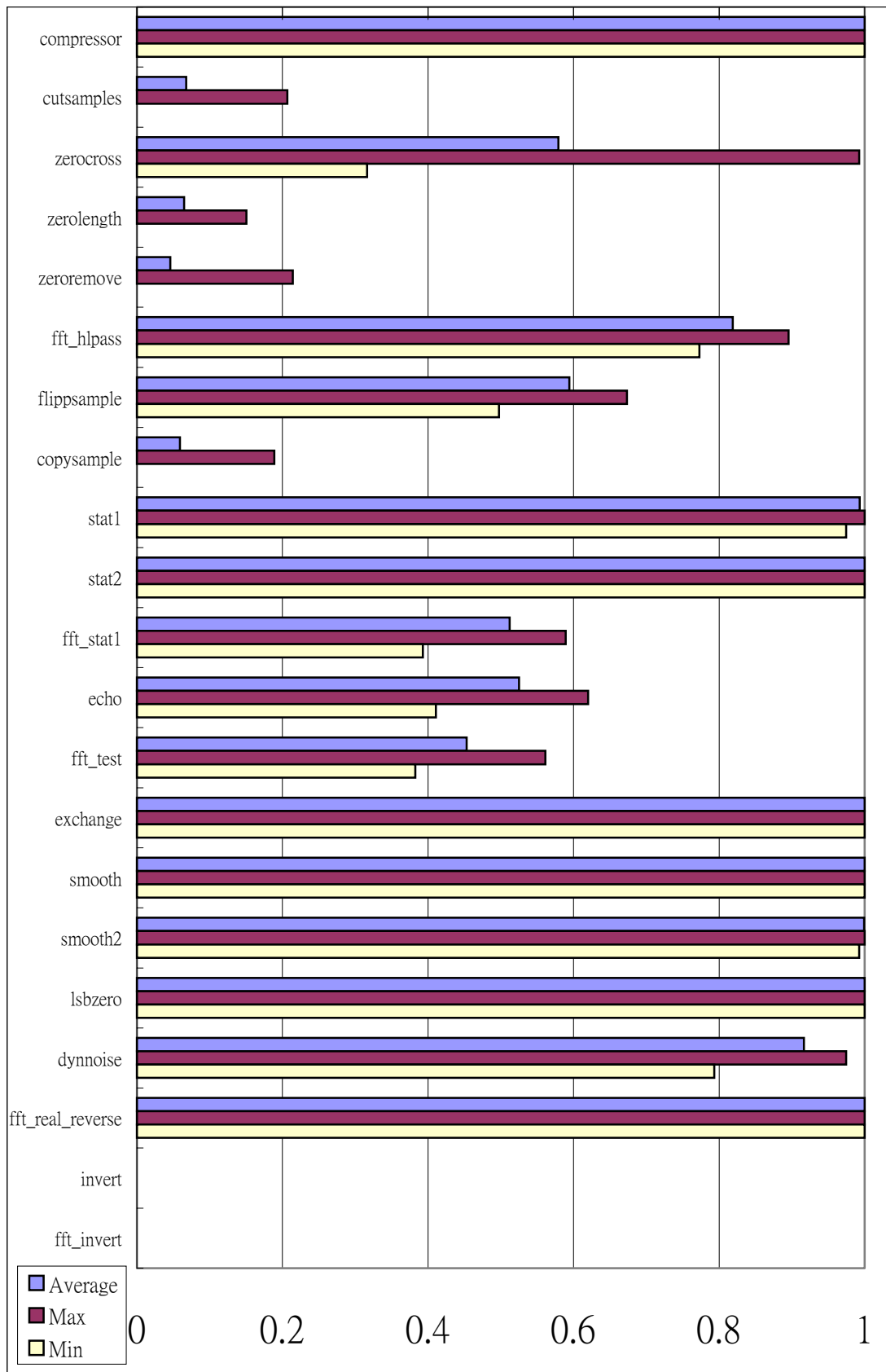


圖 5.9 方法 Level 1 中不同音訊攻擊下所萃取出之浮水印 NC 值 (2)

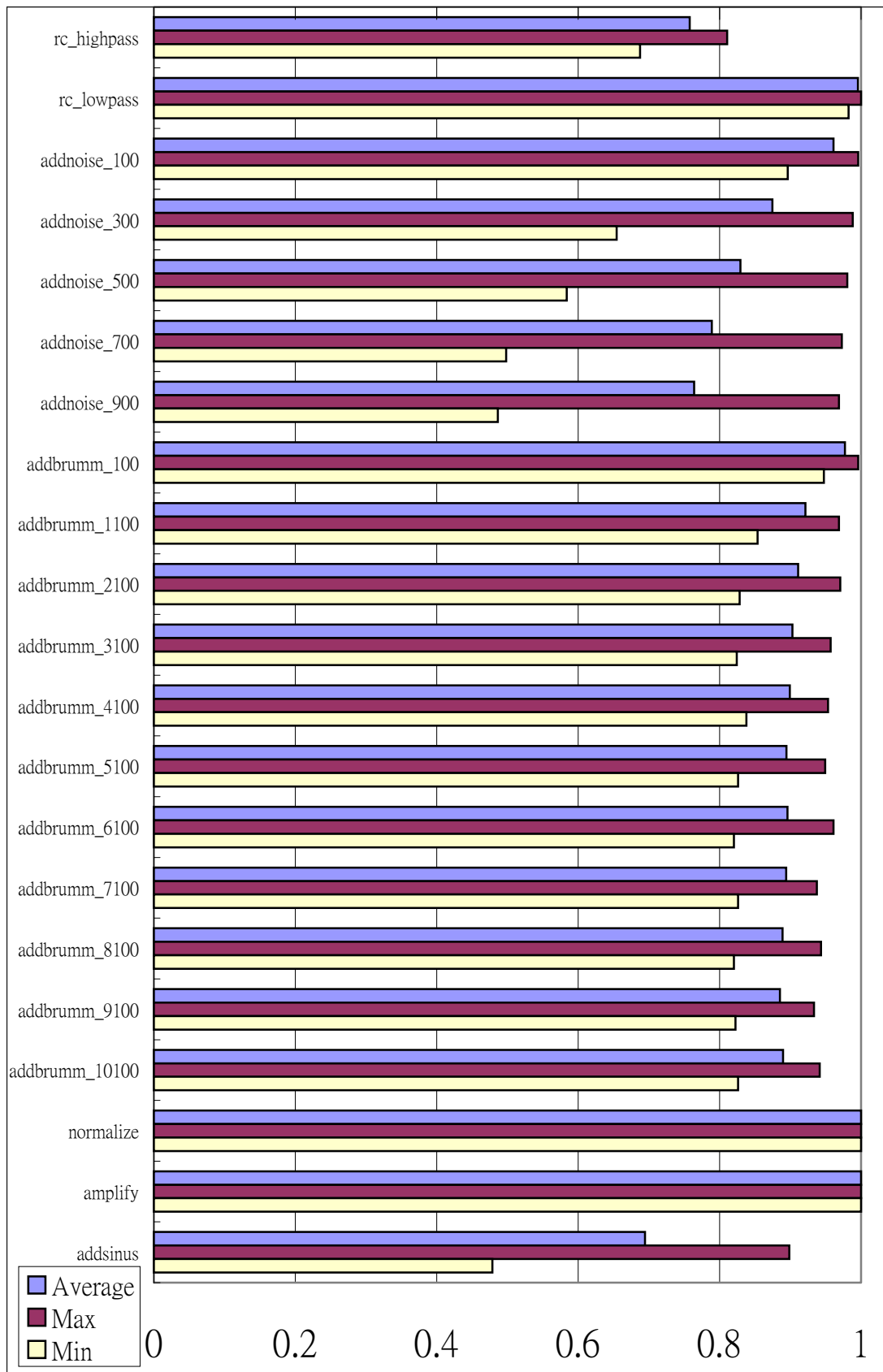


圖 5.10 方法 Level 2 中不同音訊攻擊下所萃取出之浮水印 NC 值 (1)

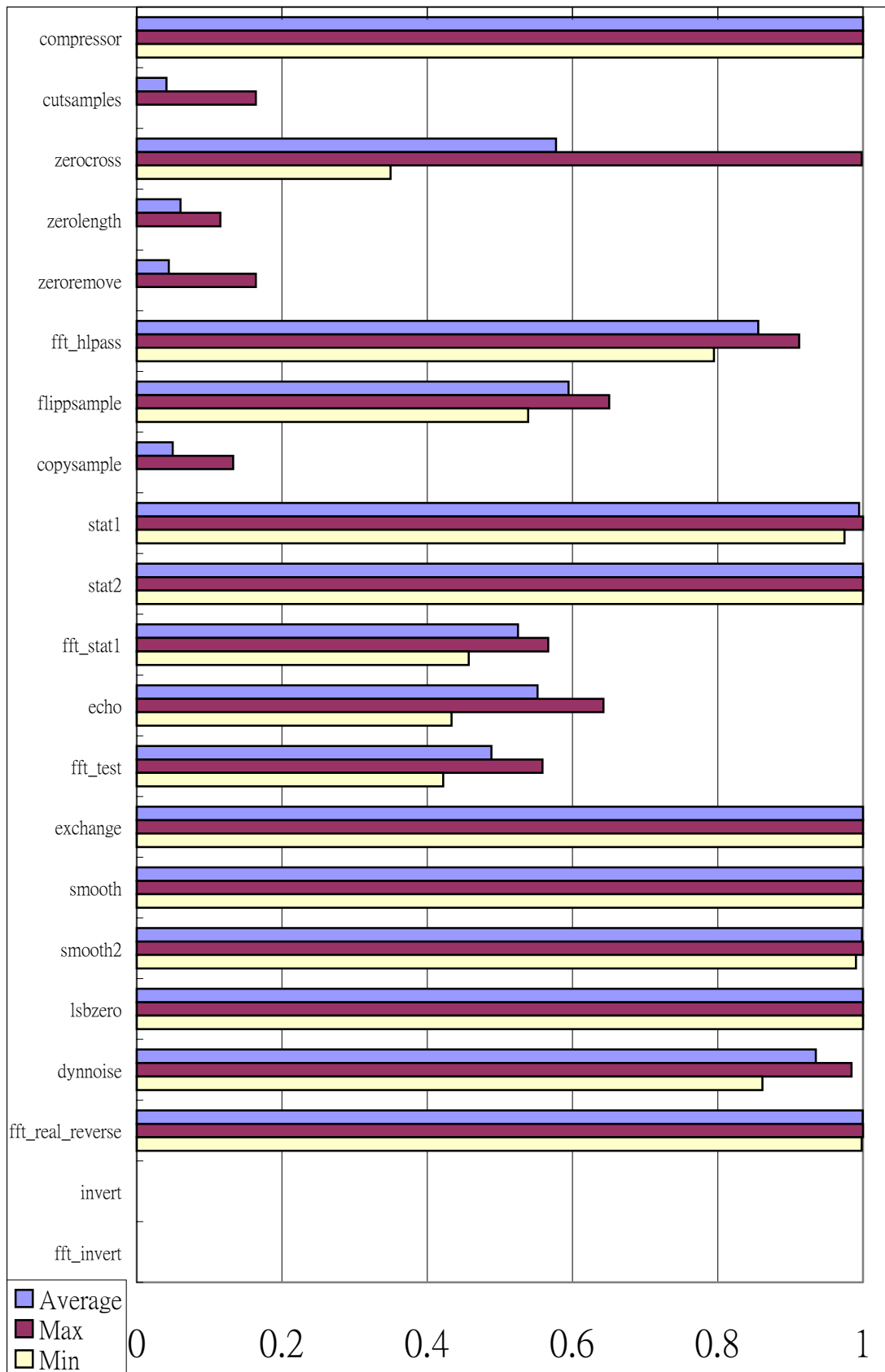


圖 5.11 方法 Level 2 中不同音訊攻擊下所萃取出之浮水印 NC 值 (2)

由以上的結果來看，normalize、amplify、compressor、exchange、smooth、lsb_zero、fft_real_reverse 等攻擊幾乎對數位浮水印沒有造成任何影響，而 rc_lowpass、addnoise、addbrumm、fft_hlpass、dynnoise 等攻擊只對數位浮水印有輕微的影響，其中雜訊強度在 addnoise_300 以上即可聽出明顯的雜訊成份存在。在 rc_highpass 及 addsinus 這兩種攻擊對於數位浮水印的影響較大，NC 值平均在 0.7 左右，萃取出來的數位二階浮水印大約還可以辨認得出來。

而 cutsamples、zerocross、zerolength、flippsample、copysample、echo 等攻擊則會大幅度的破壞數位浮水印，但是這些種類的攻擊也會對樂音音質造成嚴重的影響，經過這些攻擊後的樂音訊號也已經失去了鑑賞的價值了。

接著再看到 invert 及 fft_invert 的實驗結果，這兩種攻擊後的 NC 值都是 -1，這表示萃取出來的數位二階浮水印和嵌入時的數位二階浮水印是完全相反的，例如嵌入的是如圖 5.2 的數位二階浮水印，則萃取出來的數位二階浮水印則如圖 5.12 所示，就如同負片的效果一樣。



(a)



(b)

圖 5.12 Invert 攻擊後萃取出來的數位二階浮水印

(a) Level 1 (28 x 28) (b) Level 2 (32 x 32)

在 Stirmark for Audio 的所有測試中，平均而言 Level 2 的方法比 Level 1 的方法來的好一點點，這是因為 Level 2 所嵌入的頻帶範圍比 Level 1 要廣，受到攻擊時被影響的頻帶範圍比例相對的較低，且 Level 1 集中嵌入在低頻，在經過如 highpass filter 濾波之後，數位浮水印會被破壞的較為嚴重。

前面提到 Stirmark for Audio 中 resampling 這項攻擊有問題，因此 resampling 的測試是以 Cool-Editor 這套音訊處理軟體來做測試，測試步驟是先將嵌入浮水印後的樂音先經過 down sampling 到 22.05 kHz 後，再 up sampling 到 44.1 kHz。測試結果列於表 5.11，由測試結果可以看出 resampling 對數位浮水印幾乎沒有影響。

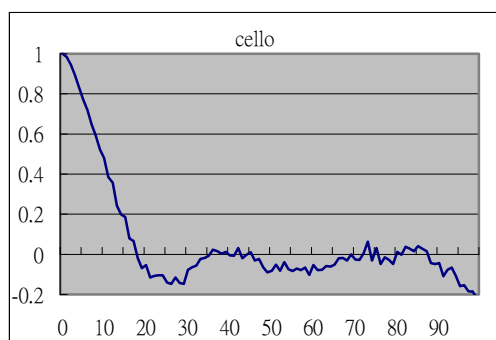
表 5.11 Resampling 攻擊後萃取出浮水印 NC 值

樂器名	cello	chorus	flute	guitar	harpsichord	horn
Level 1	0.992347	1	0.997449	0.984694	0.997449	0.997449
Level 2	0.996094	1	0.998047	0.990234	1	0.998047
樂器名	lute	orchestra	organ	piano	trumpet	violin
Level 1	1	0.994898	1	1	1	1
Level 2	1	0.998047	1	0.998047	1	1

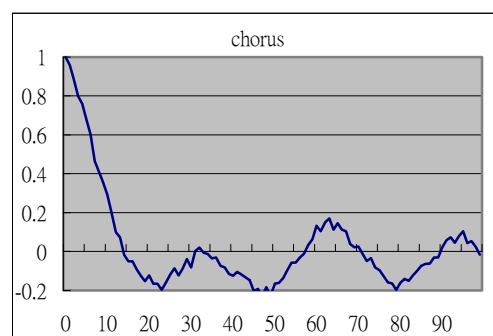
5.4 裁切測試

如同 4.3 節所述對於同步碼的需求，音訊延遲或裁切攻擊會造成音框無法對正的問題，也就是無法取得同步點而造成數位浮水印無法萃取或萃取出來的數位浮水印會嚴重失真的問題。

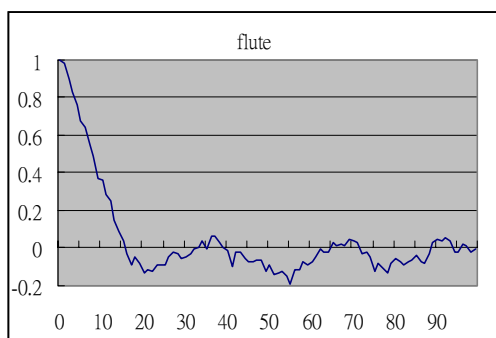
而在做裁切測試之前，我們將先測試取樣點位移對於數位浮水印的影響，首先將嵌入浮水印後的樂音訊號做取樣點位移，並測試其萃取出來的浮水印 NC 值。圖 5.13 為 Level 1 方法其中 8 種樂音經過取樣點位移 0~99 點後的結果。圖 5.14 為 Level 2 方法其中 8 種樂音經過取樣點位移 0~99 點後的結果。圖中橫軸的單位是取樣點位移的點數，縱軸的單位是 NC 值。



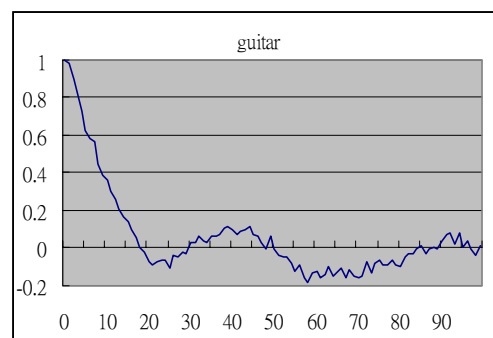
(a) cello



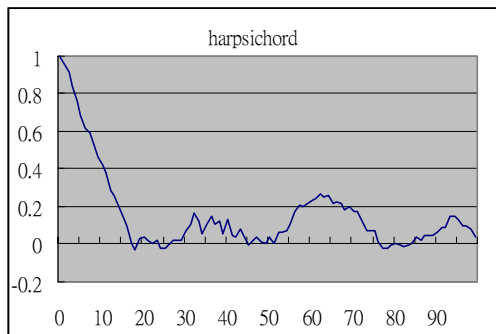
(b) chorus



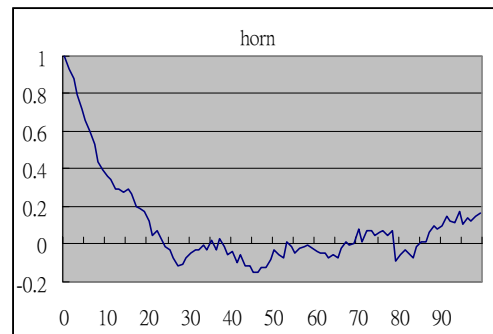
(c) flute



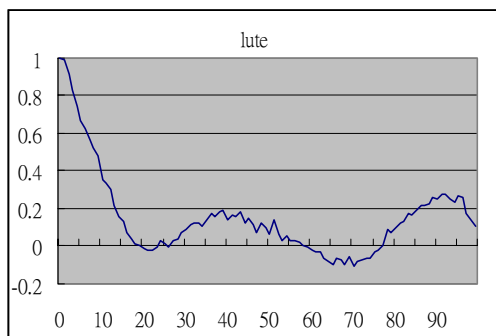
(d) guitar



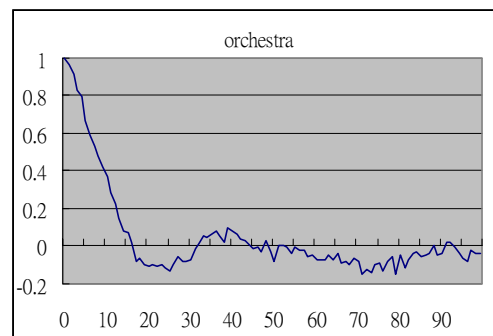
(e) harpsichord



(f) horn

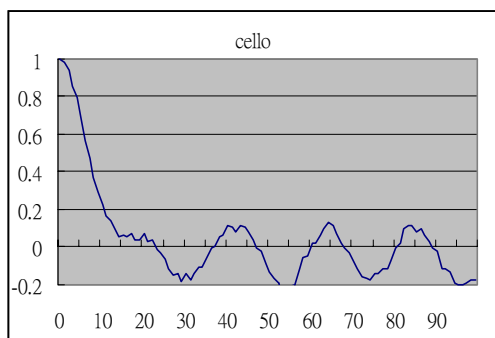


(g) lute

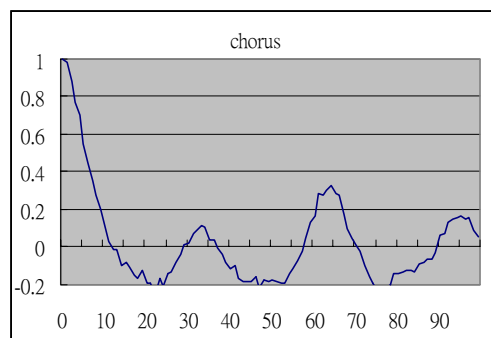


(h) orchestra

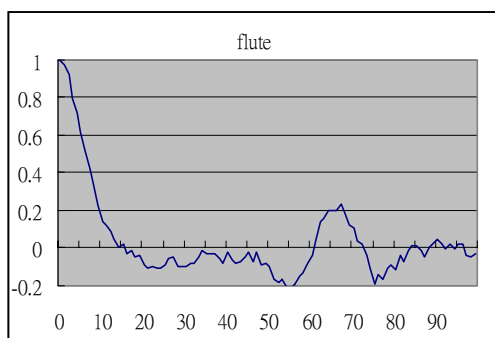
圖 5.13 Level 1 中取樣點位移後萃取出數位二階浮水印 NC 值



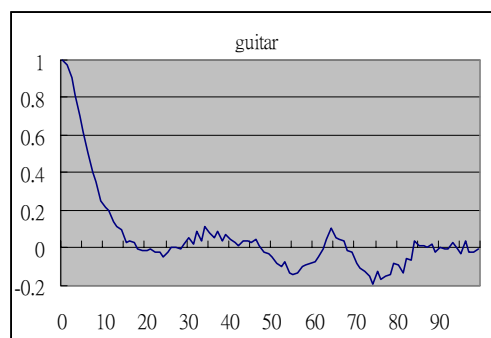
(a) cello



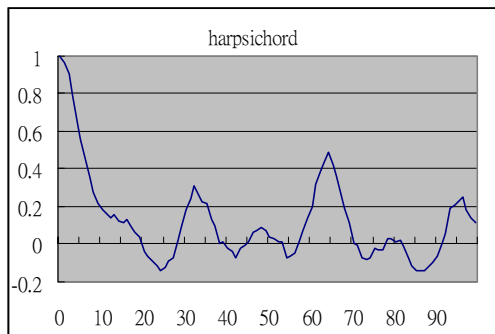
(b) chorus



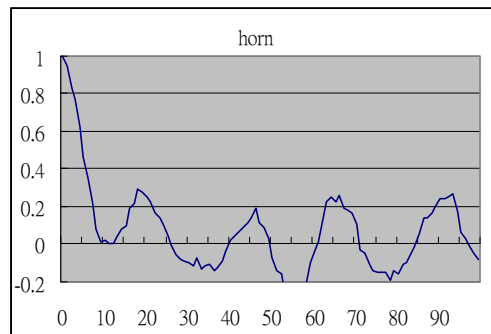
(c) flute



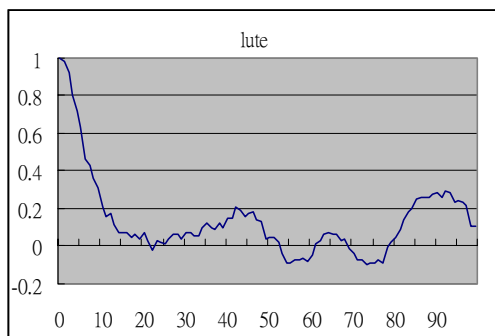
(d) guitar



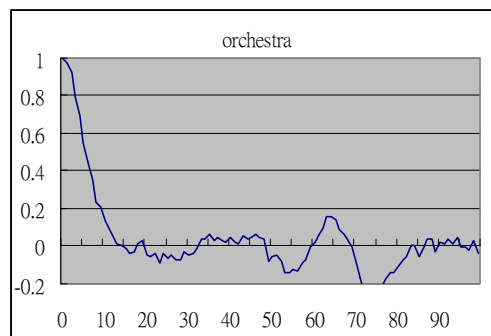
(e) harpsichord



(f) horn



(g) lute



(h) orchestra

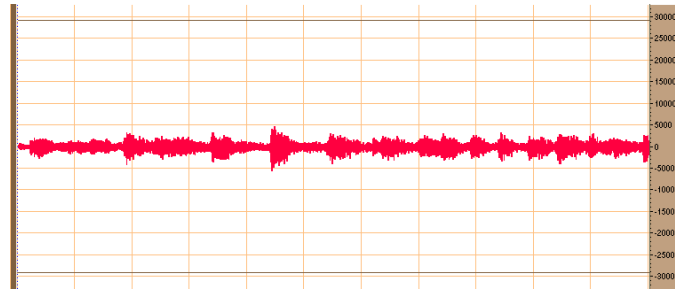
圖 5.14 Level 2 中取樣點位移後萃取出數位二階浮水印 NC 值

由實驗的結果可以得知，在大約 4 ~ 5 個取樣點的位移以內，萃取出數位浮水印的 NC 值還可以在 0.8 以上，超過 5 個取樣點的位移以上時，萃取出浮水印的錯誤率就已經相當高了。所以我們所嵌入的同步碼必須要能夠準確的搜尋到同步點的位置，至少要在 5 個取樣點的位移以內，才能將嵌入的數位浮水印正確的萃取出來。

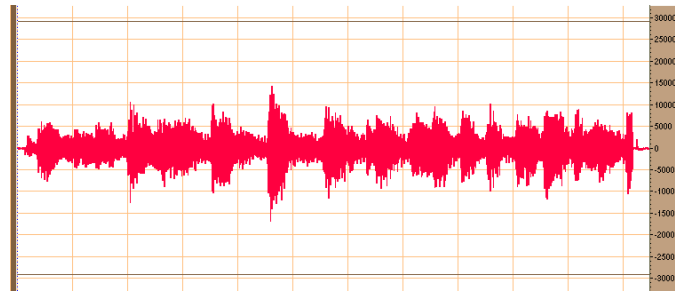
(1) D/A、A/D 測試

為了測試同步碼的效能，我們將測試經過音訊延遲後的樂音，其浮水印萃取的結果，並且配合 D/A 與 A/D 的攻擊來測試浮水印的效能，實驗方法如下：先將已嵌入同步碼與數位二階浮水印的樂音由 A 電腦的音效卡類比輸出，經由傳輸線連接到 B 電腦的類比輸入端，並且用錄音軟體錄製成 PCM 檔，由於 A 電腦的播音動作與 B 電腦的錄音動作不會同步，所以錄製出來的 PCM 檔會有若干的延遲，最後將此 PCM 檔輸入音訊浮水印系統萃取出數位二階浮水印。

我們測試兩種 D/A、A/D 的情況，一種情況為經過 D/A、A/D 後，樂音訊號的強度增強，另一種情況是經過 D/A、A/D 後，樂音訊號的強度減弱。測試樂音為吉他及小提琴，測試的浮水印方法為 Level 1，而同步碼是一個長度為 16 位元的固定二元資料 "1111000011110000"。測試結果如下，圖 5.15 為吉他的原始訊號及錄製成的 PCM 檔，圖 5.16 為吉他樂音的測試結果，圖 5.17 為小提琴的原始訊號及錄製成的 PCM 檔，圖 5.18 為小提琴樂音的測試結果。



(a) guitar 原始樂音



(b) guitar 錄製樂音

圖 5.15 Guitar 樂音經 D/A、A/D 後強度增強

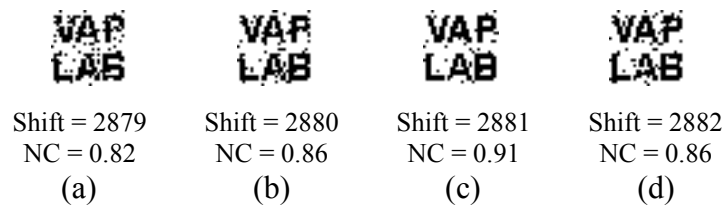
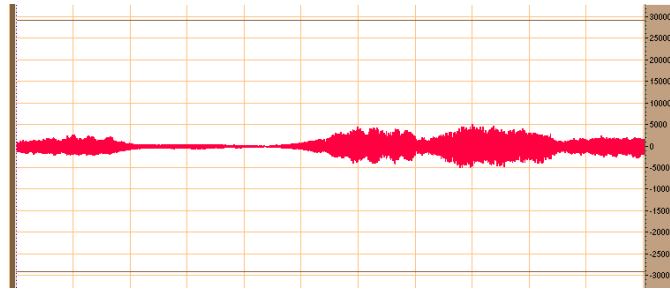


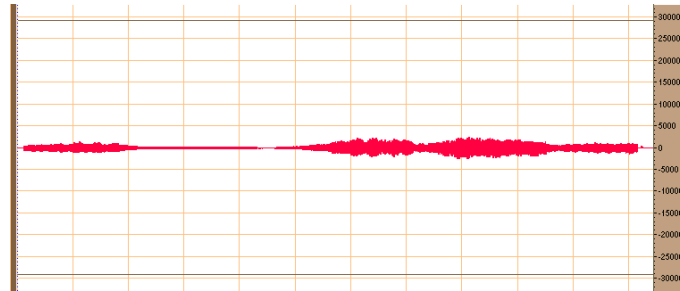
圖 5.16 Guitar 樂音測試結果 (a)手動位移 2879 個取樣點

(b)音訊浮水印偵測萃取結果 (c)手動位移 2881 個取樣點

(d)手動位移 2882 個取樣點



(a) violin 原始樂音



(b) violin 錄製樂音

圖 5.17 Violin 樂音經 D/A、A/D 後強度減弱

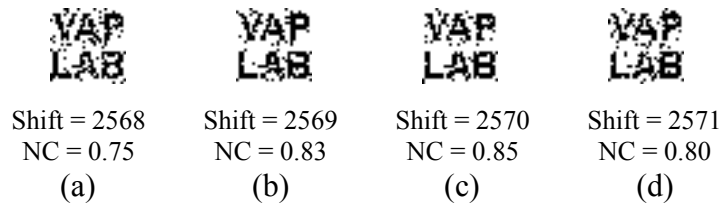


圖 5.18 Violin 樂音測試結果 (a)手動位移 2568 個取樣點

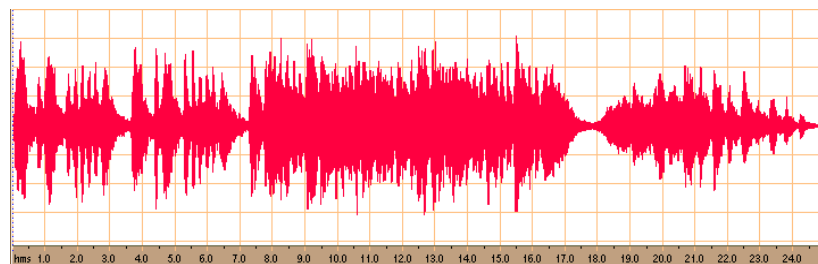
(b)音訊浮水印偵測萃取結果 (c)手動位移 2570 個取樣點

(d)手動位移 2571 個取樣點

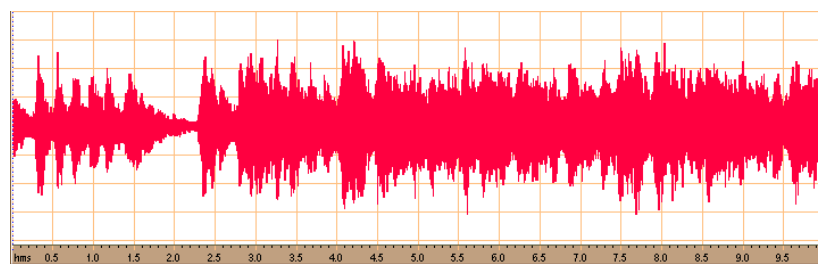
以圖 5.16 吉他樂音的測試結果來看，音訊浮水印系統偵測同步碼的結果，必須要位移 2880 個取樣點才會抓到同步點，數位浮水印的 NC 值是 0.86，而我們手動調整位移取樣點在 2880 附近找尋最佳值，得到的結果是位移 2881 個取樣點時，數位浮水印的 NC 值是 0.91。與音訊浮水印系統偵測的結果只相差一個取樣點。圖 5.18 小提琴樂音的測試，也得到相近的結果。

(2) 裁切測試

測試樂音採用 Mozart 的 Eine Kleine Nachtmusik, K525-2nd Movement，擷取前 25 秒、單聲道來嵌入同步碼與浮水印，一組【同步碼 | 浮水印】的音框長度為 250 個音框，25 秒的樂音約可重複嵌入 4 組【同步碼 | 浮水印】的資料。同步碼是一個長度為 16 位元的固定二元資料 "1111000011110000"，而數位浮水印是採用 Level 1 的方法。將此 25 秒嵌入浮水印後的樂音，裁去開頭 5 秒以及最後 10 秒，只餘下中間長度為 10 秒的樂音訊號，將此剪裁過的樂音輸入音訊浮水印系統，將浮水印萃取出來，實驗結果如圖 5.19。



(a) 原始長度 25 秒的樂音



(b) 裁切後長度 10 秒的樂音

**VAP
LAB**

Shift = 35500

NC = 1

(c) 萃取出來的數位浮水印

圖 5.19 Mozart：Eine Kleine Nachtmusik, K525-2nd Movement 裁切後樂音測試

實驗結果顯示裁切後的樂音，再經過 35500 個取樣點的位移後可以抓到同步點，萃取出數位二階浮水印 NC 值為 1，表示在萃取浮水印時，同部碼的搜尋可以有效的將裁切後的樂音音框同步對正。

第六章 結論

本論文以小波封包為基礎，配合人耳聲學模型，發展應用在樂音訊號中的數位浮水印系統。藉由小波封包將樂音訊號分成 29 個次頻帶，並且以各個次頻帶中個別的最小遮蔽臨介值為參考依據，再根據次頻帶中各個區塊的特性，將數位二階浮水印嵌入強健性較佳的區塊中，並以區塊及浮水印的極性產生一組安全鑰匙來確保數位浮水印的安全，而為了克服音訊同步的問題，也加入了同步碼搜尋的機制。

根據第五章的實驗結果顯示，在不影響感官音質的情況下，本音訊浮水印技術可以提供平均每個音框大約 4 ~ 6 個位元的浮水印容量，並且對於 MP3 音訊壓縮編碼有足夠的強健性，在 48 Kbps 以上的壓縮位元率時，NC 值平均都在 0.9 以上，與原始嵌入的浮水印十分相近。在 Stirmark 的測試中，對於大多數的音訊攻擊也有不錯的效果。而對於裁切攻擊，同步碼搜尋的機制也可以有效的找回同步點。

以同步碼搜尋雖可有效克服裁切攻擊，但是搜尋的過程需要耗費較多的時間，雖然在浮水印萃取端對於即時處理的要求較低，但是若能減低這方面的複雜度，將可以有較廣泛的應用。另外對於隨機丟棄取樣點（Random Drop）或隨機插入取樣點（Random Insert）這類型的攻擊，也是未來要克服的目標。

參考文獻

- [1] 林禎吉, 賴溪松, "數位浮水印的技術," *資訊安全通訊*, 第四卷第三期, June 1998.
- [2] 黃國峰, 張真誠, 黃明祥, "漫談數位浮水印技術," *資訊安全通訊*, 第五卷第四期, September 1999.
- [3] C. Xu, J. Wu and Q. Sun "Digital Audio Watermarking and Its Application in Multimedia Database," *Signal Processing and Its Applications*, 1999. ISSPA '99.
- [4] M. Sablatash and T. Cooklev, "Compression of High-Quality Audio Signals, Including Recent Methods Using Wavelet Packets," *Digital Signal Processing*, vol. 6, no. 10, 1996, Page(s). 96-107.
- [5] Y. Karellic and D. Malah, "Compression of High-Quality Audio Signals Using Adaptive Filterbanks and A Zero-Tree Coder," *Electrical and Electronics Engineers in Israel*, 1995.
- [6] P. Srinivasan and L. H. Jamieson, "High-Quality Audio Compression Using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, April 1998, Page(s). 1085-1093.
- [7] I.J. Cox, M.L. Miller and J.A. Bloom, "Watermarking applications and their properties," *Information Technology: Coding and Computing, 2000. Proceedings*. Page(s) 6-10.
- [8] S.Craver, N. Memon, B.-L. Yeo and M.M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques : limitations, attacks, and implications," *IEEE Journal on Selected Areas in Communications*, Volume: 16 Issue: 4 , May 1998. Page(s) 573–586.
- [9] M. Arnold, "Audio watermarking: features, applications and algorithms," *Multimedia and Expo, 2000. ICME 2000*. Volume 2. Page(s) 1013–1016.
- [10] M.D. Swanson, Bin Zhu and A.H. Tewfik, "Current state of the art, challenges and future directions for audio watermarking," in *Proc. Int. Conf. on Multimedia Computing and Systems*, 1999. Page(s) 19-24 vol.1.
- [11] M. Steinebach, F.A.P. Petitcolas, F. Raynal, J. Dittmann, C. Fontaine, S. Seibel, N. Fates, and L.C. Ferri, "StirMark benchmark: audio watermarking attacks," in *Proc. Int. Conf. on Information Technology: Coding and Computing*, 2001.

Page(s) 49–54.

- [12] Stirmarkbench "Stirmark for Audio"
<http://ms-smb.darmstadt.gmd.de/stirmark/stirmarkbench.html>
- [13] L. Boney, A.H. Tewfik and K. Hamdy, "Digital watermarks for audio signals," *Processdings of the 1996 international conference on multimedia computing and systems*.
- [14] J.F. Tilki and A.A. Beex, "Encoding a hidden auxiliary channel onto a digital audio signal using psychoacoustic masking," *engineering new century*, IEEE 1997.
- [15] Y. Wang, "A new watermarking method of digital audio content for copyright protection," *proceedings of ICSP'98 fourth international conference on signal processing*, 1998.
- [16] D. Gruhl, D. Lu and W. Bender "echo hiding, " *Lecture Notes in Computer Science, Information Hiding*, Vol.1174, pp.295~315, Springer, 1996
- [17] H.-O. Oh, J.-W. Seok, J.-W. Hong and D.-H. Youn, "New echo embedding technique for robust and imperceptible audio watermarking," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 2001. Page(s) 1341-1344 vol.3.
- [18] C.-C. Tseng, C.-B. Tseng and S.-L. Lee, "Audio Watermarking Based on Linear Prediction and Vector Quantization," *Processdings of CVGIP 2001*.
- [19] J. Lacy, S.R. Quackenbush, A.R. Reibman, D. Shur and J.H. Snyder, "On combining watermarking with perceptual coding," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1998. Page(s) 3725-3728 vol.6.
- [20] 張立鈞, "Digital watermarking of Audio Signals," 碩士論文, 中正大學, 1999.
- [21] C.-P. Wu, P.-C. Su and C.-C. Kuo, "Roubust and Efficient Digital Audio Watermarking Using Audio Content Analysis," *Proceedings of SPIE, Security and Watermarking of Multimedia Contents II*, January, 2000. Page(s) 382-392.
- [22] C. S. Burrus, R. A. Gopinath, and H. Guo, "Introdution to Wavelets and Wavelet Transforms," 1998.
- [23] P. E. Kudumakis and M. B. Sandler, "Wavelet Packet Based Scalable Audio Coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process. 1996*, pp. 41-44.
- [24] W. K. Dobson, J. J. Yang, K. J. Smart, and F. K. Guo, "High Quality Low Complexity Scalable Wavelet Audio Coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process. 1997*, pp. 327-330.

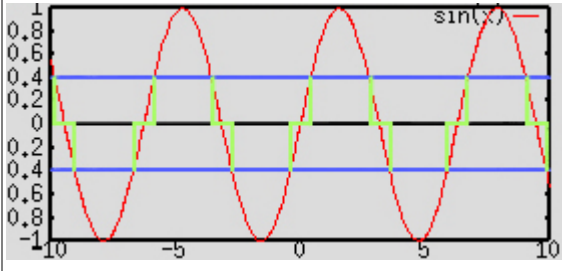
- [25]P. Philippe, F. Moreau de Saint-Martin, and L. Mainard, "On The Choice of Wavelet Filters for Audio Compression," in *Proc. Int. Conf. Acoust., Speech, Signal Process. 1995*, pp. 1045-1048.
- [26]P. E. Kudumakis and M. B. Sandler, "On The Performance of Wavelets for Low Bit Rate Coding of Audio Signals," in *Proc. Int. Conf. Acoust., Speech, Signal Process. 1995*, pp. 3087-3090.
- [27]I. Daubechies, "Ten Lectures on Wavelets," no. 61 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia, 1992.
- [28]R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Information Theory*, vol. 38, pp. 713-718, March, 1992.
- [29]E. Zwicker and H. Fastl, *Psychoacoustics Facts and Models*, Springer, Berlin, Heidelberg, 1990.
- [30]ISO/IEC 11172-3 : "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio (MPEG-1) ," 1992.
- [31]M. Rosa, F. Lopez, P. Jarabo and S. Maldonado, "New method to translate the psycho-acoustic information to the wavelet domain," *EURASIP Conf. DSP for multimedia Communications and Services*, Krakow, Jun 1999.
- [32]M.-T. Chou, "Audio Compression Using Wavelet Packets And A Zero-Tree Coder With Psychoacoustic Modeling," *Master thesis, National Central University, Taiwan*, 1999.
- [33]R.-S. Lin, "Scalable Audio Compression Using Wavelet Packets Decomposition and Embedded Zero Tree Coding," *Master thesis, National Central University, Taiwan*, 2000.
- [34]D.-K. Lien, "Packet Loss Recovery Using Data Hiding for Embedded Zero Tree Wavelet Packet Audio," *Master thesis, National Central University, Taiwan*, 2001.
- [35]L.-L. Lee, "A Key-based image Watermarking system using subblock composition in DCT domain and DWT domain," *Master thesis, National Central University, Taiwan*, 2001.

附錄 A

Documentation for "Stirmark for Audio" v0.2

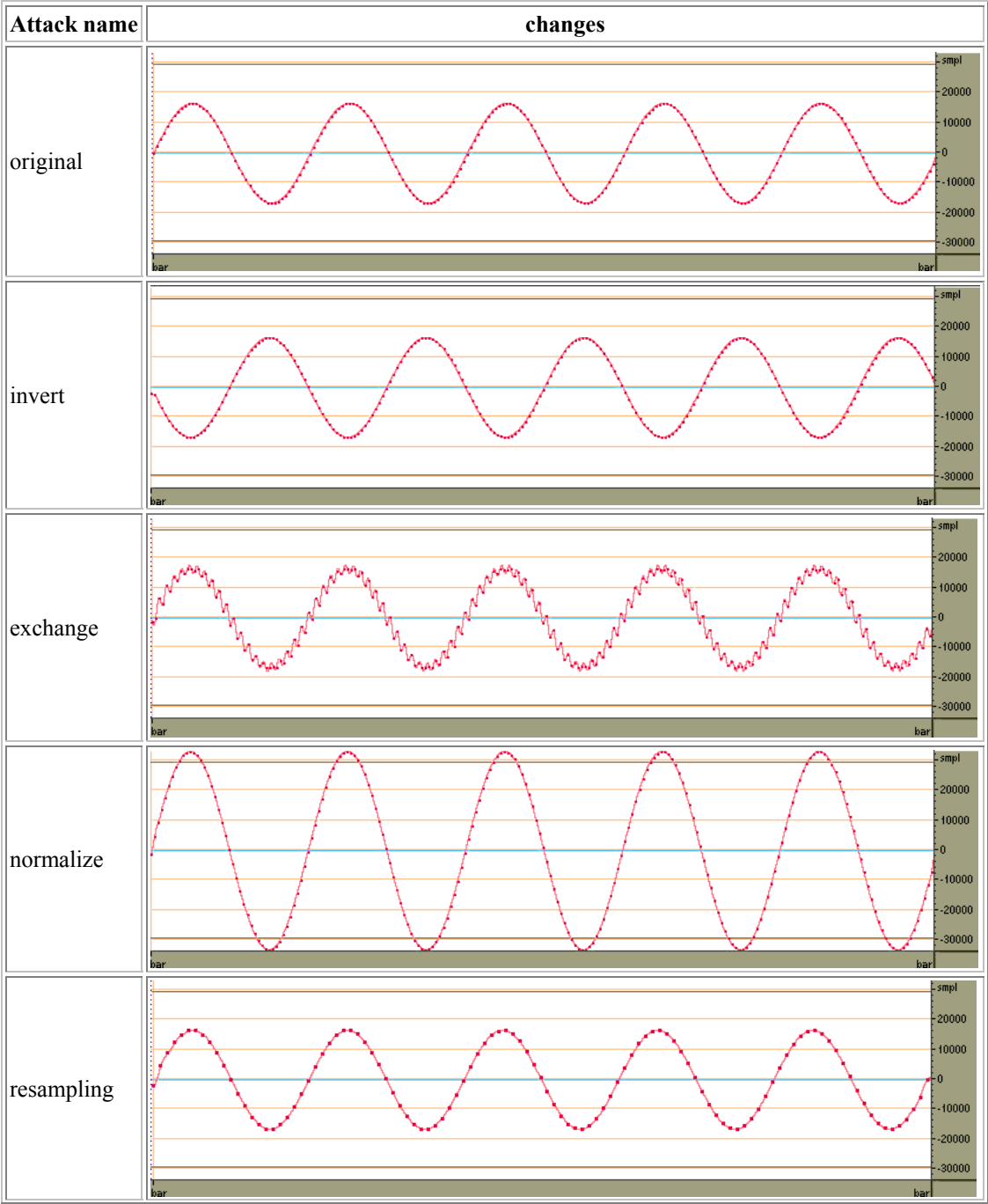
This table shows you all attacks and the parameter used from the parameter file.

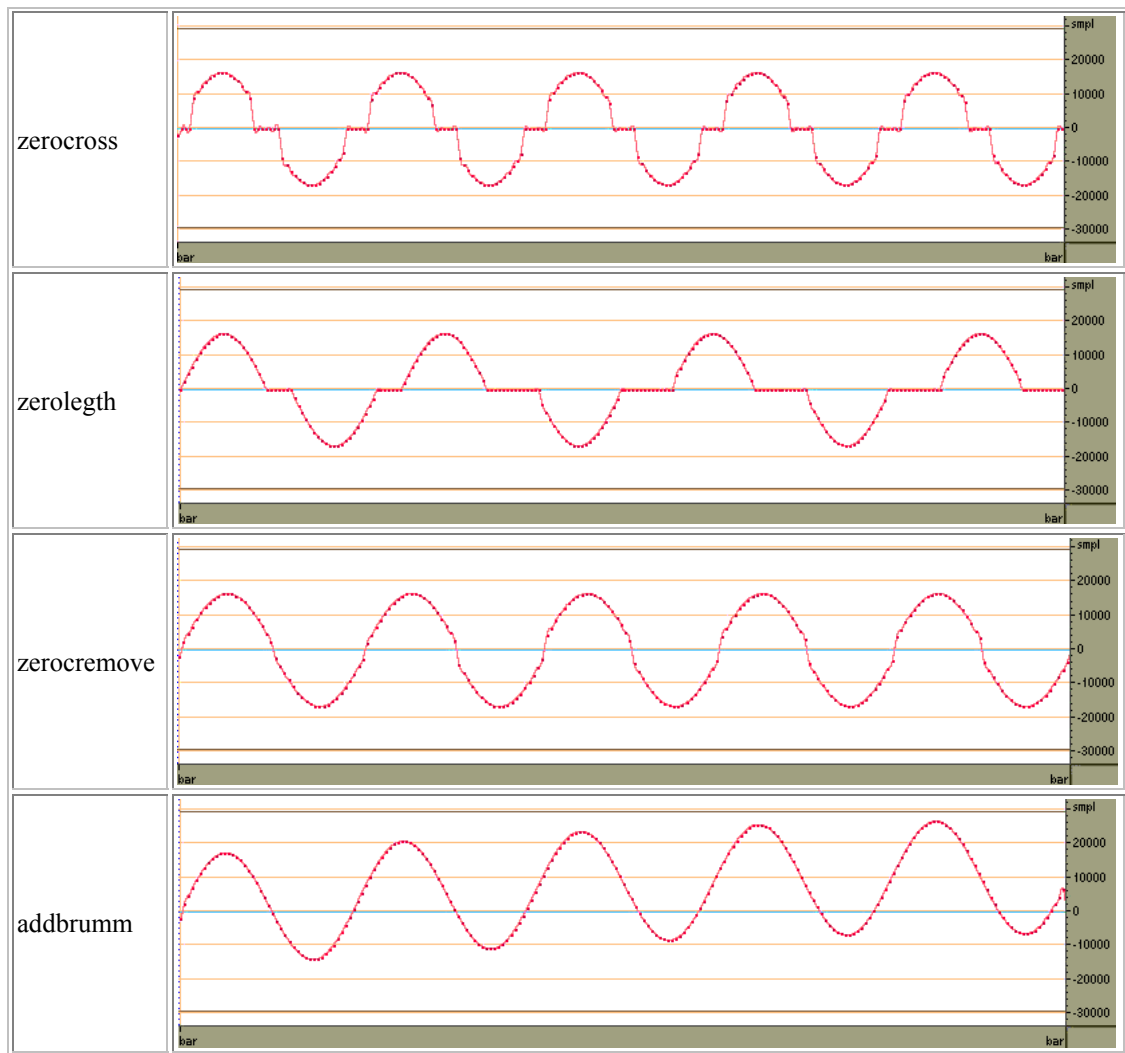
Attack Name	Description	used parameter
AddBrumm	Adds buzz or sinus tone to the sound. The unit of three values are samples and for the frequency hertz (Hz).	AddBrummfrom AddBrummto AddBrummstep AddBrummFreq
AddDynNoise	This attack adds a dynamic white noise part to the samples. The given parameter sets the maximum noise value.	Dynnoise
AddFFNoise	Adds white noise to the samples in the FFT room. The value "FFNoise" sets the power of this attack to add the noise.	FFTSIZE FFNoise
AddNoise	Adds in differnt steps white noise to the samples. The unity is in sample values. The value "0" adds nothing and "32768" the absolite distorted maximum.	Noisefrom Noiseto Noisestep
AddSinus	Adds a sinus signal to the sound file. With this attack you can insert a disturb signal in the frequency band where the watermark is located. The unity of the frequency parameter is hertz (Hz) and samples.	AddSinusFreq AddSinusAmp
Amplify	Changes the loudness of the audio file. For example the value "100" does not change the amplify and a value "50" means a half loudness.	Amplify
Compressor	This attack works like a compressor. You can increase or decrease the loudness of quietly passages. The unit of the threshold is decibel (dB). The "CompressValue" describes how the sample can be changed. "2" means that the loudness of all samples in the threshold will be the half. If the value is less than "1", the compressor is an expander and will increase the loudness.	ThresholdDB CompressValue
CopySample	Is like the FlippSample attack but this attack copies the samples between the samples with a distance of FlippDist=.	Period FlippDist FlippCount
CutSamples	Removes samples from the audio file. If the value of "Remove" is "10000" then this attack removes every "10000" samples "RemoveNumber" samples periodic.	Remove RemoveNumber
Echo	Adds an echo to the sound file. The given value means the distance the echo.	Period
Exchange	Swaps two sequent samples for all samples	
ExtraStereo	Increases the stereo part of the file. If the file does not have a stereo part (expl. mono), then this attack do not have an effect	ExtraStereofrom ExtraStereo ExtraStereo ExtraStereo ExtraStereo ExtraStereo
FFT_HLPassQuick	Is like the RC-High- and RC-LowPass attack and use the same	FFTSIZE

	frequencies from the parameter file. FFT window size can be set with the "FFTSIZE" parameter. This attack do not fade between the FFT windows so it is possible to hear knacks	HighPassFreq LowPassFreq
FFT_Invert	Inverts all samples (real and imaginary part) in the FFT room.	FFTSIZE
FFT_RealReverse	Reverses only the real part from the FFT.	FFTSIZE
FFT_Stat1	Attack	
FFT_Test	I will do some tests on this attack. At present I swap some samples inside from FFT.	FFTSIZE
FlippSample	Swaps samples inside the sound file periodically. It swaps every "Period" "FlippCount" samples with samples which have a distance of "FlippDist". Important: Period > FlippDist > FlippCount! (PICTURE)	Period FlippCount FlippDist
Invert	Inverts all samples in the audiofile.	
LSBZero	This attack sets all LSB to "0" (zero).	
Normalize	Normalize the amplify to the maximum value.	
Nothing	This attack does nothing with the audio file. The watermark could be retrieved. If not, the watermarking algorithm can be a snake oel!	
RC-HighPass	Simulates a high pass filter build with a resistance (R) and a capacitor (C).	HighPassFreq
RC-LowPass	Simulates a low pass filter like RC-HighPass.	LowPassFreq
Resampling	Changes the sample rate of the sound file.	SampleRate
Smooth	This attack smooth the samples. The seting sample value depends on the samples before and after the modify point.	
Smooth2	Is like Smooth, but the neighbor samples are voted a little bit different.	
Stat1	Attack	
Stat2	Attack	
VoiceRemove	Is the opposite to ExtraStereo. This attack removes the mono part of the file (mostly where the voice is). If the file do not have a stereo part (expl. only mono) then everything will be removed.	
ZeroCross	 <p>This attack is like a limiter. If the sample value is less the given value (threshold - blue line), all samples are set to zero (green line).</p>	ZeroCross
ZeroLength	If a sample value is exactly "0" (zero) then this attack inserts more samples with the value "0" (zero). (PICTURE)	ZeroLength
ZeroRemove	This attack removes all samples where the value is "0" (zero).	

This table shows you the signal after special attacks.

The attacks are used with very high parameters to see what are changed. if you use normal parameters, then you can't see a difference between the original and attackd file.





Author : [Andreas Lang](#)

Last modified: Tue Feb 12 12:50:59 CET 2002