

國立交通大學

電機與控制工程系

碩士論文

MP3 編碼法之研究與實現

Research and Implementation of MP3 Encoding
Algorithm

研究生：張芷燕

指導教授：吳炳飛 教授

中華民國九十一年七月

MP3 編碼法之研究與實現

Research and Implementation of MP3 Encoding Algorithm

研 究 生：張芷燕

Student：Tze-Ying Chang

指導教授：吳炳飛

Advisor：Bing-Fei Wu

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

July 2002

Hsinchu, Taiwan, Republic of China

中華民國九十一年七月

MP3 編碼法之研究與實現

學生：張芷燕

指導教授：吳炳飛

國立交通大學電機與控制工程學系碩士班

摘 要

MPEG-1/Audio 標準規格提供了對高品質數位音樂的編解碼方法，其中又以 MPEG-1/Audio 第三層編碼法（MP3）在相同位元傳率下的效果最佳，其複雜度亦最高。

本論文主要探討 MP3 的編碼法，並且期望在一 16 位元定點運算之數位訊號處理器（DSP）上實現 MP3 編碼器。文中介紹 MP3 音訊編碼標準的原理與實現方法，並且說明在實作上為配合 DSP 有限的資源所做的種種改善，包含了定點數的運算與減少記憶體空間使用之方法。

Research and Implementation of MP3 Encoding Algorithm

Student: Tze-Ying Chang

Advisor: Prof. Bing-Fei Wu

Department of Electrical and Control Engineering
National Chiao Tung University

Abstract

The MPEG-1/Audio standard provides a method for the compression of high-quality digital audio, especially the MPEG-1/Audio layer3. It performed the best quality of the three layers but also needs the most computational complexity.

This thesis is to investigate the MP3 encoding algorithm and expects to implement MP3 encoder on a 16-bit, fixed-point digital signal processor (DSP). This thesis introduces the principles of MP3 encoding standard and the implementation of MP3 encoder. Because of the limited resource of the DSP, this thesis also describes some modified methods for the DSP, including the fixed-point operation and memory space reducing.

誌謝

首先我要感謝我的指導老師吳炳飛教授，在老師的指導下，我學習到治學應有的態度與方法，也才能夠順利的完成此篇論文。

感謝胡益強學長、許子偉同學、魏宏宇學弟在研究上對我的幫助。也要謝謝實驗室所有的學長姊、同學與學弟妹們，除了課業上的協助與討論之外，他們讓我在交大度過了愉快的研究生生活。

最後我要謝謝我親愛的父母張昭景先生與林美雲女士和妹妹瑋玲，因為他們的支持與照顧使我能夠順利完成碩士學位。

目錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖目錄	vi
表目錄	vii
第一章 簡介	1
1.1 研究背景與動機	1
1.2 章節安排	2
第二章 MP3 編碼原理	3
2.1 濾波器排 (The Filter Bank)	4
2.1.1 分析子頻帶濾波器 (Analysis Subband Filter)	4
2.1.2 改良式離散餘弦轉換 (MDCT)	7
2.2 聲響心理模型 (The Psychoacoustic Model)	9
2.2.1 靜音門檻曲線 (The Absolute Threshold of Hearing)	10
2.2.2 臨界頻帶 (Critical Bands)	11
2.2.3 頻域上的遮蔽效應 (Frequency masking)	13
2.2.4 時軸上的遮蔽效應 (Temporal Masking)	15

2.2.5	Perceptual Entropy	17
2.2.6	第二聲響心理模型	19
2.3	位元分配及量化 (Bit Allocation and Quantization)	21
2.3.1	位元分配(Bit allocation)	21
2.3.2	非均勻量化(Nonuniform quantization)	23
2.3.3	霍夫曼編碼(Huffman Encoding)	25
2.4	位元串格式 (Bitstream Formatting)	27
2.4.1	位元儲藏處(bit reservoir)	28
2.4.2	MPEG-1/Audio 的檔頭格式	29
第三章 MP3 編碼實作		32
3.1	定點數運算	33
3.2	分析子頻帶濾波器	35
3.3	改良式餘弦轉換	39
3.4	第二聲響心理模型	41
3.5	量化和位元分配	46
3.5.1	Iteration Loop	46
3.5.2	外部迴圈	47
3.5.3	內部迴圈	50
第四章 結果與未來展望		54
參考資料		57

圖目錄

圖 2-1	MPEG-1/Audio Layer3 編碼流程	3
圖 2-2	分析子頻帶濾波器與 MDCT	4
圖 2-3	分析子頻帶濾波器	4
圖 2-4	$C[n]$ 與 $h[n]$ 之係數 ($n=0\sim511$)	5
圖 2-5	分析子頻帶濾波器的頻譜分析	6
圖 2-6	MDCT 的四種窗框及使用情形	7
圖 2-7	長窗框假象處理	8
圖 2-8	靜音門檻曲線 (取樣頻率 44.1 kHz)	10
圖 2-9	遮噪門檻曲線與靜音門檻曲線	13
圖 2-10	遮噪門檻曲線與 SMR	14
圖 2-11	時軸上的遮蔽效應	15
圖 2-12	前迴音與轉換編碼的關係	16
圖 2-13	量化器之輸入輸出關係圖	23
圖 2-14	主要資訊的組織	25
圖 2-15	MPEG-1 layer3 的資料格式	27
圖 2-16	MP3 的位元流 (bitstream) 與儲藏機制示意圖	28
圖 2-17	MPEG-1/Audio 檔頭區的位元串格式 (共 32 位元)	29
圖 3-1	反正切函數的逼近方法	34
圖 3-2	分析子頻帶濾波器運算流程圖	35
圖 3-3	暫存器 \underline{X}_{512}	36
圖 3-4	$\underline{M}_{32\times64}$ 的大小由 32×64 縮減成 32×31	36
圖 3-5	$\underline{M}'_{32\times31}$ 的大小由 32×31 縮減成 16×31 , $\underline{M}''_{16\times31}$	38
圖 3-6	適用於第三編碼層的第二聲響心理模型流程圖	43
圖 3-7	長區塊的展開函數 (取樣頻率 44.1 kHz)	45
圖 3-8	Iteration Loop 的流程圖	47
圖 3-9	外部迴圈的流程圖	48
圖 3-10	內部迴圈的流程圖	51
圖 4-1	有無使用聲響心理模型之位元傳率與 MNR 的關係	55

表目錄

表 2-1	理想的臨界頻帶	11
表 2-2	適用 MP3 編碼的第二聲響心理模型之比例因子頻帶 (44.1 kHz)	19
表 2-3	big_value 區間所用 32 個霍夫曼表的特性	26
表 2-4	位元傳率對照表	30
表 2-5	模式延伸對照表	31
表 3-1	假象處理所用的係數 c_i	40
表 3-2	MP3 編碼器各方塊運算時間的比較	46
表 4-1	MP3 編碼器在 DSP 上所使用的記憶體空間	54
表 4-2	有無使用聲響心理模型之位元傳率與 MNR 的關係	56

第一章 簡介

1.1 研究背景與動機

近十年來，音樂的表示、儲存方式漸漸由類比變成了數位。市面上，音樂 CD 逐漸取代了錄音帶而成為主流。數位音訊在網路、通訊、多媒體與電腦系統的應用也越來越多。使用數位方式來表示音樂訊號有許多的好處，包括資料不易受環境雜訊的干擾、容易再製、方便保存等等，並且較容易對音樂訊號做特殊的處理，如混音 (mix)、濾波 (filtering) 和等化 (equalization)。但相對的，數位資料也有其限制，如頻寬、儲存容量與價格上的限制。

數位音訊是由類比訊號轉換而來，轉換的方式是透過取樣與量化。一般而言，數位音訊是用 Pulse Code Modulation (PCM) 的格式表示。而所謂的 **CD 音質** (CD-quality)，相當於雙聲道的聲音訊號以 44.1 kHz 的頻率來取樣，再使用 16 個位元來做量化而得到的數位資料所能達到之聲音品質。

所以為了達到 CD 音質，每秒鐘約有 1.4 Mbits 的資料量，若以一首四分半鐘的歌曲來估計，就需要至少 45 MByte 的儲存空間，更不用說要透過網路傳送數位音訊了！在這個前提下，數位音訊的壓縮成為了一個重要的課題。

由 ISO/IEC 組織所制訂的 MPEG-1/Audio 編碼[1]標準是目前最被廣泛使用的音訊壓縮方法之一。MPEG-1/Audio 總共定義了三層獨立的編碼方式，其中又以第三層編碼最為複雜，在相同位元傳率 (bitrate) 下亦能達到最好的音樂品質。經過 MPEG-1/Audio 第三層編碼後所得到的結果，就是一般常聽到的 MP3 音樂格式。

在個人電腦上，MP3 的編解碼軟體已經相當普遍，但在一般的可攜式產品或家電用品中，仍以 MP3 的解碼功能為主，如 MP3 隨身聽之類的產品。雖然 MP3 編碼的複雜度比解碼高出許多，但在資訊家電的時代即將來臨與使用 MP3 格式壓縮的音樂日漸普及的情況下，發展非電腦平台上的 MP3 編碼器勢在必行。

在本篇論文中，主要研究 MPEG-1/Audio 第三層編碼的原理與方法，並且在實作上配合 DSP (Digital Signal Processor) 的特性做一些調整與化簡的工作，以期能在 DSP 上實現 MP3 編碼法。

1.2 章節安排

本論文的章節安排如下：

第一章說明研究背景與動機。

第二章介紹 MP3 的編碼原理。

第三章敘述 MP3 編碼為了在 DSP 上實現所做的改進。

第四章為結果與未來展望。

第二章 MP3 編碼原理

MPEG-1/Audio 包含三個獨立編解碼層，它們具有不同運算複雜度的編解碼架構，提供不同的音訊品質及壓縮率，以符合各式各樣的應用需求。第一編碼層最簡單，提供最少量的壓縮，常用於一般消費性音訊系統。第二編碼層較複雜，常用於消費性或專家級音訊系統。第三編碼層最複雜，可提供最大的壓縮，適用於整體服務數位網路上的音訊傳輸[11]。本章將介紹第三層編碼的原理。

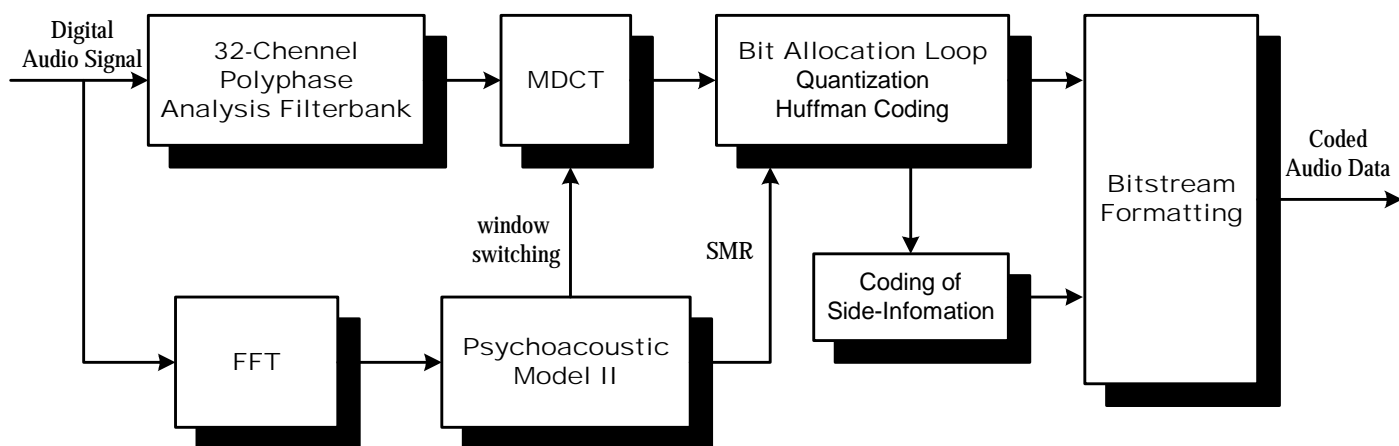


圖 2-1 MPEG-1/Audio Layer3 編碼流程

圖 2-1 是 MPEG-1 Layer3 (以下簡稱 MP3) 音訊編碼的過程。以單聲道而言，MP3 的一個編碼框包含 1152 個聲音取樣（一個編碼框又相當於 2 個 granules，每個 granule 包含 576 個聲音取樣），每個取樣為 16 位元。MP3 編碼時，首先將原始輸入的 16-bit PCM 音訊經過濾波器排分析（Filter Bank Analysis），轉換成 32 個等頻寬的子頻帶訊號（Subband Signals），然後透過改良式離散餘弦轉換（MDCT，Modified Discrete Cosine Transform），將每個子頻帶訊號，再細分為 18 個次頻帶。然後根據第二聲響心理模型（Psychoacoustic Model II）所提供的訊號遮噪比（SMR, Signal-to-Mask Ratio），對每一子頻帶訊號，做位元分配及量化編碼。最後只要將編碼後的資料依照 MPEG-1 定義的位元串的类型輸出即可。

2.1 濾波器排 (The Filter Bank)

濾波器排包含分析子頻帶濾波器和改良式離散餘弦轉換兩部分，如圖 2-2 所示。

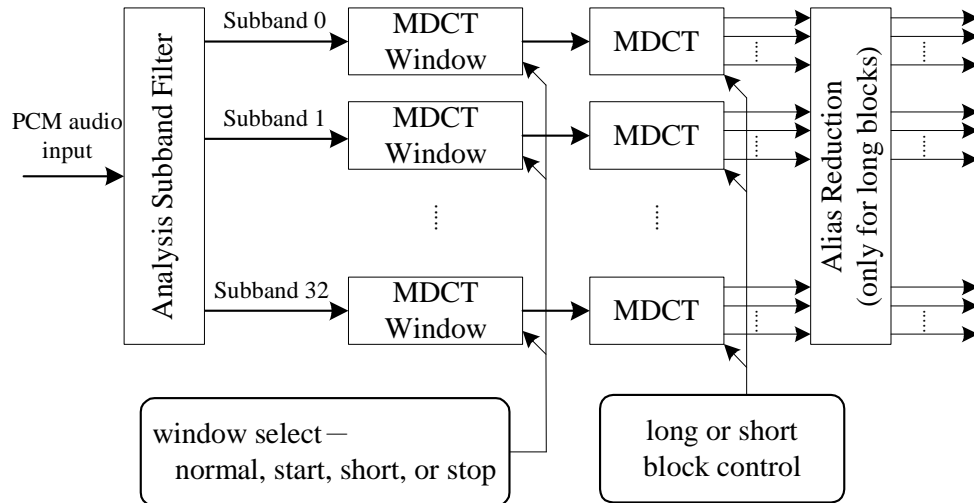


圖 2-2 分析子頻帶濾波器與 MDCT

2.1.1 分析子頻帶濾波器 (Analysis Subband Filter)

MPEG-1/Audio 使用多重相位濾波器 (Polyphase Filters) 來做訊號的子頻帶分析，當聲音訊號輸入多重相位濾波器之後會被轉成 32 個等頻寬的子頻帶訊號。每輸入 32 個 PCM 訊號做一次濾波器排分析，然後可以得到 32 個輸出（每個子頻帶有一個結果），由於一個編碼框有 1152 個 PCM 訊號，所以共需要 36 次的子頻帶分析。

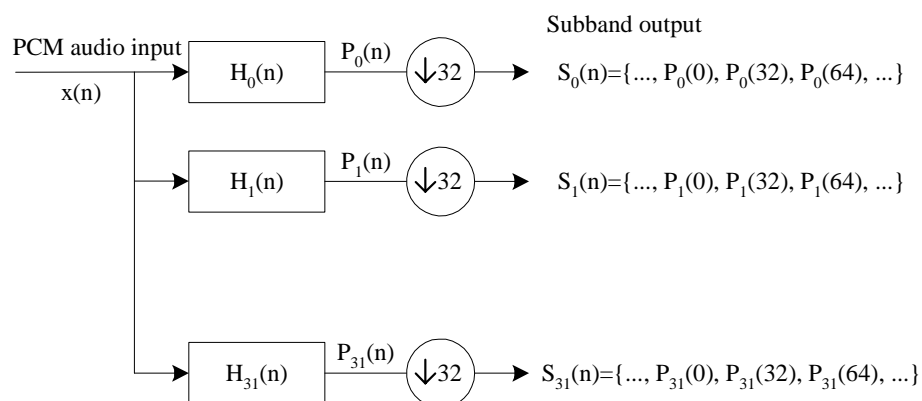


圖 2-3 分析子頻帶濾波器[5]

多重相位濾波器的數學式如下：

$$H_i[n] = h[n] \times \cos\left[\frac{\pi \times (2 \times i + 1) \times (n - 16)}{64}\right] \quad \text{where } i = 0 \sim 31, n = 0 \sim 511 \quad (2-1)$$

其中 $h[n]$ 是一個低通濾波器。我們可以根據[1]中所定義的一組窗框係數 $C[n]$ ($n=0 \sim 511$)，找出 $h[n]$ ，見圖 2-4。 $C[n]$ 與 $h[n]$ 的關係如下：

$$h[n] = \begin{cases} -C[n], & \text{if the integer part of } \frac{n}{64} \text{ is odd} \\ C[n], & \text{o.w.} \end{cases} \quad \text{where } i = 0 \sim 511 \quad (2-2)$$

或者可以寫成

$$C(m = k + 64j) = (-1)^j \times h(k + 64j), \text{ where } k=0 \sim 63, j=0 \sim 7, \text{ and } m=0 \sim 511. \quad (2-3)$$

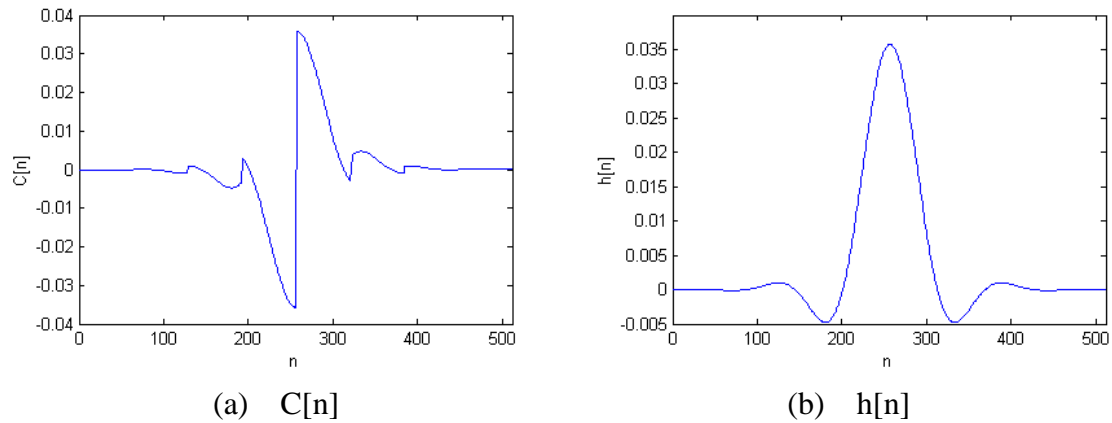


圖 2-4 $C[n]$ 與 $h[n]$ 之係數 ($n=0 \sim 511$)

參考圖 2-3，將輸入的 PCM 訊號分別通過 32 個濾波器 $H_i[n]$ ($i=0 \sim 31$) 之後可以得到 $P_i[n]$ 如下：

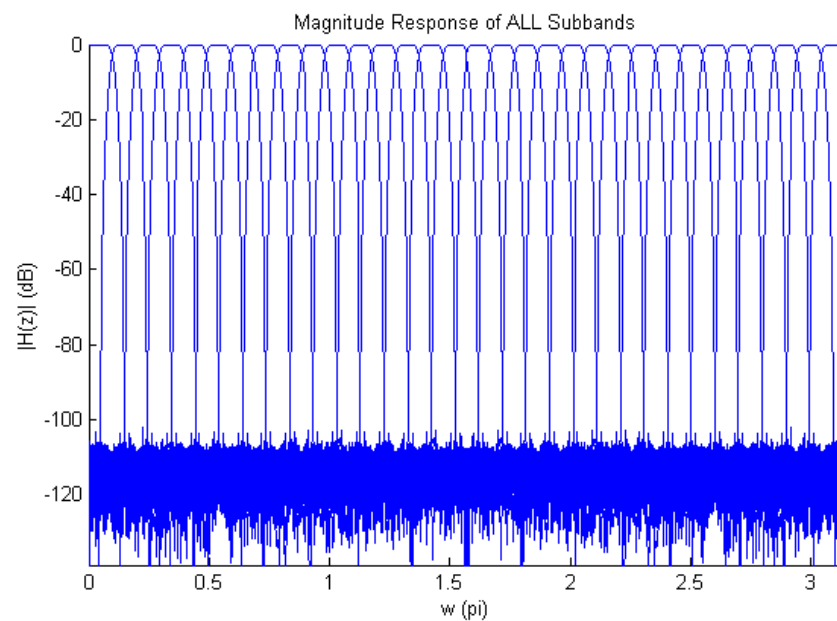
$$P_i[n] = \sum_{m=0}^{511} x[n-m] \times H_i[m] \quad \text{where } i = 0 \sim 31 \quad (2-4)$$

最後，再將 $P_i[n]$ down-sampling by 32，則可得到 32 個子頻帶訊號：

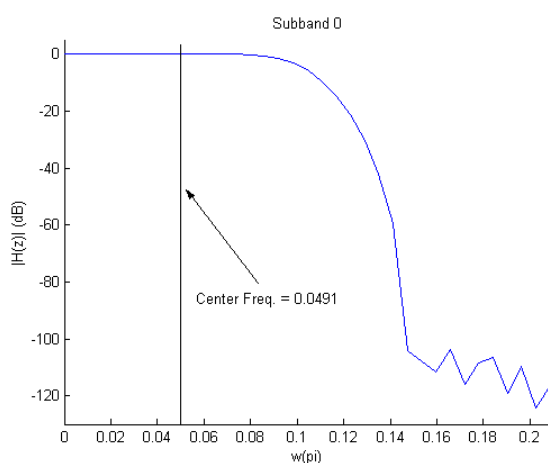
$$\begin{aligned} S_i[n] &= P_i[32n] \\ &= \sum_{m=0}^{511} x[32n-m] \times H_i[m] \quad \text{where } i = 0 \sim 31 \end{aligned} \quad (2-5)$$

其中 $S_0[n] \sim S_{31}[n]$ 是濾波器排分析後的結果。

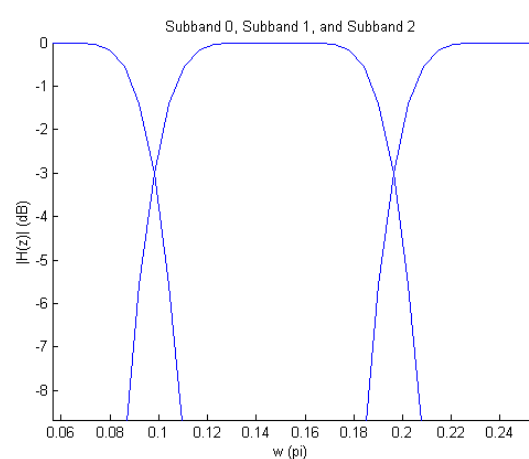
圖 2-5 畫出了 32 個子頻帶的頻率響應，每個子頻帶的頻寬皆為 $\pi/32$ ，第 i 個子頻帶的中心頻率則是 $\pi \times (2 \times i + 1) / (2 \times 32)$ ，而相鄰的子頻帶交會在 3dB 處，如圖 2-5(c)所示。以取樣頻率為 44.1 kHz 的訊號為例(將頻域範圍 2π 對應到 44.1 kHz)，每個子頻帶的頻寬為 689 Hz；在圖 2-5(b)中，第 0 個子頻帶的中心頻率約為 345 Hz。



(a) 32 個等寬的子頻帶



(b) 第 0 個子頻帶



(c) 各子頻帶的交點在 -3 dB

圖 2-5 分析子頻帶濾波器的頻譜分析

2.1.2 改良式離散餘弦轉換 (MDCT)

將原始的音訊經過濾波器排分析，分成 32 個等寬的子頻帶訊號後，為了提高頻譜的解析度，將每個子頻帶訊號再經 MDCT 細分成數個頻線訊號。MDCT 的運算共包含了 MDCT 窗框、MDCT 與長窗框假象處理三個部分，見圖 2-2。

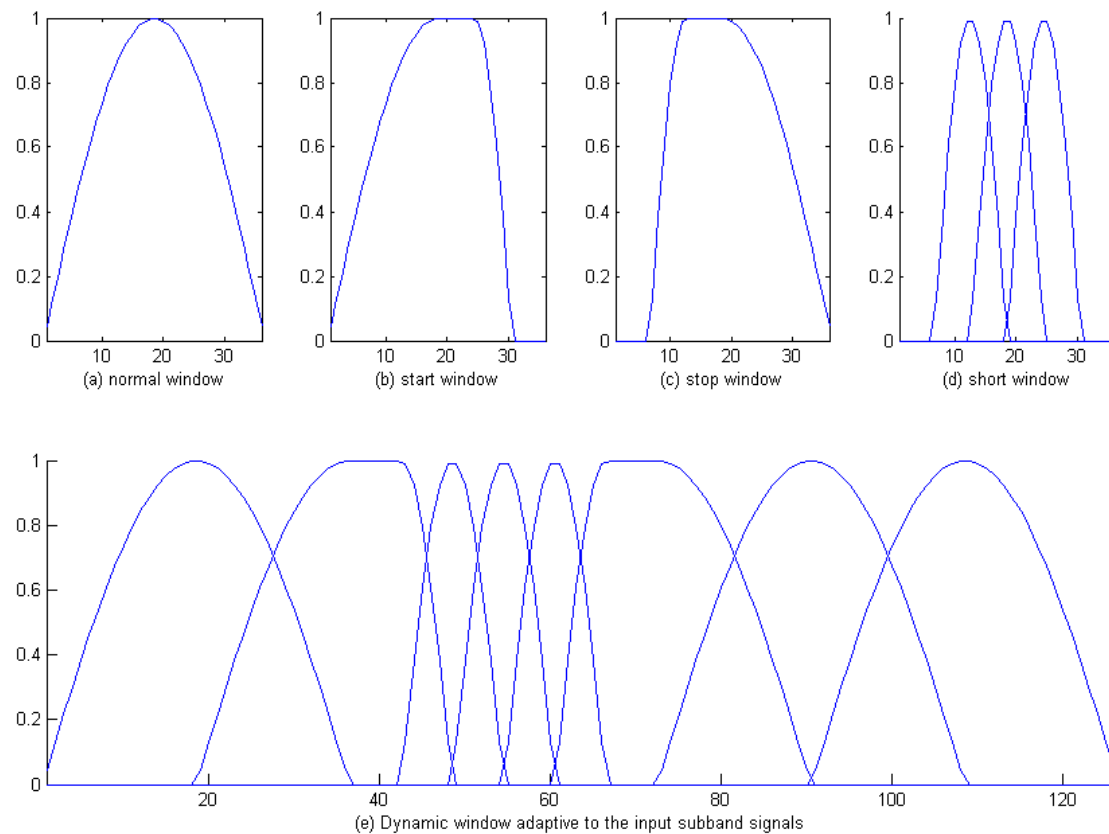


圖 2-6 MDCT 的四種窗框及使用情形

首先介紹 MDCT 的四種窗框：長窗框 (Normal Window)、長短窗框 (Start Window)、短窗框 (Short Window) 與短長窗框 (Stop Window)，如圖 2-6(a)~(d) 所示。使用長窗框做轉換，可得較好的頻譜解析度，而使用短窗框做轉換，則可得較佳的時間軸解析度。至於長短窗框是用於長窗框要轉換到短窗框時的過渡窗框，短長窗框則相反，四種窗框的切換方式如圖 2-6(e) 所示。窗框的選擇是依據第二聲響心理模型分析音訊特性之後所得到的資訊，在一般音訊穩定的情形下，使用長窗框來提供最細的頻譜解析度。然而當子頻帶訊號變動大時，需變化窗框長度以提供較精細的時間軸解析度，以控制前迴音 (pre-echo) 雜訊不被人

耳察覺，前迴音的產生原因與避免其不被人耳察覺的方法將於 2.2.4 小節中介紹。

決定好窗框後就可以做 MDCT 的運算，運算式如下：

$$X_i = \sum_{k=0}^{n-1} Z_k \cos \left[\frac{\pi}{2n} \left(2k + 1 + \frac{n}{2} \right) (2i + 1) \right] \quad \text{for } i = 0, 1, \dots, \frac{n}{2} - 1. \quad (2-6)$$

其中 Z_k 是子頻帶訊號與 MDCT 窗框相乘後的結果，若此處選擇短窗框則做短區塊的 DCT 運算 ($n=12$)，否則就做長區塊的 DCT 運算 ($n=36$)。

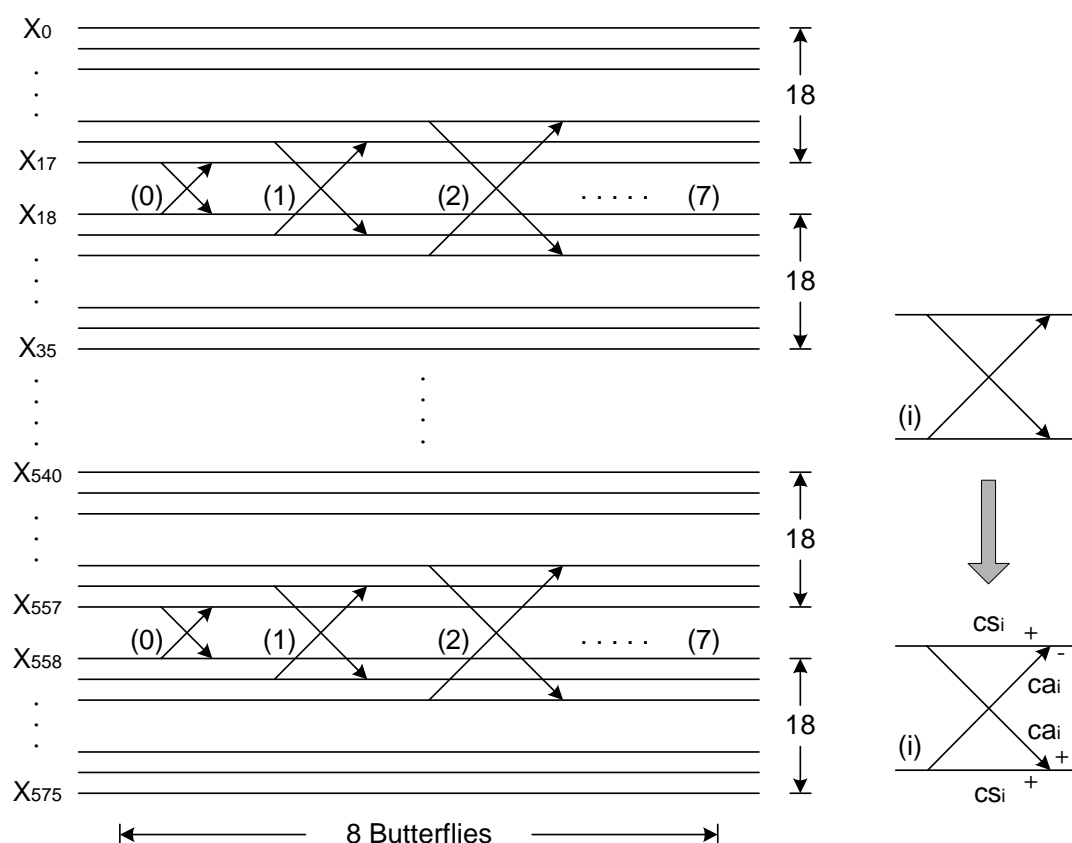


圖 2-7 長窗框假象處理

在使用長窗框得到較細的頻譜解析度時，同時會有假象 (Aliasing) 的產生。由於濾波器排的特性，當原始音訊被分成 32 個子頻帶時，在頻譜上可見鄰近的子頻帶間有明顯的重疊現象，而處於重疊區間的訊號將會同時影響兩個子頻帶。所以在經過 MDCT 轉成頻線訊號時，需對鄰近相對應的頻線訊號做特別處理，以減少因假象所造成的雜訊，影響音訊品質。假象處理的方式是將處在相對應位置的頻線之能量做一定比例的增減，如圖 2-7 所示，其中 ca_i 與 cs_i 在[1]中有定義。

2.2 聲響心理模型 (The Psychoacoustic Model)

MPEG-1/Audio 之所以能夠達到高壓縮率並維持一定的聲音品質，最主要的原因，就是它採用了聲響心理模型來模擬人耳的聽覺。利用人耳聽覺感知上形成的遮蔽效應 (Masking Effect) 所得到的遮噪門檻曲線 (Masking Thresholds)，決定各個子頻帶所容許的最大量化誤差，使得量化後的失真能不被聽見。

MPEG-1/Audio 提供了兩個聲響心理模型，對於任一編層皆可使用。其中第一聲響心理模型較為簡單，適用於高傳輸率；第二聲響心理模型則比較複雜，在較低傳輸率的情況下，也能維持聲音的品質。此兩模型皆先將音訊經傅立葉頻譜轉換，再對映到臨界頻帶 (Critical Bands)，並區分出單頻 (Tonal, sinusoid-like) 及非單頻 (Non-Tonal, noise-like) 成份，依其所在的頻率位置與強度大小，分別計算遮噪門檻曲線，而整合成整體遮噪門檻曲線，並對映成每個子頻帶訊號編碼時所需的位元數。

在介紹人耳的聽覺特性之前，必須先定義一個表示聲音強度的名詞，Sound Pressure Level (SPL)。SPL 是一個評量聽覺刺激強度的標準[9]，單位為 dB。接著說明人耳的聽覺特性。

人耳對音訊的遮噪能力與訊號的能量大小、訊號的頻率位置，以及訊號的特性有關都有關係。能量大的訊號能遮蔽較大的噪音、非單頻訊號的遮噪性比單頻訊號來得好，而高頻訊號的遮蔽效應也比低頻訊號要強。此外，遮噪能力也與雜訊的頻率位置，以及雜訊發生的時間有關：對同一音訊而言，對高頻雜訊的遮噪能力優於低頻；對其後發生的雜訊之遮噪性優於較其前發生的雜訊。因此，經由以上各種的聲響心理實驗，推導出模擬人耳聽覺的感知模式，建立各種遮噪門檻曲線[11]，下面就一一的來介紹。

2.2.1 靜音門檻曲線 (The Absolute Threshold of Hearing)

在安靜的環境下，人耳能夠聽到較細微的聲音，例如針掉落地面的聲音，但仍有許多聲音無法被聽到。依照這樣的特性，我們可以找出一條靜音門檻曲線(又稱為 The Threshold in Quiet)，如圖 2-8 所示，若聲音的強度低於靜音門檻曲線，表示人耳聽不到這個聲音。這條曲線可以由 (2-7) 的非線性方程式來逼近[10]。

$$Tq(f) = 3.64 \times \left(\frac{f}{1000}\right)^{-0.8} - 6.5 \times e^{-0.6 \times \left(\frac{f}{1000} - 3.3\right)^2} + 10^{-3} \times \left(\frac{f}{1000}\right)^4 \quad (\text{dB SPL}) \quad (2-7)$$

其中 f 代表頻率 (Hz)，一般而言，人耳所能感知的頻率範圍是 10~20,000 Hz。

$Tq(f)$ 則是在頻率 f 處的靜音門檻值。

觀察這條曲線可以發現人類的聽覺在低頻與高頻的地方比較不敏銳，而在頻率大約 3~4 kHz 處最為敏銳。

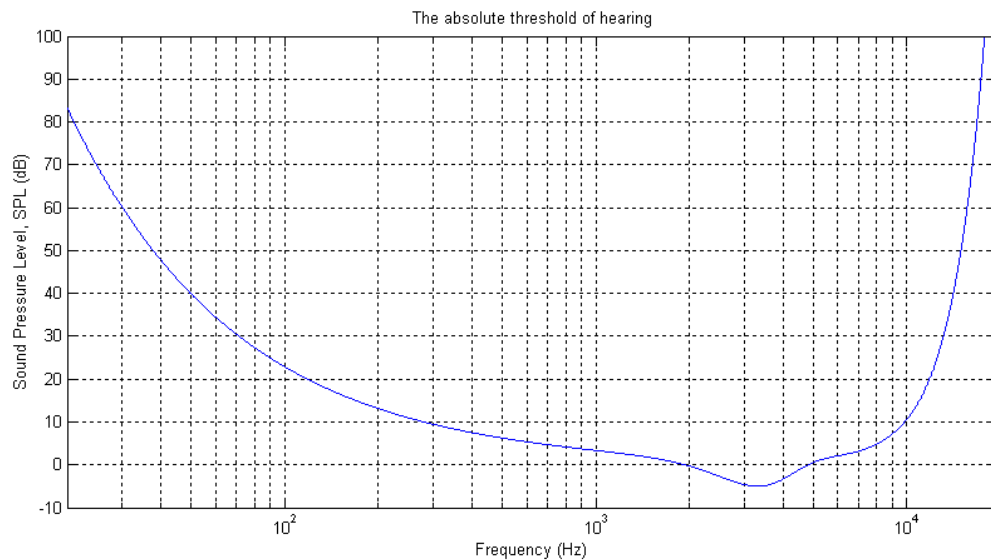


圖 2-8 靜音門檻曲線

2.2.2 臨界頻帶 (Critical Bands)

由於人耳對不同頻率的解析度不同，MPEG-1/Audio 將 22 kHz 以內可感知的頻率範圍，依不同編碼層、不同取樣頻率，劃分成 23 到 26 個臨界頻帶 (critical band)。表 2-1 列出理想的臨界頻帶之中心頻率與頻寬[4]。

一個臨界頻帶的”距離”通常又稱為一個巴克 (Bark)，(2-8) 式可以將頻率 (Hz) 對映到 Bark 表示[9]：

$$z(f) = 13 \times \tan^{-1}(0.00076 \times f) + 3.5 \times \tan^{-1}\left[\left(\frac{f}{7500}\right)^2\right] \quad (\text{Bark}) \quad (2-8)$$

而對於每個臨界頻帶的頻寬也可以由 (2-9) 式求得[9]：

$$BW_c(f) = 25 + 75 \times \left[1 + 1.4 \times \left(\frac{f}{1000}\right)^2\right]^{0.69} \quad (\text{Hz}) \quad (2-9)$$

Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	– 100	14	2150	2000 – 2320
2	150	100 – 200	15	2500	2320 – 2700
3	250	200 – 300	16	2900	2700 – 3150
4	350	300 – 400	17	3400	3150 – 3700
5	450	400 – 510	18	4000	3700 – 4400
6	570	510 – 630	19	4800	4400 – 5300
7	700	630 – 770	20	5800	5300 – 6400
8	840	770 – 920	21	7000	6400 – 7700
9	1000	920 – 1080	22	8500	7700 – 9500
10	1175	1080 – 1270	23	10500	9500 – 12000
11	1370	1270 – 1480	24	13500	12000 – 15500
12	1600	1480 – 1720	25	19500	15500 –
13	1850	1720 – 2000			

表 2-1 理想的臨界頻帶

若以第 9 個臨界頻帶（中心頻率為 1000Hz）為例，可以算出頻寬與 Bark，見式（2-10）。結果與表 2-1 所列之數值十分接近。

$$\begin{cases} z(1080) - z(920) = 1.022 \text{ Bark} \\ BW_c(1000) = 162.22 \text{ Hz} \end{cases} \quad (2-10)$$

觀察表 2-1，臨界頻帶在頻率 500 Hz 以下幾乎維持著等寬的頻帶，頻寬大約是 100 Hz，當頻率超過 500 Hz，臨界頻帶的頻寬就隨著頻率的增加而遞增，一直到第 24 個臨界頻帶頻寬甚至已經到達 3.5 kHz。從這裡也可看出人耳對低頻的解析度較高頻為佳。

2.2.3 頻域上的遮蔽效應 (Frequency masking)

在頻域上，SPL 較大的訊號會對頻率相近的訊號產生遮蔽效應，如圖 2-9 所示。圖中有三個訊號被 masker 遮蔽，其中一個訊號甚至低於靜音門檻曲線。注意到圖中的遮噪曲線在往低頻的方向較為傾斜，而往高頻的方向則較為平緩，這表示高頻的訊號比較容易被遮蔽。

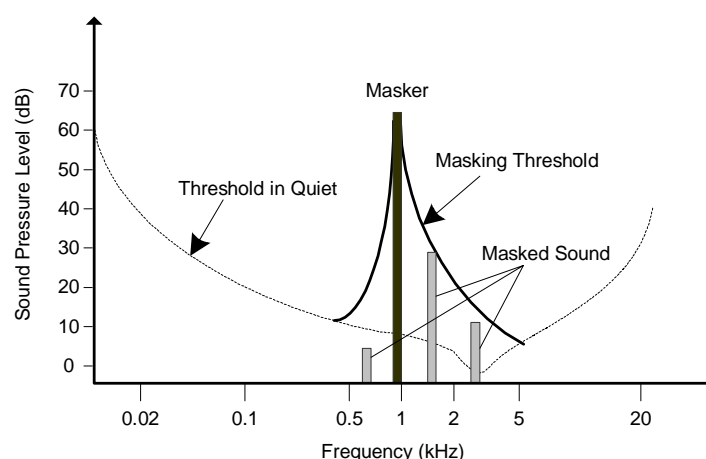


圖 2-9 遮噪門檻曲線與靜音門檻曲線

圖 2-10 為 SMR、SNR、NMR 與遮噪門檻曲線之間關係的示意圖。從 masker 到遮噪門檻值的距離稱為 signal-to-mask ratio (SMR)，SMR 的最大值出現在這個臨界頻帶的左邊界 (圖 2-10 中的 A 點)，聲響心理模型必須提供 SMR 作為位元分配與量化的依據。假設用 SNR(m) 來表示訊號經過 m 個位元量化之後的 signal-to-noise ratio (SNR)，那麼在這個臨界頻帶中，用來量測人耳可感知的失真之參數 mask-to-noise ratio (MNR) 可以計算如下：

$$MNR(m) = SNR - SMR(m) \quad (\text{in dB}) \quad (2-11)$$

MNR 會隨著量化層的不同而改變，當 MNR 的值大於零，量化誤差就聽不出來。

目前我們就單獨一個 masker 的遮蔽效應來討論，而實際上聲響心理模型會整合所有 masker 的遮噪門檻曲線，計算出一條整體遮噪曲線 (global masking threshold)，用來估計人耳在各個頻帶對於量化失真的容忍度。

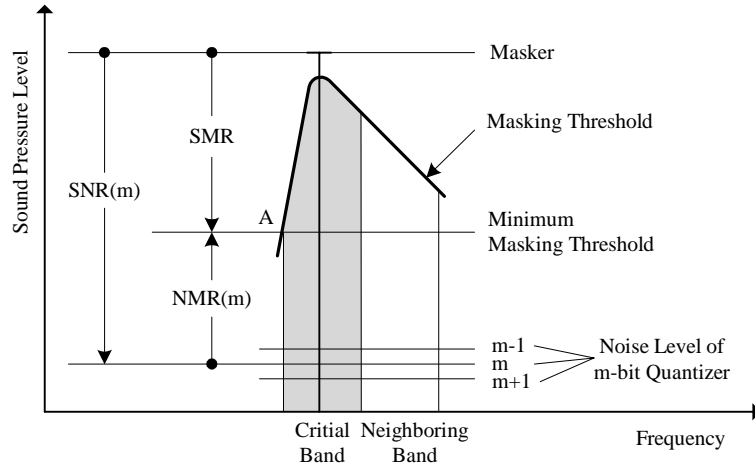


圖 2-10 遮噪門檻曲線與 SMR

遮蔽效應並不只限於 masker 所存在的臨界頻帶，對於鄰近的頻帶亦有所影響。位於一個臨界頻帶中心的 masker，對於其他臨界頻帶的遮噪曲線有著一些可預測的影響，這種效應稱為遮噪延展性（the spread of masking）。通常可以用一個數學式來表示[12]，稱之為展開方程式（The Spreading Function）：

$$SF_{dB}(x) = 15.81 + 7.5 \times (x + 0.474) - 17.5 \sqrt{1 + (x + 0.474)^2} \quad \text{dB} \quad (2-12)$$

其中 x 與兩個臨界頻帶之間的距離有關，單位是 Bark； $SF_{dB}(x)$ 的單位是 dB。

2.2.4 時軸上的遮蔽效應 (Temporal Masking)

人耳聽覺在時間軸上也會產生遮蔽效應（又稱為 Non-simultaneous Masking）。在一個很短的時間內，若出現了兩個聲音，不論聲音出現的先後順序，其中 SPL 較大的聲音（masker）會遮蔽 SPL 較小的聲音（maskee）。這時 SPL 較大的聲音在時軸上會產生兩種遮蔽效應：若 maskee 出現在前，稱為 pre-masking；反之則稱為 post-masking，如圖 2-11 所示。其他的聲音若出現在圖中深灰色的地方就會被遮蔽。值得注意的是 post-masking 所影響的時間較長，至少有 160ms，而 pre-masking 大約只有 post-masking 十分之一的時間[6]。

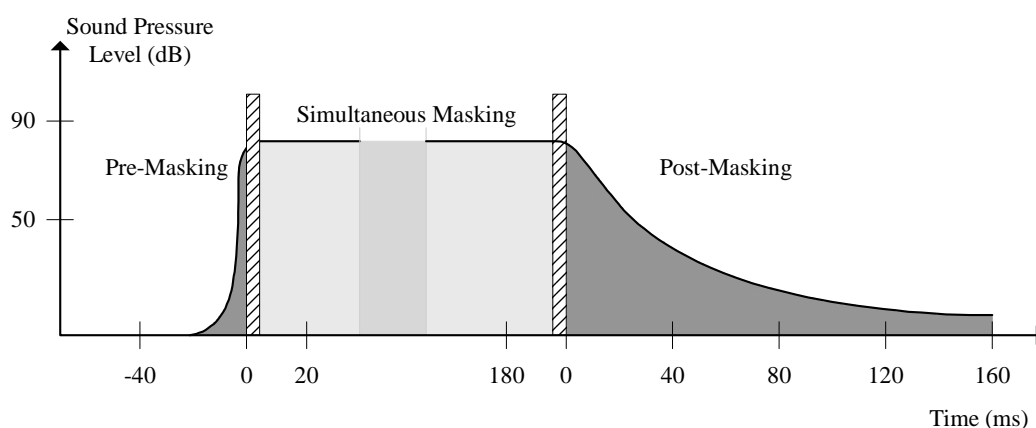


圖 2-11 時軸上的遮蔽效應

時軸上的遮蔽效應的重要之處在於能夠讓前迴音（pre-echo）的雜訊不被人耳察覺。先來說明前迴音產生的原因：若有一段振幅較小的聲音之後，突然出現了振幅較大的聲音，如圖 2-12(a)所示，此時會產生較大的量化誤差。如果在解碼過程中又經過反轉換的運算（Inverse Transform，如 IDCT），則這個量化誤差會再被放大，就產生了前迴音的現象，見圖 2-12(b)。在相同的情況下，如果所採用的轉換法的點數較少，就可以減少前迴音出現的時間，如圖 2-12(c)所示。

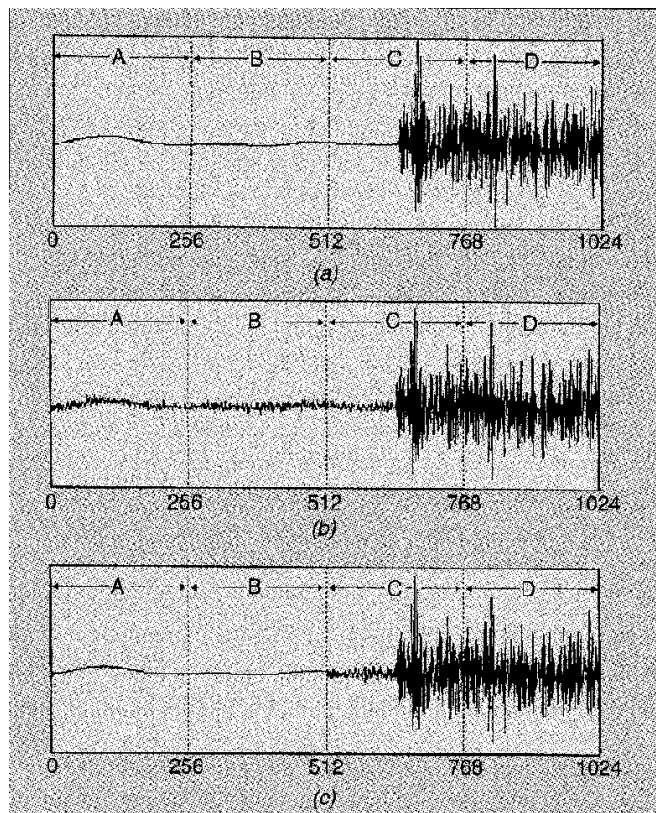


圖 2-12 前迴音與轉換編碼的關係[6]：

(a)會產生前迴音的聲音訊號

(b)經過 1024 點的轉換運算後出現的前迴音。

(c)經過 512 點的轉換運算後出現的前迴音。

2.2.5 Perceptual Entropy

Johnston 結合了聲響心理的遮蔽效應與量化的觀念定義了 Perceptual Entropy (PE)，用來評估在任何音樂中所包含與感知特性有關的訊息[4, 13]。對於特定的訊號，PE 能夠顯示其在理論上壓縮能力的限制。以下說明估計 PE 的方法。

首先聲音訊號必須先經過 Hann 窗框與傅力葉轉換對應到頻域上，然後算出每個頻線的能量，如 (2-13) 式。

$$P(\omega) = \text{Re}^2(\omega) + \text{Im}^2(\omega) \quad (2-13)$$

然後計算每個臨界頻帶頻線的能量總和如下：

$$B_i = \sum_{i=bl_i}^{bh_i} P(\omega) \quad (2-14)$$

其中 i 代表第 i 個臨界頻帶，而 bl_i 與 bh_i 分別代表第 i 個臨界頻帶的邊界。以下運算式中所提到的 i 皆代表臨界頻帶的編號。

為了將遮噪延展性之影響也考慮進來，(2-15) 式把各頻帶頻線能量總和 (2-14) 與展開方程式 (2-12) 相乘：

$$C_i = B_i \times SF_i \quad (2-15)$$

接下來估測 C_i 比較接近單頻 (tone-like) 或是雜訊 (noise-like)，(2-16) 式計算頻譜的平滑度：

$$SFM = \frac{\mu_g}{\mu_a} \quad (2-16)$$

其中 μ_g 與 μ_a 分別是各臨界頻帶 PSD (Power Spectrum Density) 的幾何平均數與算數平均數，所以 SFM 介於 0~1 之間。有了 SFM 之後，就可以找出單頻係數 α (coefficient of tonality)：

$$\alpha = \min\left(\frac{SFM_{dB}}{-60}, 1\right) \quad (2-17)$$

根據 \acute{a} 可以算出 offset 如下：

$$O_i = \acute{a} \times (14.5 + i) + (1 - \acute{a}) \times 5.5 \quad (\text{dB}) \quad (2-18)$$

(2-19) 式就算出各臨界頻帶的門檻值：

$$T_i = 10^{\log_{10} C_i - 0.1 \times O_i} \quad (2-19)$$

當然也要記得將靜音門檻曲線考慮進來，故

$$T_i = \max (T_i, T_q(i)) \quad (2-20)$$

最後可以由 (2-21) 式求出 PE：

$$PE = \sum_{i=1}^{25} \sum_{w=b_{l_i}}^{b_{h_i}} \left[\log_2 \left(n \operatorname{int} \left| \frac{\operatorname{Re}(\mathbf{w})}{\sqrt{6T_i / k_i}} \right| + 1 \right) + \log_2 \left(n \operatorname{int} \left| \frac{\operatorname{Im}(\mathbf{w})}{\sqrt{6T_i / k_i}} \right| + 1 \right) \right] \quad (2-21)$$

其中 $\operatorname{nint}()$ 是四捨五入的函數；PE 的單位是 bits/sample，代表每個取樣在維持 CD 音質的前提下，能夠壓縮到最低的位元數。

2.2.6 第二聲響心理模型

在介紹了聲響心理模型的一些原理之後，我們再來看看 MP3 所用到的第二聲響心理模型。第二聲響心理模型使用於第三編碼層時，並非如 2.2.2 節所說，將頻率範圍分為 23~26 個臨界頻帶，而是以 21 個比例因子頻帶（Scalefactor Bands）來取代。我們將這 21 個頻帶所對應的中心頻率與頻寬列於表 2-2。

Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	82	– 175	12	2732	2481 – 2983
2	251	175 – 328	13	3311	2983 – 3639
3	404	328 – 481	14	4065	3639 – 4491
4	557	481 – 634	15	4993	4491 – 5495
5	732	634 – 830	16	6053	5495 – 6610
6	907	830 – 983	17	7347	6610 – 8084
7	1098	983 – 1213	18	9017	8084 – 9950
8	1328	1213 – 1443	19	10929	9950 – 11907
9	1596	1443 – 1749	20	13070	11907 – 14233
10	1924	1749 – 2098	21	16678	14233 – 19124
11	2290	2098 – 2481			

表 2-2 適用 MP3 編碼的第二聲響心理模型之比例因子頻帶（44.1 kHz）

觀察表 2-2 可發現，第二聲響心理模型大約模擬人耳到頻率 19 kHz 左右，因為在 19 kHz 時的靜音門檻值已超過了 96dB（見圖 2-8），所以可以不需考慮頻譜上大於 19 kHz 的訊號。比較表 2-1 與表 2-2 可發現，比例因子頻帶在低頻時劃分得更細，但兩者都在越高頻時頻寬越寬。

雖然第二聲響心理模型最後是算出 21 個比例因子頻帶的 SMR 供位元分配與量化作參考，但是在計算過程中，卻將頻帶劃分得更細。以取樣頻率 44.1 kHz 的音訊而言，第二聲響心理模型將整個頻率的範圍劃分成 63 個區間來計算遮噪門檻值，最後再將這 63 個區間對應到 21 個比例因子頻帶求出 SMR。

另外為了使得前迴音不被人耳察覺，在 MP3 的編碼中採用了 MDCT 的方法。當偵測到前迴音的情況可能出現時，就採用短區塊的轉換，使得前迴音因為時軸上的遮蔽效應（pre-masking）而不被人耳察覺。以 44.1 kHz 取樣頻率的聲音來做說明：MP3 的一個編碼框包含 1152 個聲音訊號，相當於 26.1 ms 的時間長度。而參考圖 2-11，pre-masking 只涵蓋不到 20 ms 的時間，所以無法遮蔽前迴音。由於短區塊的 MDCT 長度僅為長區塊的三分之一，所以量化誤差會被放大的部分就只剩下原先三分之一的時間，即 8.7 ms。這樣一來，就能夠發揮 pre-masking 的效應，使得前迴音不被發覺[8]。

在 MP3 編碼中所用到的第二聲響心理模型中也有計算 PE 值，是用來決定 MDCT 使用長窗框或者短窗框。MP3 的定義中，當 PE 值大於 1800 時，選用短區塊的 MDCT 來處理該 granule（包含 576 個訊號）的子頻帶訊號。這是因為當 PE 值大於 1800 時，為了維持解碼後的音訊品質，需要使用較多的位元數來編碼。這表示此段音訊之變化較大，可能會產生前迴音，所以不適合用長區塊的 MDCT 來處理。

2.3 位元分配及量化 (Bit Allocation and Quantization)

2.3.1 位元分配 (Bit allocation)

位元分配的目的，在於使得每個比例因子頻帶之遮噪雜訊比 (MNR, Mask-to-Noise Ratio) 達到最大，以得到最佳之音訊品質。其為一反覆的過程，每次找出最小 MNR 的頻帶，分配位元給此頻帶以提高 MNR，接著重新計算各頻帶的 MNR，然後再不斷重覆此調整過程，直到沒有足夠的位元可供調整為止。因此在進行位元分配之前，需要知道可以用來編碼之位元數，以及各子頻帶的 MNR。

(2-22) 式是每個編碼框可用來編碼位元數的計算方法，其中 1152 是每個編碼框的取樣個數。以單聲道為例，在位元傳率為 128 kbps 且取樣頻率為 44.1kHz 時，每個編碼框有 3344 個編碼位元。

$$\text{可編碼之位元數} = \frac{\text{bitrate}}{\text{sampling freq.}/1152} \quad \text{bits/frame} \quad (2-22)$$

此外還要扣掉其它的資料所用到的位元數，如檔頭 (header) 需要 32 位元，而附屬資料 (side information) 需要 136 位元，另外可選擇是否要加入 16 位元的錯誤偵測碼 (CRC)。所以平均每個編碼框可用來對音樂訊號編碼的位元數為 $3344 - 32 - 136 - 16 = 3160$ bits / frame，又因為實作上編碼時的最小單位為 granule (一個編碼框相當於兩個 granules)，所以我們算出每個 granule 可用來編碼的位元數為 $3160/2=1580$ bits / granule。

MNR 可由 (2-11) 式求出，其中 SMR 是由聲響心理模型提供。有了 MNR 之後，就可以開始位元分配與量化。

MP3 編碼在量化時，將頻域劃分成 21 個比例因子頻帶 (scalefactor band) 為單位來處理。量化器對頻譜量化後，計算需要多少位元來對做霍夫曼編碼，若超過可用之位元數，就要調整 stepsize 的大小，如此可以降低所需的位元數。決定好量化後的頻線後，再和未量化前的頻線做雜訊估計，如 (2-23) 式所示。

$$distortion(band) = \sum_{low(band)}^{high(band)} \frac{(|xr(i)| - ix(i)^{4/3} * \sqrt[4]{2}^{stepsize})}{bandwidth(band)} \quad (2-23)$$

其中 $xr(i)$ 是經過 MDCT 轉換後所得到的頻線、 $ix(i)$ 是量化後的結果、 $bandwidth(band)$ 代表該頻帶的頻寬、 $low(band)$ 與 $high(band)$ 分別代表該頻帶的邊界，而 $ix(i)^{4/3} * \sqrt[4]{2}^{stepsize}$ 則是反量化的運算。

如果經由 (2-23) 式所求出該比例因子頻帶的失真超過可容忍的大小（由聲響心理模型提供），編碼器就會放大該頻帶內的值，這樣就可以使得該頻帶分到更多的位元，也就和前面所提到有關利用 MNR 來做位元分配的觀念呼應。如此一來，再對 $xr(i)$ 做量化的動作，重覆上述動作一直到沒有任何比例因子頻帶的誤差超過可容忍的失真大小即可。

2.3.2 非均匀量化 (Non-uniform quantization)

$$ix(i) = n \text{int}((\frac{|xr(i)|}{\sqrt[4]{2}^{stepsize}})^{0.75} - 0.0946) \quad (2-24)$$

(2-24) 式是 MP3 編碼的量化公式，其中 $xr(i)$ 是從 MDCT 輸出並調整過的頻線， $ix(i)$ 為量化過後的頻線（整數值），0.75 次方是用來使得量化器能提供一致的 SNR 值，而 $\text{nint}()$ 代表四捨五入的函數。另外， $stepsize$ 可以由 (2-25) 式求得，式中的 $system_const$ 為 8.0[1]。

$$stepsize = system_const \times \ln(sfm) \quad (2-25)$$

其中

$$sfm = \frac{e^{\frac{1}{n}(\sum_{i=0}^{575} \ln xr(i)^2)}}{\frac{1}{n} \sum_{i=0}^{575} xr(i)^2} \quad (2-26)$$

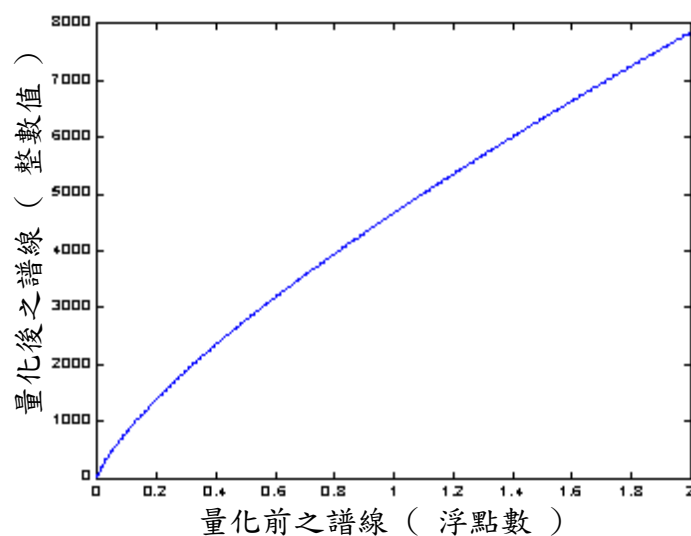


圖 2-13 量化器之輸入輸出關係圖

圖 2-13 是 (2-24) 式的輸入輸出關係圖，可以看出這是一個非均勻量化的量化器。0.75 次方是造成非均勻量化的原因，如果把 0.75 次方改成 1 次方，就會變為均勻量化。頻線經量化之後必須再做霍夫曼編碼，因為用來編碼的霍夫曼表所能表示的最大值為 8207，參考表 2-3，故輸出的量化後之頻線 $ix(i)$ 值不得大於 $8191+14=8205$ ，若超過就會產生溢位的現象，此時就要重新調整 stepsize，使其不產生溢位。而 (2-24) 式中的 stepsize 是以每次增加 1 的方式不斷的調整，因為 stepsize 愈負（通常 stepsize 為負值），輸出的頻線 $ix(i)$ 值就愈大，經由此一步驟可以得到適當的 stepsize。

2.3.3 霍夫曼編碼 (Huffman Encoding)

為了提高壓縮率，在量化之後又使用了無失真且非固定長度的霍夫曼編碼，將量化後的頻線 $ix(i)$ ($i=0\sim575$) 編碼。在量化後除了在 MDCT 使用短區塊的情況之外，量化器將輸出的頻線依頻率來排序。對於使用短區塊的情況而言，在相同的頻率下，又有三個窗框值，所以在每個比例因子頻帶裡其順序為頻率再來才是窗框。排序的優點在於可使得最大的值分佈在低頻，而在高頻時有一連頻的零 (zero values)，能夠提高壓縮比。

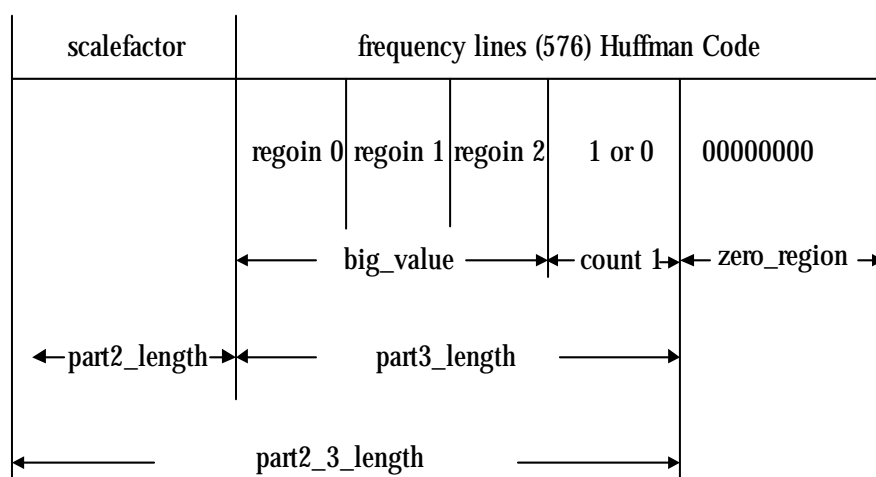


圖 2-14 主要資訊的組織[8]

而量化器將頻線 $ix(i)$ 分成三個區間，如圖 2-14 所示，這樣就可以讓編碼器依照不同的區間使用不同的霍夫曼表 (Huffman table)，而此霍夫曼表是利用用各個區間裡的統計特性而建立的。在高頻區間，編碼器將連續的一串零視為一個區間，稱為零區間 (zero region)，在此區間是不用編碼的，因為這個區間的長度的資訊可由另兩個區間的長度求之。而第二個區間稱為 count1 區間，是由一連串的 0 和 1 組成，此區間的霍夫曼編碼方式是以四個取樣為一組來查表，所以此區間的長度為 4 的倍數。第三個區間為 big_value 區間，此區間會出現較大的數

值，這裡所用的霍夫曼表是以兩個量化後的頻線為一組來編碼，而此區間可再細分成三個區間，每個區間的霍夫曼表不一定相同，端看各區間的最大數值來決定合適的霍夫曼表。

索 引	最大值	索 引	可編碼之最大值	linbits
0	0	16	16	1
1	1	17	19	2
2	2	18	23	3
3	2	19	31	4
4	not used	20	79	6
5	3	21	271	8
6	3	22	1039	10
7	5	23	8207	13
8	5	24	31	4
9	5	25	41	5
10	7	26	79	6
11	7	27	143	7
12	7	28	271	8
13	15	29	527	9
14	not used	30	2016	11
15	15	31	8207	13

表 2-3 big_value 區間所用 32 個霍夫曼表的特性[8]

霍夫曼表共有 34 個，其中有 2 個表是給 count1 區間使用，是以 4 條頻線為一組的方式來編碼，因為此區域的頻線是由 0 或者是 1 組成，所以共有 16 種不同的組合。而另外的 32 個表是給 big_value 區間用的，其特性如表 2-3，這 32 個霍夫曼表是以 2 條頻線為一組來編碼的。big_value 區間可再分為三個子區間，每個子區間擁有各自的霍夫曼表。見表 2-3，霍夫曼表 0~15 的最大值不超過 15，所以只能用來對最大值不超過 15 的子區間編碼，而霍夫曼表 15~31 的最大值皆為 15，如果要對值超過 15 的頻線編碼，就要使用 linbits 的方法，如 (2-27) 式。

$$\text{ESCAPE} = \text{量化過後的值} - 15$$

$$\text{可編碼之最大值} = 15 + 2^{\text{linbits}} \quad (2-27)$$

2.4 位元串格式 (Bitstream Formatting)

圖 2-15 是 MP3 的位元串格式，共分為四個部分，分別是檔頭 (header)、錯誤偵測碼 (CRC)、附屬資料 (side information) 和主要資訊 (main data)。檔頭和附屬資料記載解碼時所需的資訊，其中單聲道的音訊必須包含 136 位元的附屬資料，雙聲道的音訊則需要 256 位元。在檔頭區的第 16 個位元會記錄是否使用錯誤偵測碼，若有，則在檔頭區之後會接著 16 位元的偵測碼。錯誤偵測碼是用來對檔頭在解碼時做偵錯的動作，避免因為檔頭出現錯誤而無法正確的解碼資料；主要資料區是存放著比例因子 (scalefactor) 和經過量化、位元分配與無失真的霍夫曼編碼之後的音樂訊號。

檔 頭 區 (32 bits)	錯誤偵測碼 (0, 16 bits)	附 屬 資 料 (136, 256 bits)	主 要 資 訊 (編碼過後的訊號)
--------------------	-----------------------	----------------------------	----------------------

圖 2-15 MPEG-1 layer3 的資料格式[3]

在 MPEG-1/Audio 的第一、二層編碼中，當位元傳率決定了之後，每個編碼框可用來編碼的位元預算也就固定了。但在實際的情況下，每個編碼框所包含的音訊都有不同的特性，若想要維持一定的音訊品質，則所含音訊變化較大的編碼框勢必需要較多的位元來編碼。所以在第三層編碼中，使用了位元儲藏處的機制來對包含不同音訊特性的編碼框做一適當的調節。2.4.1 節就來說明 MP3 的位元儲藏機制。

2.4.1 位元儲藏處 (bit reservoir)

因為每個編碼框所包含的音訊特性不盡相同，所以為了維持一定的聲音品質，每個編碼框所需的編碼位元數也不同。於是 MP3 使用了一種位元儲藏處的機制來做調節，有了這個機制，當一個編碼框實際編碼所需的位元少於位元預算，編碼器就可以將多出來的位元預算存到位元儲藏處；反之若編碼後超過允許的位元數，則可經由位元儲藏處對過去的編碼框借位元來用。每個編碼框都有一個 9 位元的暫存器，用來記錄每個編碼框主要資訊開始的位置。此暫存器稱為 `main_data_begin`，存放於每個編碼框的附屬資料內。這樣一來就可以動態地決定每個編碼框的開端。

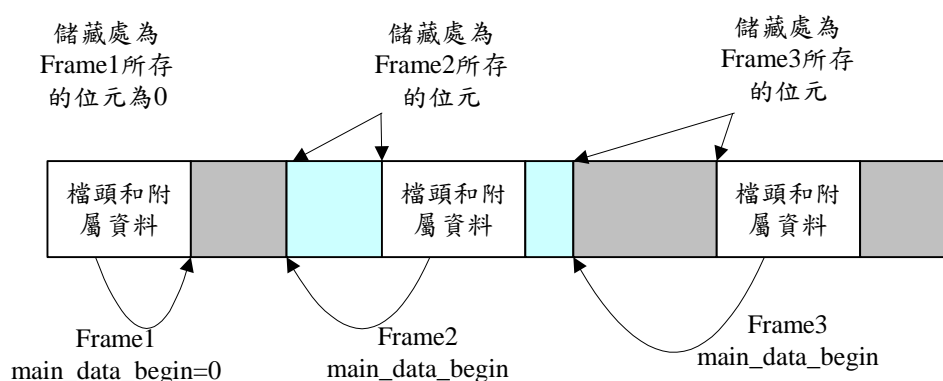


圖 2-16 MP3 的位元流 (bitstream) 與儲藏機制示意圖[3]

如圖 2-16 所示，每個編碼框的檔頭間的間距相同，這表示檔頭是在位元流固定的位置上，但是每個編碼框的資料部份卻不一定在固定的位置上，因為 MP3 利用位元儲藏處的機制，使得圖 2-16 中的 frame2，因為在位元儲藏處有存下 frame1 未用完的位元空間，所以 frame2 的指標 `main_data_begin` 指向 frame1 未用完的區域繼續使用，直到 frame2 編碼完後，再將多餘的位元，留給後來的編碼框使用。

2.4.2 MPEG-1/Audio 的檔頭格式

MPEG-1/Audio 各編碼層的檔頭格式是相同的，如圖 2-17 所示。以下對這 32 位元的檔頭所包含的資訊加以說明，其含義如下：

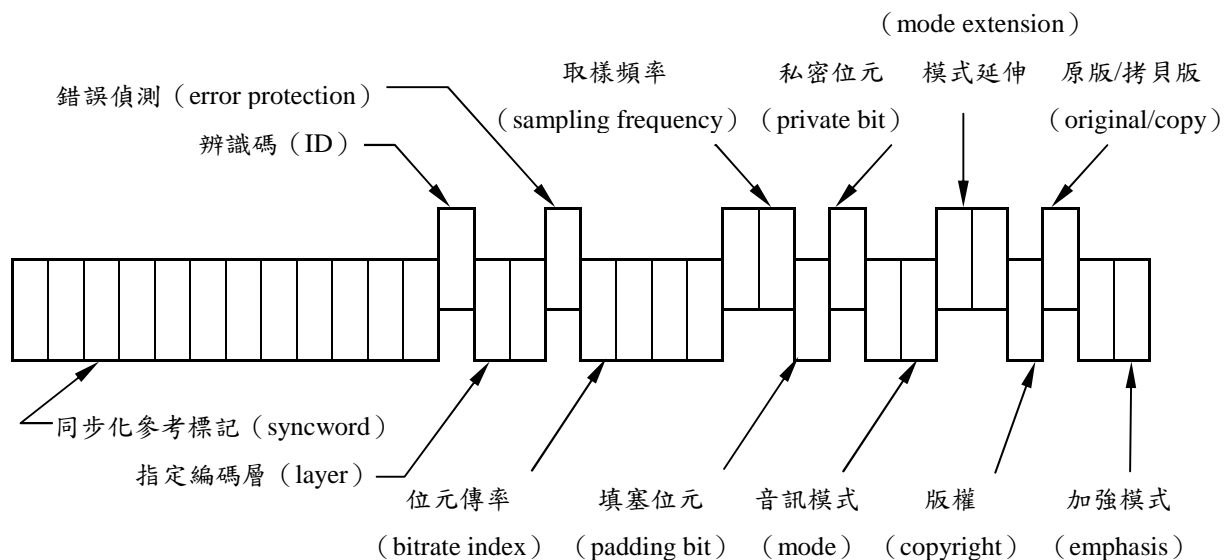


圖 2-17 MPEG-1/Audio 檔頭區的位元串格式（共 32 位元）

- 同步化參考標記 (syncword)：用來標記一首歌之開頭，用連續 12 個 "1" 來表示。
- 辨識碼 (ID)：以 1 個位元 "1" 表示為 MPEG/Audio。
- 編碼層 (layer)：以 2 個位元來表示是使用哪一層編碼，"11" 表示為第一層編碼，"10" 表示第二層編碼，"01" 則表示為第三層編碼。
- 錯誤偵測 (protection_bit)：以 1 個位元來表示是否要加上額外的 16 位元的錯誤偵測碼，可提供檔頭的錯誤偵測。
- 位元傳率指標 (bit_rate_index)：利用 4 個位元來指定不同的位元傳率，可經由表 2-4 得之。

Bitrate Index	Bitrate (kbps)					
	Layer I		Layer II		Layer III	
	free	format	free	format	free	format
'0000'						
'0001'	32		32		32	
'0010'	64		48		40	
'0011'	96		56		48	
'0100'	128		64		56	
'0101'	160		80		64	
'0110'	192		96		80	
'0111'	224		112		96	
'1000'	256		128		112	
'1001'	288		160		128	
'1010'	320		192		160	
'1011'	352		224		192	
'1100'	384		256		224	
'1101'	416		320		256	
'1110'	448		384		320	

表 2-4 位元傳率對照表[1]

- 取樣頻率 (sampling_frequency)：指出這首歌曲是用何種取樣頻率， ”00” 表 44.1 kHz， ”01” 表 48k Hz， ”10” 表 32k Hz。
- 填塞位元 (padding_bit)：使得每個編碼框可用來對音樂訊號編碼的位元數為 8 的倍數，利用進位或捨去來使其可以位元組 (bytes) 為單位。只有在取樣頻率為 44.1k Hz 時要用到填塞位元。
- 音訊模式 (mode)： ”00” 表立體聲 (stereo)， ”01” 表雙聲合一 (joint_stereo)， ”10” 表對偶聲式 (dual_channel)， ”11” 表單聲式 (single_channel)。
- 模式延伸 (mode_extension)：在第三層編碼中用來標記使用何種雙聲合一 (joint_stereo) 的模式，見表 2-5。

- 版權（copyright）：”0”表此位元串沒有版權的限制，”1” 此位元串表有版權的限制。
- 原版/拷貝版（original /home）：”0”此位元串是拷貝過的，”1”表此位元串是原版的。

	intensity_stereo	ms_stereo
'00'	off	off
'01'	on	off
'10'	off	on
'11'	on	on

表 2-5 模式延伸對照表

第三章 MP3 編碼實作

前一章大致說明了 MP3 編碼的原理與方法，但考量實際的狀況，我們希望在 DSP 上實現 MP3 編碼器時，能夠盡量減少編碼時間並且降低記憶體空間的使用。由於我們選擇了 TI C5416 DSP[7, 14]來實現 MP3 編碼器，所以在撰寫程式時也必須配合這顆 DSP 的特性。因此為了增進整體的編碼效能，在實現 MP3 時做了許多的考量與調整，包含了定點數的運算與 MP3 編碼器的各個方塊之化簡。本章將針對 MP3 編碼實作的方法與改進的部分加以說明。

3.1 定點數運算

TMS320C5416 是一顆 160 MHz、16 位元定點運算的 DSP。雖然它也能夠支援浮點的運算，不過相對的也需要較多的運算時間。因此為了能達到較好的效能，在撰寫程式時採用定點數運算的方式。

在使用定點數運算時，原先所使用的一些數學函數，如正弦、餘弦、對數和次方等函數皆無法使用。所以首要的課題就是如何以其他的方式來取代些函數。

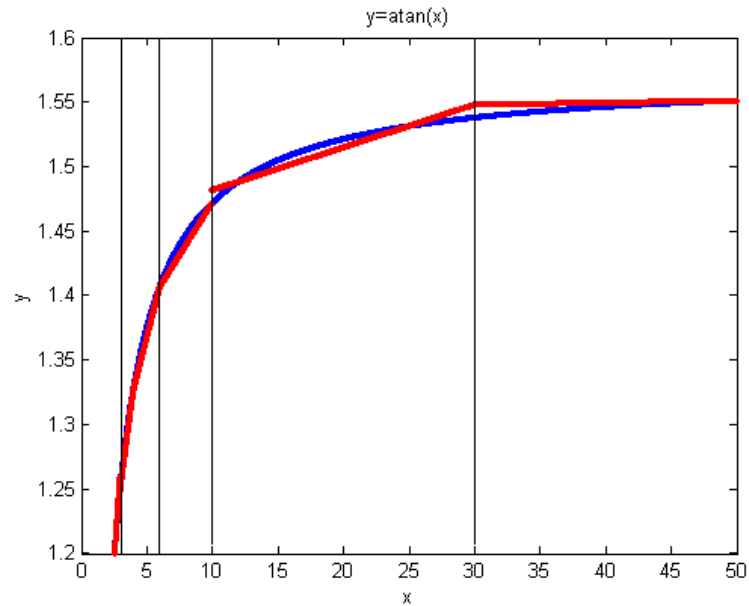
在 MP3 的編碼過程中，有許多的運算都與正弦、餘弦函數有關，主要是濾波器與窗框的係數，包含了多重相位濾波器的係數、MDCT 窗框與 MDCT 的係數、Hann 窗框係數、FFT 的係數等。這些係數因為已經固定，所以可以事先求得這些係數，將其放在記憶體中，運算時就用查表的方式找到所需的係數。每個係數皆以 16 位元表示，若這些係數全部存放在記憶體中將會耗費不少的記憶體空間，所以在實作時也有簡化一些係數，在 3.2~3.4 小節將會提到。

另外，對於指數（exponent）、對數（logarithm）與反正切（arc-tangent）函數是採用分區間直線逼近的方式。在可容許的誤差下，在會使用到的範圍內，訂出數個區間，每個區間使用一條直線來逼近這個區間內的對數值。上述方法相對於使用泰勒展開式或查表法，不但計算方法簡單（只需一次乘加），而且又不會耗費太多的記憶體空間（在一個區間內只需兩個係數）。以反正切函數為例說明之。

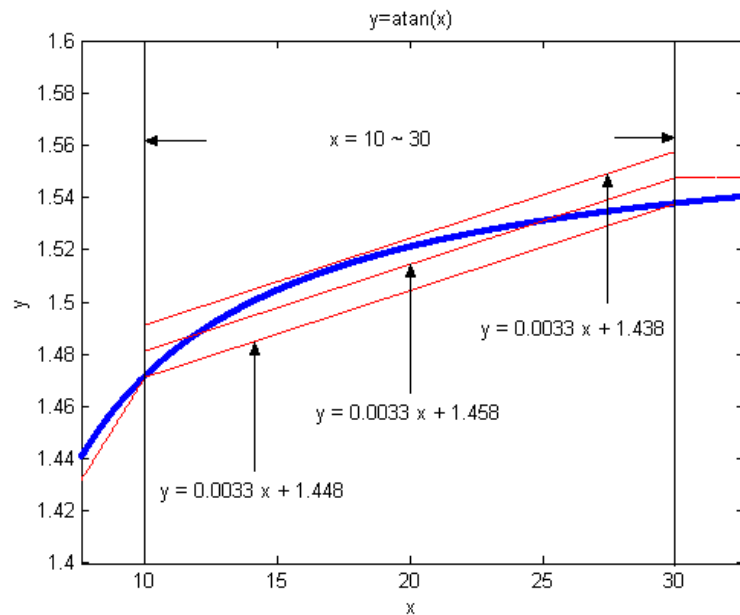
見圖 3-1(a)，在 $y = a \tan(x)$ 函數中，我們將 x 分為數個區間，每個區間都用一條直線方程式來逼近 y 的值。圖 3-1(b) 放大了 $x=10\sim30$ 的區間，在此區間中，我們先利用反正切函數的兩個端點決定直線的斜率，然後取三個不同的 offset(個數視情況而定)，找出圖上的三條直線。然後分別計算這三條直線與反正切函數的最大誤差值，取三者中最小值即可。圖 3-1(b) 中，以中間的直線誤差最小，其最大誤差為 0.68%。

最後則是次方的問題。當指數部分不是整數時，意味著有根號的運算。在 MP3 的編碼中，主要是在量化時會使用到 0.75 次方，見 (2-24) 式。由於此式

較為複雜，也非常耗費計算時間，所以我們將此式與其他的運算合併化簡後，除了採用上述分區間直線逼近的方式之外，也配合了其他更合適的方法來解決這個問題，在 3.5 節中會有更進一步的討論。



(a) 將 x 分為數個區間以直線逼近



(b) $x=10\sim30$ 的區間

圖 3-1 反正切函數的逼近方法

3.2 分析子頻帶濾波器

每當有 32 個 PCM 訊號輸入到分析子頻帶濾波器，經過運算後各個濾波器分別可以得到一個輸出，也就是說頻率變為原先的 32 分之一。若以一個編碼框為單位，則每個子頻帶皆可得到 36 條頻線。

在開始之前首先需要一個能存放 512 個 PCM 訊號的暫存器 $X[n]$, $n=0\sim511$ ，如圖 3-3 所示。 n 越小， $X[n]$ 所存放的 PCM 訊號就越新。以下是實際運算的流程（圖 3-2）：

1. 暫存器中的資料往右移動 32 個位置，將原本存放在 $X_{480} \sim X_{511}$ 的訊號移出暫存器，並空出 $X_0 \sim X_{31}$ 的位置。
2. 將最新輸入的 32 個 PCM 訊號放入 $X_0 \sim X_{31}$ ，從 X_{31} 這個位置開始放起，最新的一個 PCM 訊號放在 X_0 的位置。
3. 將 $X[n]$ 與窗框係數 $C[n]$ 相乘，得到 $Z[n]$, $n=0\sim511$ 。窗框係數 $C[n]$ 定義在 [1] 中 Table 3-C.1。
4. 將每間隔 64 個位置的 $Z[n]$ 累加起來，得到 $Y[k]$, $k=0\sim63$ 。
5. 最後只要讓 $Y[k]$ 與矩陣 $M[i][k]$ 相乘即可在每個子頻帶得到一個結果 $S[i]$ 。

$$M[i][k] = \cos\left[\frac{\pi(2i+1)(k-16)}{64}\right] \quad \text{其中,}$$

$$i=0\sim31, k=0\sim63。$$

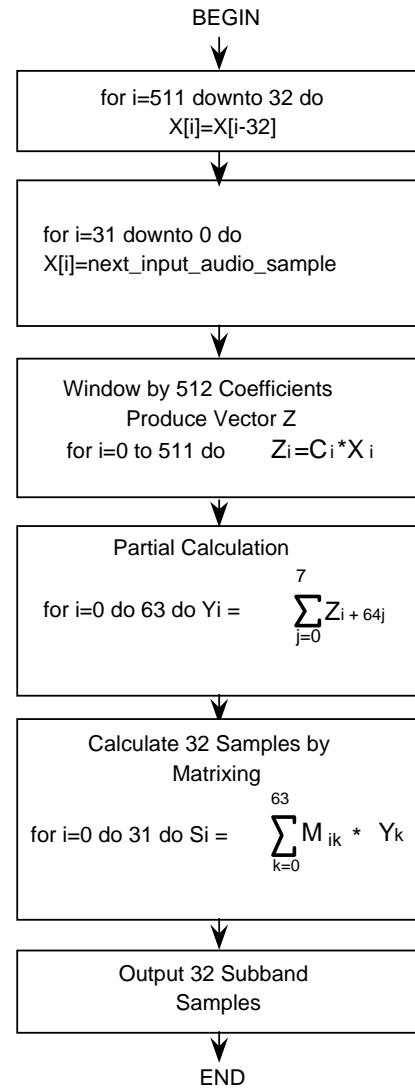


圖 3-2 分析子頻帶濾波器運算流程圖[1]

$X_0 \sim X_{31}$	$X_{32} \sim X_{63}$	$X_{64} \sim X_{95}$	$\cdot \cdot \cdot$	$X_{448} \sim X_{479}$	$X_{480} \sim X_{511}$
-------------------	----------------------	----------------------	---------------------	------------------------	------------------------

圖 3-3 暫存器 \underline{X}_{512}

觀察這 32 個濾波器的係數，即矩陣 $\underline{M}_{32 \times 64}$ [1]，發現它們有著一些相同的特性——對稱，利用這些特性可以降低運算量並省下一些記憶體空間。

$$\underline{M}_{32 \times 64} : M[i][k] = \cos\left[\frac{\pi}{64}(2i+1)(k-16)\right], \quad i = 0 \sim 31, \quad k = 0 \sim 63. \quad (3-1)$$

我們從兩個方向找到化簡濾波器係數 $\underline{M}_{32 \times 64}$ 的方法：其一，32 個濾波器的係數有著相同的對稱方式。其二，32 個濾波器彼此間的關係。以下將一一說明：

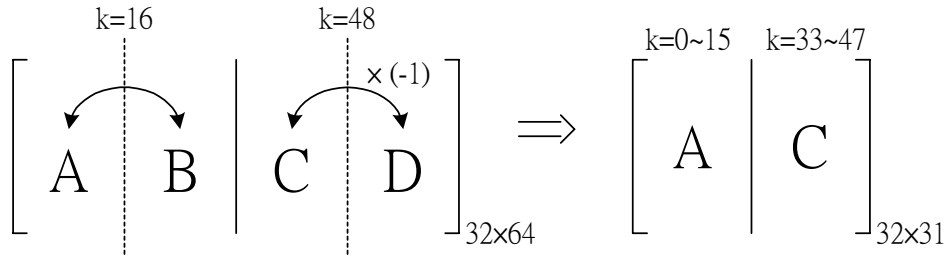


圖 3-4 $\underline{M}_{32 \times 64}$ 的大小由 32×64 縮減成 32×31

首先，對 32 個濾波器而言，第 16 個係數皆為 1，第 48 個係數皆為 0。第 0~32 個係數以第 16 個係數為中心左右對稱；第 33~63 個係數以第 48 個係數為中心左右差一負號對稱，如圖 3-4 所示。將 $\underline{M}_{32 \times 64}$ 經過上述規則化簡所得到的矩陣稱為 $\underline{M}'_{32 \times 31}$ ，數學式歸納於下：

$$S[i] = \sum_{k=0}^{63} M[i][k] \times Y[k] \quad \text{for } i = 0 \sim 31 \quad (3-2)$$

$$\begin{aligned} &= Y[16] + \sum_{k=0}^{15} M[i][k] \times (Y[k] + Y[32-k]) + \sum_{k=33}^{47} M[i][k] \times (Y[k] - Y[96-k]) \\ &= Y[16] + \sum_{j=0}^{15} M'[i][j] \times (Y[j] + Y[32-j]) + \sum_{j=16}^{30} M'[i][j] \times (Y[j+17] - Y[79-j]) \end{aligned} \quad (3-3)$$

對每一個濾波器而言，原本需要 64 個乘法與 63 個加法（3-2）式才能完成運算，經過上述的化簡後，只剩下 31 個乘法與 62 個加法（3-3）式。而矩陣的大小也從原本的 32×64 變成 32×31 。

接下來，再觀察 $\underline{M}'_{32 \times 31}$ 中各濾波器之間的關係：

$$M'[i][j] = |M'[31-i][j]|, \text{ for } i=0 \sim 15, j=0 \sim 31. \quad (3-4)$$

其中

$$M'[i][j] = M'[31-i][j], \text{ for } j = 0, 2, 4, 6, 8, 10, 12, 14, 17, 19, 21, 23, 25, 27, 29.$$

$$M'[i][j] = -M'[31-i][j], \text{ for } j = 1, 3, 5, 7, 9, 11, 13, 15, 16, 18, 20, 22, 24, 26, 28, 30.$$

根據上述特性，我們定義一個新的矩陣 $\underline{M}''_{16 \times 31}$ （如圖 3-5 所示）與四個集合 J1~J4：

$$\underline{M}''_{16 \times 31} : M''[n][j] = M'[n][j], \text{ for } n=0 \sim 15, j=0 \sim 30.$$

$$J1 = \{j | 0, 2, 4, 6, 8, 10, 12, 14\}, J2 = \{j | 1, 3, 5, 7, 9, 11, 13, 15\},$$

$$J3 = \{j | 17, 19, 21, 23, 25, 27, 29\}, J4 = \{j | 16, 18, 20, 22, 24, 26, 28, 30\}.$$

接著，從（3-3）式再繼續化簡：

$$S[i] = Y[16] + \sum_{j=0}^{15} M'[i][j] \times (Y[j] + Y[32-j]) + \sum_{j=16}^{30} M'[i][j] \times (Y[j+17] - Y[79-j])$$

$$= \begin{cases} Y[16] + \sum_{J1} M'[i][j] \times (Y[j] + Y[32-j]) + \sum_{J2} M'[i][j] \times (Y[j] + Y[32-j]) \\ + \sum_{J3} M'[i][j] \times (Y[j+17] - Y[79-j]) + \sum_{J4} M'[i][j] \times (Y[j+17] - Y[79-j]) \\ \text{for } i = 0 \sim 15 \end{cases} \quad (3-5)$$

$$= \begin{cases} Y[16] + \sum_{J1} M'[31-i][j] \times (Y[j] + Y[32-j]) - \sum_{J2} M'[31-i][j] \times (Y[j] + Y[32-j]) \\ - \sum_{J3} M'[31-i][j] \times (Y[j+17] - Y[79-j]) + \sum_{J4} M'[31-i][j] \times (Y[j+17] - Y[79-j]) \\ \text{for } i = 16 \sim 30 \end{cases} \quad (3-6)$$

然後令

$$T_1[n] = Y[16] + \sum_{j_1} M''[n][j] \times (Y[j] + Y[32 - j]) + \sum_{j_4} M''[n][j] \times (Y[j + 17] - Y[79 - j])$$

for $n = 0 \sim 15$ (3-7)

$$T_2[n] = \sum_{j_2} M''[n][j] \times (Y[j] + Y[32 - j]) + \sum_{j_3} M''[n][j] \times (Y[j + 17] - Y[79 - j])$$

for $n = 0 \sim 15$ (3-8)

最後可以得到 \underline{S}_{32}

$$\begin{cases} S[n] = T_1[n] + T_2[n] \\ S[31 - n] = T_1[n] - T_2[n] \end{cases} \quad \text{for } n = 0 \sim 15 \quad (3-9)$$

根據 (3-2) 式，原本需要 $64 \times 32 = 2048$ 個乘法與 $63 \times 32 = 2016$ 個加法才能得到 $\underline{M}_{32 \times 64} \times \underline{Y}_{64 \times 1}$ 的結果。而化簡成 (3-7) 式~(3-9) 式之後，則只需要 $31 \times 16 = 496$ 個乘法與 $61 \times 16 + 32 = 1008$ 個加法就能算出 $\underline{S}_{32 \times 1}$ 。

另外，為了不增加 DSP 的負擔，我們並非直接呼叫 cos 函數來產生 $\underline{M}_{32 \times 64}$ 的數值，而是採用查表的方式，事先將 $\underline{M}_{32 \times 64}$ 存放在記憶體中。因此經過化簡後，所使用的記憶體空間也跟著變少了。原本需要 $32 \times 64 = 2048$ words，最後後只剩下 $16 \times 31 = 496$ words。

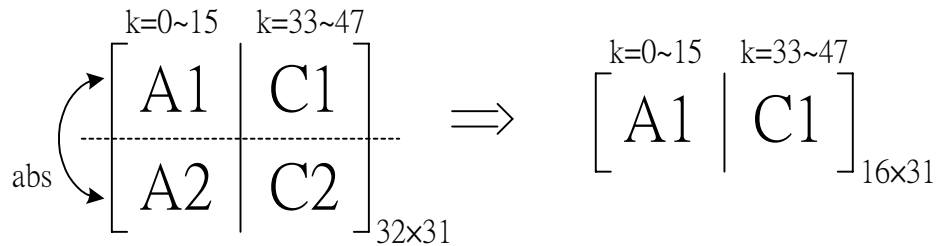


圖 3-5 $\underline{M}'_{32 \times 31}$ 的大小由 32×31 縮減成 16×31 ， $\underline{M}''_{16 \times 31}$

3.3 改良式餘弦轉換

以 granule 為單位來看，經過濾波器排分析後，每個子頻帶皆有 18 條頻線。將前一個 granule 與目前 granule 加起來的 36 條頻線組合起來，記為 $x'_0 \sim x'_{35}$ ，經過選定的窗框後，得到結果 $z_0 \sim z_{35}$ 。過程與四種窗框的係數如下：

$$\text{長窗框} \quad z_i = x'_i \sin\left[\frac{P}{36}\left(i + \frac{1}{2}\right)\right] \quad \text{for } i = 0 \sim 35 \quad (3-10)$$

$$\text{長短窗框} \quad z_i = \begin{cases} x'_i \sin\left[\frac{P}{36}\left(i + \frac{1}{2}\right)\right] & \text{for } i = 0 \sim 17 \\ x'_i & \text{for } i = 18 \sim 23 \\ x'_i \sin\left[\frac{P}{12}\left(i - 18 + \frac{1}{2}\right)\right] & \text{for } i = 24 \sim 29 \\ 0 & \text{for } i = 30 \sim 35 \end{cases} \quad (3-11)$$

$$\text{短長窗框} \quad z_i = \begin{cases} 0 & \text{for } i = 0 \sim 5 \\ x'_i \sin\left[\frac{P}{12}\left(i - 6 + \frac{1}{2}\right)\right] & \text{for } i = 6 \sim 11 \\ x'_i & \text{for } i = 12 \sim 17 \\ x'_i \sin\left[\frac{P}{36}\left(i + \frac{1}{2}\right)\right] & \text{for } i = 18 \sim 35 \end{cases} \quad (3-12)$$

$$\text{短窗框} \quad \begin{cases} y_i^{(0)} = x'_{i+6} & \text{for } i = 0 \sim 11 \\ y_i^{(1)} = x'_{i+12} & \text{for } i = 0 \sim 11 \\ y_i^{(2)} = x'_{i+18} & \text{for } i = 0 \sim 11 \end{cases}$$

$$z_i^{(k)} = y_i^{(k)} \sin\left[\frac{P}{12}\left(i + \frac{1}{2}\right)\right] \quad \text{for } i = 0 \sim 11, k = 0 \sim 2 \quad (3-13)$$

然後就做 MDCT 運算，

$$x_i = \sum_{k=0}^{n-1} z_k \cos\left[\frac{P}{2n}\left(2k + 1 + \frac{n}{2}\right)\left(2i + 1\right)\right] \quad \text{for } i = 0, 1, \dots, \frac{n}{2} - 1. \quad (3-14)$$

其中若此處選擇短窗框則 $n=12$ ，否則 $n=36$ 。

若非選擇短窗框，最後還要記得加上假象處理。方法如圖 2-7 所示，其中 cs 與 ca 兩組係數是由 (3-15) 式而來，至於 c_i 則列於表 3-1。

$$\begin{cases} cs_i = \frac{1}{\sqrt{1+c_i^2}} \\ ca_i = \frac{c_i}{\sqrt{1+c_i^2}} \end{cases} \quad (3-15)$$

對於 MDCT 在實作上我們採用 Fast MDCT 的方式，主要的方法是找出 (3-14) 式中餘弦係數的規則。經過數值的分析可以發現 MDCT 的係數與 (3-1) 式的 $\underline{M}_{32 \times 64}$ 有相似的對稱方式，因此可以用類似過程來降低運算量。

i	Ci
0	-0.6
1	-0.535
2	-0.33
3	-0.185
4	-0.095
5	-0.041
6	-0.0142
7	-0.0037

表 3-1 假象處理所用的係數 c_i

3.4 第二聲響心理模型

我們在 MP3 的編碼中採用了第二聲響心理模型，而這個模型必須加以修改，才能符合第三編碼層使用，運算流程如圖 3-6 所示。對第三編碼層的每一編碼框而言，總共要經過兩次聲響心理模型之運算(每個 granule 一次)。因為 MDCT 有長區塊與短區塊之選擇，所以聲響心理模型也包含了長區塊與短區塊的計算。

參考圖 3-6，首先，輸入的 PCM 訊號必須先經過 Hann Window 與相同點數的快速傅利葉轉換 (FFT)。其中長區塊的部分是每次運算時移動 576 個 PCM 訊號，將 PCM 訊號置於窗框中央，經過 1024 點的 Hann Window 與 FFT；而對短區塊而言，每次移動 192 個 PCM 訊號，將 PCM 訊號置於窗框中央，重複三次 256 點的運算。

接著，將 FFT 運算的結果以 $r-\theta$ 的方式表示，根據[1]中的定義，利用這些資訊計算出不可預測之值 (unpredictability measure)。我們必須先分別算出長區塊與短區塊的不可預測值，然後再求出整體的不可預測值。

在計算 threshold 之前，我們要先準備好展開函數，這個函數的計算與輸入音訊的取樣頻率有關，而與聲音本身無關，所以這個函數是可以先準備好的。展開函數的計算方式在[1]中有說明，大致上與 (2-11) 式相同，只是式中與臨界頻帶距離有關的變數 x ，在不同的編碼層有不同的計算方式。因為對於相同的取樣頻率而言，展開函數都相同，故只需計算一次之後記錄下來即可，不需每次做聲響心理模型時都計算一次。

接著就來計算長區塊的 threshold (前半段)，見圖 3-6(b)。此處之所以只做前半段的運算是為了要先求得 perceptual entropy (以下簡稱 pe 值)，當算出 pe 值後，依照 pe 值的大小才決定最後要提供長區塊或是短區塊的 SMR (ratio) 給位元分配與量化做參考，所以長區塊後半部的 threshold 計算留到最後有需要時才做。參考[1]與 2.2.5 節所描述 PE 的計算方式，可以發現兩者十分雷同。

算出 pe 值之後就可以決定 MDCT 要選用長窗框或是短窗框了。當 pe 值小

於 1800，MDCT 就選擇長窗框，聲響心理模型就繼續完成長區塊 threshold 的後半段運算（圖 3-6(c)），這裡就只剩下 SMR 的計算了。之前有提到，通常聲響心理模型會根據人耳聽覺感知的範圍與特性，劃分為 23~26 個臨界頻帶。但在這裡，第二聲響心理模型應用在第三層編碼上的劃分方式略有不同，它是分成 21 個比例因子頻帶（scalefactor band）。而最後聲響心理模型會計算出對應到這 21 個頻帶的 ratio 值（即 SMR），作為量化的參考。

若 pe 值大於 1800，則 MDCT 選擇短窗框，聲響心理模型就必須計算短區塊的 threshold，其計算方式與長區塊的 threshold 算法（前半段加上後半段）雷同，只要把長區塊的參數換成短區塊的參數即可。另外短區塊的 SNR 值不需計算，只要根據取樣頻率查表即可。以上就是第二聲響心理模型的實際計算方式。

在聲響心理模型中，我們對於展開函數相關的計算做了一些化簡的工作。以長區塊為例加以說明：

第一，由於展開函數的計算方式牽涉到次方與根號，為了減低 DSP 的負擔，我們決定事先將展開函數的數值算出，存放在記憶體中。假設以 $SF(x, y)$ 表示展開函數，其中 x 與 y 的值為 0~62，也就是在頻域上分成 63 個區間來計算各區間彼此之間的影響。

第二，觀察展開函數可以發現，當 $x=y$ 時， $SF(x, y)$ 的值為 1，然後不論是往高頻或低頻的方向 $SF(x, y)$ 的值都會遞減，如圖 3-7 所示。這是因為距離越近的頻帶之間原本就比較容易互相影響。但我們發現其影響範圍不會超過 17 個區間，也就是說，對於某個固定的 x ，63 個 $SF(x, y)$ 數值中至少有 46 個數值為 0，並且有一數值必為 1。根據上述結果，我們不但可以將 $SF(x, y)$ 所佔的記憶體空間從 $63 \times 63 = 3969$ words（16-bit）降低到 $63 \times 16 = 1008$ words，而且與展開函數相關的運算（相乘與累加）也同時減少。

而短區塊與長區塊的情況相同，用來存放短區塊展開函數的記憶體空間亦從 $39 \times 39 = 1521$ words 下降到 $39 \times 10 = 390$ words。

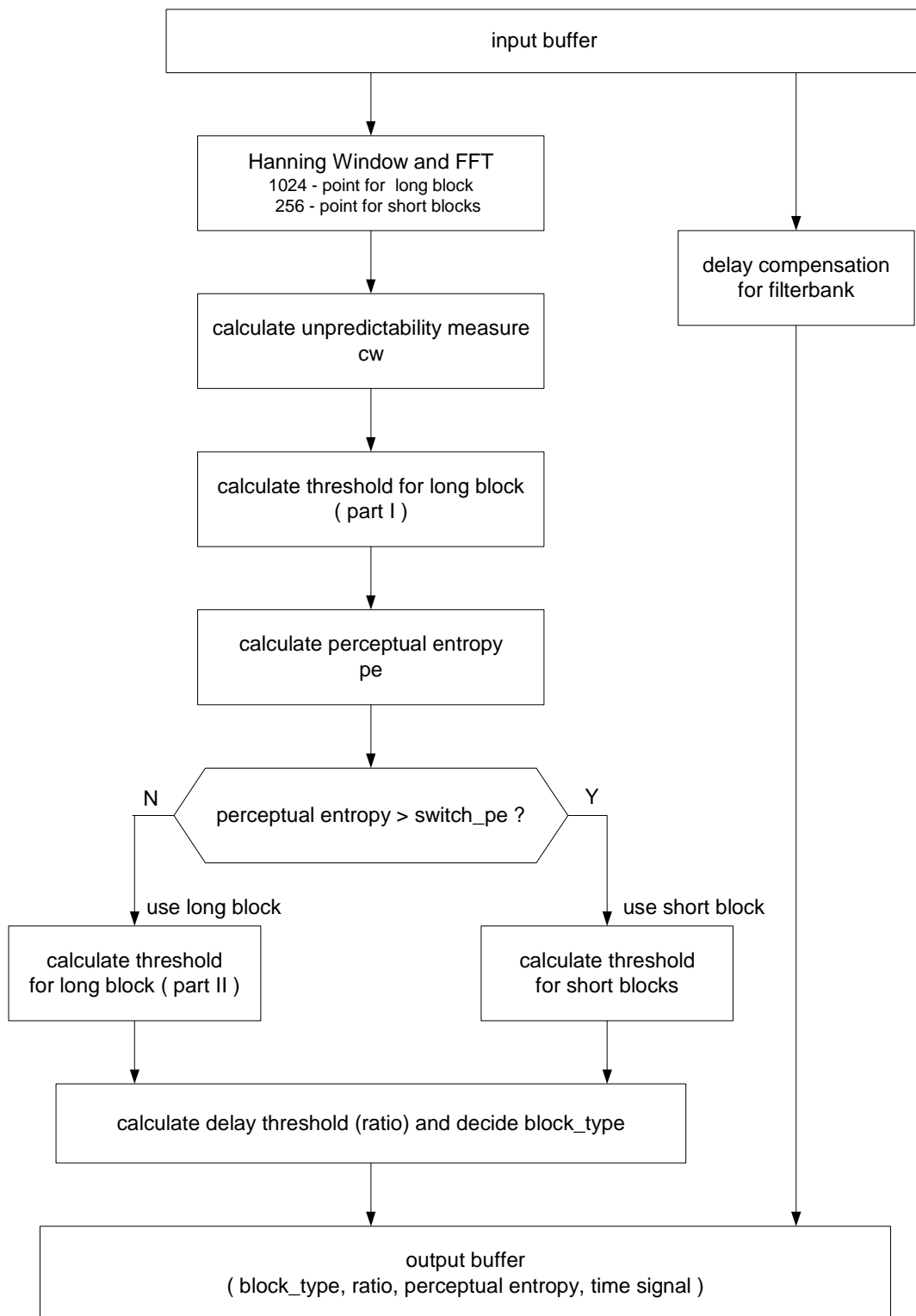


圖 3-6 (a)適用於第三編碼層的第二聲響心理模型流程圖[1]

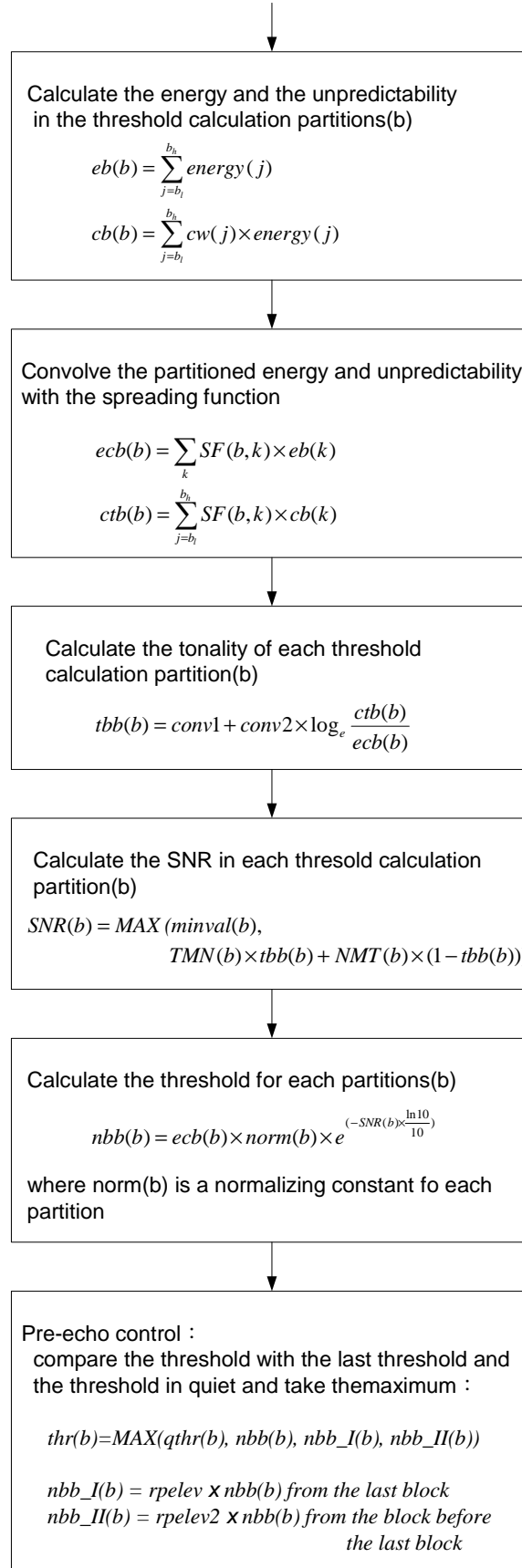


圖 3-6 (b) calculate threshold for long block (part I) [1]

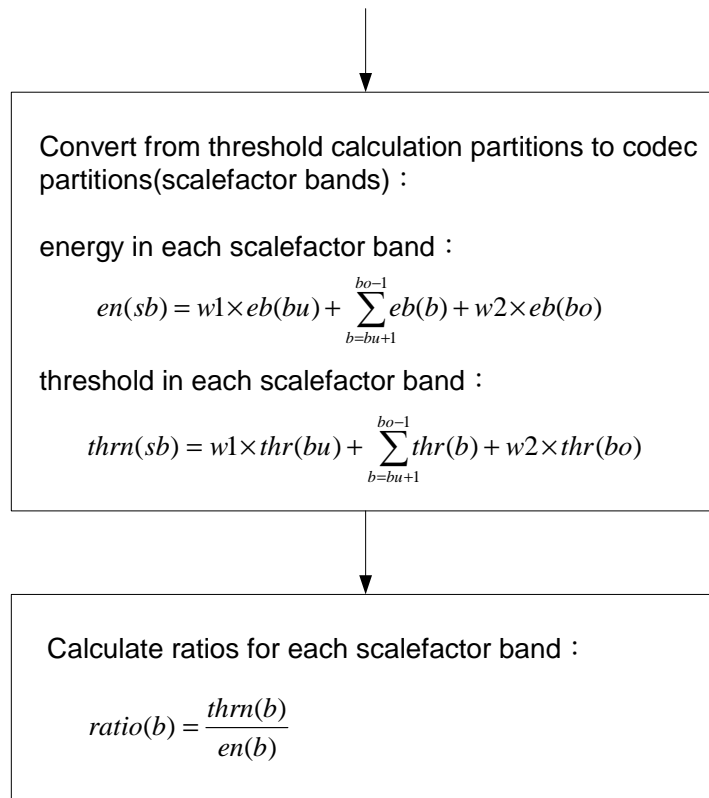


圖 3-6 (c) calculate threshold for long block (part II) [1]

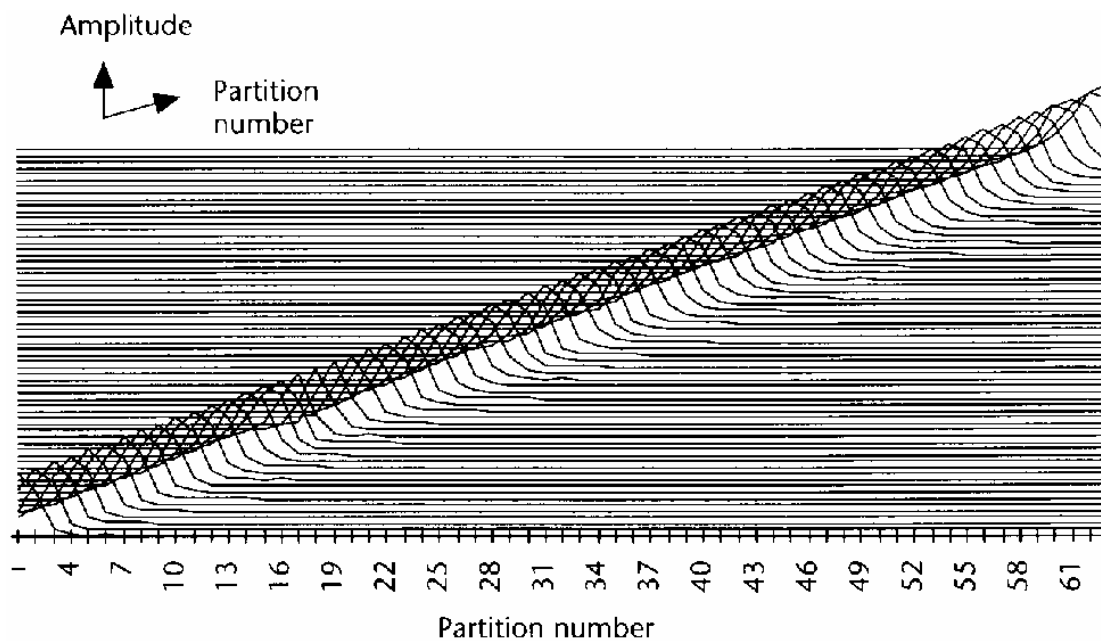


圖 3-7 長區塊的展開函數（取樣頻率 44.1 kHz） [3]

3.5 量化和位元分配

在音訊經過分析濾波器排和聲響心理模型之後，由這兩個部分提供資訊，才能進行位元分配和量化編碼。而位元分配和量化編碼是在 MP3 編碼內最花時間的一個部份，其運算量相當大。此部份的好壞，影響速度和品質甚鉅。表 3-2 是關於 MP3 編碼，在各個方塊所花的時間佔整體編碼的比例。

聲響心理模型	36 %
濾波器排	10.3 %
MDCT	4.9 %
位元分配和量化編碼	45 %
資料打包格式化	3 %

表 3-2 MP3 編碼器各方塊運算時間的比較

所以從上面的時間比，可以明顯看出位元分配和量化編碼，所花的時間的確是最多的。

接下來要提到的是如何去實作位元分配和量化編碼這個部份，這個部份是用三個迴圈來做成，分別是 Iteration Loop、外部迴圈(Outer Loop)和內部迴圈(Inner Loop)，其中內部迴圈在外部迴圈之內，而外部迴圈又在 Iteration Loop 內。以下分別以這三個迴圈說明位元分配和量化編碼的流程。

3.5.1 Iteration Loop

圖 3-8 為 Iteration Loop 的流程，其中大部份的工作和運算都在外部迴圈，其它部份只是一些初始值設定的工作，和運算完成後需要計算剩餘多少位元，可以拿來放在儲藏處(reservoir)，以便後面的編碼框可以使用。

3.5.2 外部迴圈 (Outer Loop)

圖 3-9 為外部迴圈的流程，一開始就會先進入內部迴圈，內部迴圈是在做量化的工作，而外部迴圈是在估計量化後失真的大小，算完了失真的大小，再和每個比例因子頻帶所能忍容的最大失真比較。若失真太大，就要經過某些調整後，再重新做量化的工作，直到失真程度都小於可忍容的最大失真，才跳出外部迴圈。以下一一說明外部迴圈各個方塊的內容。

A. 計算每個比例因子頻帶的失真大小

做完量化之後，必須計算已量化後頻線的失真大小，計算方式如 (2-23) 式所示，在此重列如下。利用此式，就可以算出每個比例因子頻帶的失真的大小。

$$distortion(band) = \sum_{low(band)}^{high(band)} \frac{(|xr(i)| - ix(i))^{4/3} * \sqrt[4]{2}^{stepsize}}{bandwidth(band)} \quad (2-23)$$

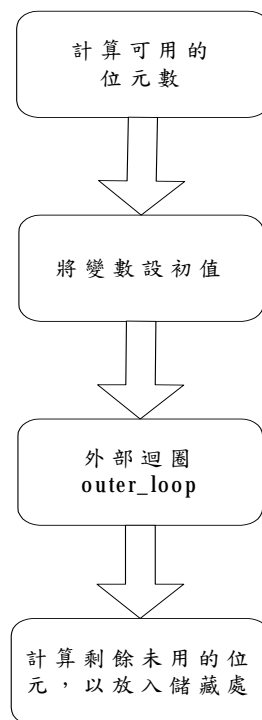


圖 3-8 Iteration Loop 的流程圖

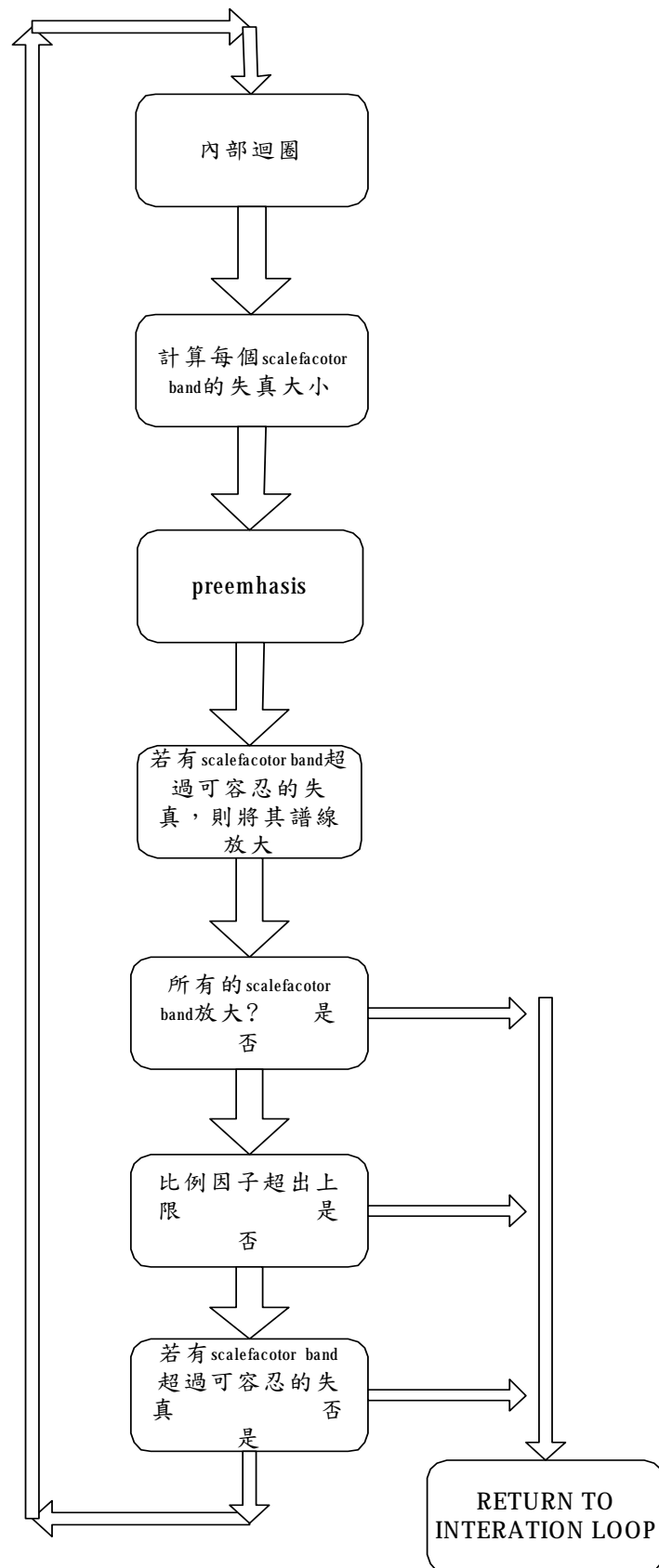


圖 3-9 外部迴圈的流程圖

B. Preemphasis :

當最高的 4 個比例因子頻帶的量化誤差大於人耳所能忍容的最大失真，則必須將**每個**比例因子頻帶可忍容的最大失真放大，並且將每個未量化前的頻線 $xr(i)$ 放大。

C. 若比例因子頻帶超過可容忍的失真，則將其頻線放大：

若有某個比例因子頻帶的失真大於能忍容的最大失真，那麼將**該**頻帶可忍容的最大失真和每個未量化前的頻線 $xr(i)$ 放大。

D. 外部迴圈結束的條件：

當以下三個條件任一個成立，則跳出外部迴圈，返回 Iteration Loop。

1. 當所有的比例因子頻帶 都有因失真過大而被放大過，則外部迴圈結束。
2. 當比例因子超出上限時，則外部迴圈結束。第 0~15 個比例因子頻帶的上限為 15，而第 11~20 個比例因子頻帶的上限為 7。
3. 若沒有任何比例因子頻帶的失真大過可容許的失真，則可跳出外部迴圈，返回 Iteration Loop。

3.5.3 內部迴圈 (Inner Loop)

內部迴圈主要是做量化，並且要確保不會造成溢位，也不能超過可用的位元數，所以是整個迴圈的核心部份。以下一一說明內部迴圈各個方塊的內容。

A. 量化：

量化的關係式如式 (2-24)，式中的 stepsize 通常都為負值。此處重列如下：

$$ix(i) = n \text{int}((\frac{|xr(i)|}{\sqrt[4]{2}^{\text{stepsize}}})^{0.75} - 0.0946) \quad (2-24)$$

若量化後的頻線 $ix(i)$ 發生溢位或是位元數不夠用的情況，則增加 stepsize 重新調整，stepsize 每次增加 1 直到沒有上述情形發生。

B. 溢位：

當量化後的數值超過 $8191+14$ 時，就會發生溢位的狀況，因為霍夫曼表最大能表示的數值就為 $8191+14$ ，所以如果發生溢位的話，就要調整 stepsize，使得溢位不再發生。

C. 計算編碼時所需的位元數：

1. 計算在高頻部份零區間 (zero_region) 的長度。
2. 計算 count1_region 的長度。
3. 計算需要多少位元去對 count1_region 編碼。

由於這裡有兩個霍夫曼表可供選擇，所以先各別計算編碼所需的位元數，再從這兩個之中，選擇編碼後位元數較少者。

D. 將 big_value region 分成三個子區間：

此三個子區間稱為 region0、region1、region2，參考圖 2-14。而切割 big_value

region 的原則是:將 big_value region 中的三分之一比例因子頻帶分配給 regon0 ,
四分之一分配給 region2 。

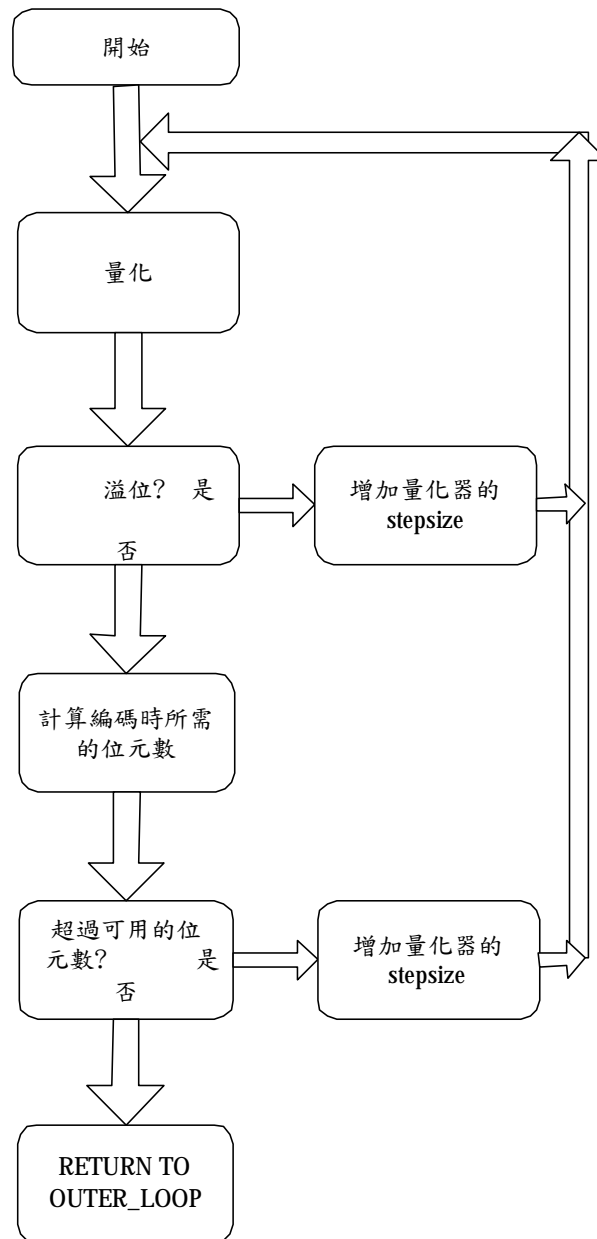


圖 3-10 內部迴圈的流程圖

E. 計算 big_value region 三個子區間所需要的霍夫曼表：

MP3 編碼總共有 32 個霍夫曼表可供使用，其特性如表 2-3。選擇霍夫曼表的方法為各別將該部份的最大值找出，再利用此最大值去尋找可表示其大小，且所用位元數較少的霍夫曼表。但是當最大值 ≥ 16 時，就不能只靠霍夫曼表，還要外加一個 escape，來表示大於等於 16 的值，而 escape 求法如 (2-27) 式。

F. 計算需要多少位元來對 big_value region 編碼：

分別計算 big_value region 的三個子區間各自所需的位元數，再相加即可。

G. 超過可用的位元數：

若超過可用的位元數的話，就要重新調整 stepsize，並重新量化，重覆上述動作直到位元數夠用，才可跳回外部迴圈。

以上大致說明了位元分配與量化的整個流程。誠如先前所提到的，在 MP3 編碼中位元分配與量化這個方塊佔了大部分的運算時間。其主要的問題還是在於量化方程式 (2-24) 的計算太過複雜，而且隨著 stepsize 的調整，每條頻線可能都經過多次的量化計算。所以必須特別針對此式做一些化簡的動作，以加快編碼速度。以下說明之。

原本 MP3 編碼所定義的量化運算式為

$$ix(i) = n \text{int}((\frac{|xr(i)|}{\sqrt[4]{2^{stepsize}}})^{0.75} - 0.0946) \quad (2-24)$$

因為採用定點數運算，所以可以將四捨五入的 nint() 函數去除，然後改寫成

$$ix(i) = |xr(i)|^{0.75} \times 2^{\frac{3 \times stepsize}{16}} - 0.0946 \quad (3-16)$$

如果量化後發現失真太大，還必須對 $xr(i)$ 做調整

$$xr(i) = xr(i) \times ifqstep \quad (3-17)$$

其中

$$ifqstep = 2^{\frac{1 + scalefac - scale}{2}} \quad (3-18)$$

因為 $ifqstep$ 也是以 2 為基底，所以可以和 (3-16) 式中以 2 為基底的部份做次方相加即可。但如果 (3-17) 式中失真太大，則會造成 xr 累乘的次數不只一次，而定點運算將難以處理這個問題。所以使用上述方法就能避免累乘，而以累加的方式取代，如 (3-19) 式：

$$\begin{aligned} ix(i) &= |xr(i)|^{0.75} \times 2^{-\frac{3 \times stepsize}{16}} \times 2^{\frac{1 + scalefac - scale}{2}} - 0.0946 \\ &= |xr(i)|^{0.75} \times 2^{-\frac{3 \times stepsize + 8 + 8 \times scalefac - scale}{16}} - 0.0946 \\ &= |xr(i)|^{0.75} \times 2^{-\frac{expo16}{16}} - 0.0946 \end{aligned} \quad (3-19)$$

所以在 $|xr(i)|^{0.75}$ 的部分就採用 3.1 節所提到的分區間直線逼近的方法來處理，而每常失真太大要做調整時，我們只要改變 (3-19) 式中的 $expo16$ 部份即可，不再重複 0.75 次方的運算。

另外，針對 $2^{-\frac{expo16}{16}}$ 的部分可以採用查表的方式，因為 $expo16$ 為整數，所以我們將 $expo16$ 再寫成 (3-20) 式：

$$expo16 = 16 \times q + r \quad (3-20)$$

式中 q 的部分代表 2 的除法次數，所以只要將 $|xr(i)|^{0.75}$ 右移 q 個位元即可。而 r 介於 0~15 之間，所以我們只要記錄 $2^{-\frac{1}{16}} \sim 2^{-\frac{15}{16}}$ 共 15 個數值，再根據 r 的值決定要與 $|xr(i)|^{0.75}$ 相乘的數值即可。

經過上述方法化簡後，編一首歌曲所需的時間只有原本的一半而已。因為量化是 MP3 編碼中最耗費時間的部份，所以用有效率的演算法來處理，對整體速度的提升有很大的幫助。

第四章 結果與未來展望

實現 MP3 編碼法的工作，分為兩個階段進行：

第一個階段先了解 MP3 的原理，並在 PC 上使用 C 語言驗證 MP3 編碼法。

第二個階段則是希望能用 DSP 來實現 MP3 編碼器。為了達到這個目標，我們配合 TI TMS320C5416 DSP 的特性，將原本為浮點運算的 MP3 編碼器改寫成 16 位元定點數的運算，並且嘗試許多方法來增進編碼器的效能，如第三章所提到的。

表 4-1 列出這個 MP3 編碼器在 DSP 上所需要的記憶體空間。雖然在實現 MP3 編碼時已經做了許多的修改，但仍耗費相當多的記憶體空間。主要的原因在於我們完全使用 C 語言來撰寫程式，經過 C compiler 編譯後，整體效能自然不如預期。未來需將程式中幾個重要的運算單元逐步改寫成 DSP 的組合語言，以提升整體的效能。

Program memory	58.29 k word
Data memory	45.93 k word

表 4-1 MP3 編碼器在 DSP 上所使用的記憶體空間

另外，在 MP3 編碼中以聲響心理模型和位元分配與量化最為耗時，它們總共佔掉至少 75% 的編碼時間。所以將來若想增快編碼時間，勢必要從這兩個部分著手。

在聲響心理模型的部分，由於[1]所提出的兩個模型只是一個參考，所以在這個部分的彈性較大，有較多的改良空間。聲響心理模型的作用在於提高編解碼後的聲音品質，所以聲響心理模型的選擇對整體的壓縮品質有很大的影響。既然如此，我們就先來看看使不使用聲響心理模型的 MP3 編碼對聲音品質之影響。

我們可用 MNR 的大小做為聲音品質之參考，當 MNR 越大則代表聲音的品質越佳。圖 4-1 畫出兩條曲線，分別代表在使用與不使用第二聲響心理模型的情形下，位元傳率與 MNR 之間的關係（詳細數值列於表 4-2）。觀察圖中的兩條曲線，隨著位元傳率的增加，兩條曲線越來越靠近，在位元傳率大於 160 kbps 時，不論是否使用聲響心理模型，MNR 的數值已經非常相近。故在位元傳率越低的情況下越是需要聲響心理模型之幫助以維持聲音品質，而在位元傳率大於 160 kbps 時，我們可以去計算第二聲響心理模型也能維持良好的聲音品質。

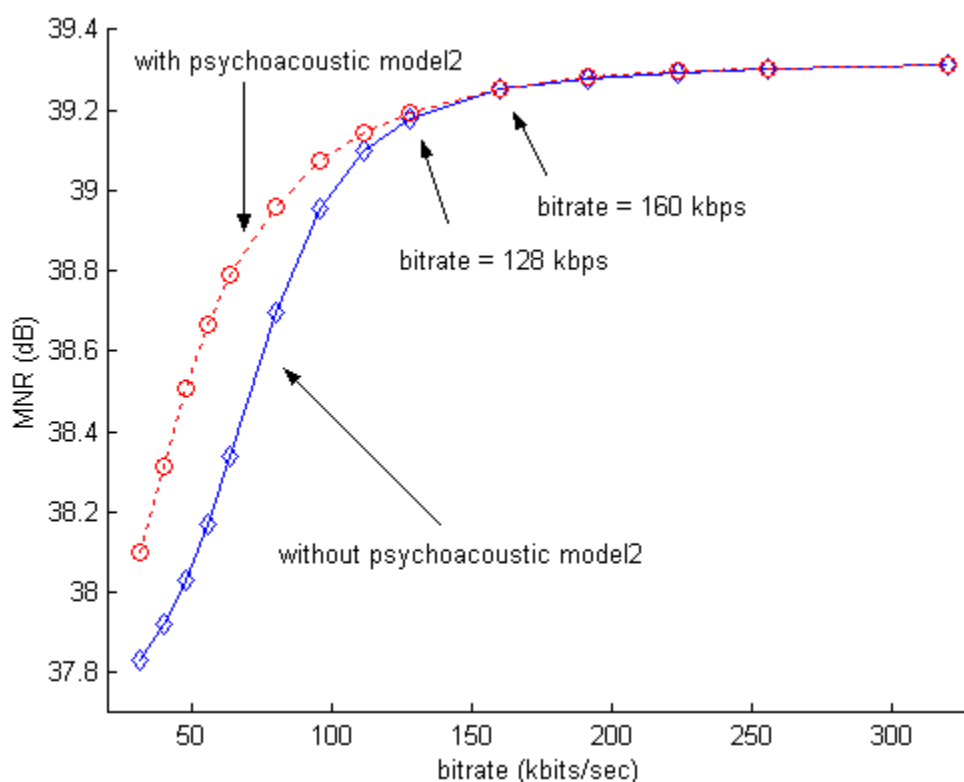


圖 4-1 有無使用聲響心理模型之位元傳率與 MNR 的關係

而在位元分配與量化的部分，由於執行迴圈的次數太多而耗費時間。若能夠找到一個方法，使得在初始化時就決定一個比較恰當的 stepsize，那麼想必能大大降低迴圈的次數。另外，執行迴圈的次數也與 stepsize 的調整方式有關，我們每次調整 stepsize 只有增加 1，也就是說，如果初始後的 stepsize 為-90，而最後

調整的結果為-80 時，我們就必須重複 10 次的迴圈運算，如果能夠彈性的調整 stepsize 每次增加的數值，那麼也能有效的降低迴圈的次數。

位元傳率 (kbits/sec)	使用 PM2 MNR (dB)	不使用 PM2 MNR (dB)	位元傳率 (kbits/sec)	使用 PM2 MNR (dB)	不使用 PM2 MNR (dB)
32	38.099873	37.832235	112	39.144649	39.096819
40	38.312876	37.921897	128	39.196040	39.179599
48	38.507716	38.030957	160	39.254509	39.253856
56	38.664636	38.171024	192	39.281187	39.280286
64	38.790752	38.337588	224	39.295744	39.293823
80	38.959583	38.696191	256	39.304471	39.303081
96	39.072675	38.953951	320	39.313353	39.313240

表 4-2 有無使用聲響心理模型 (PM2) 之位元傳率與 MNR 的關係

最後，為了讓編碼器的功能更完整，未來也考慮加入 MPEG-2[2]的部分。由於 MP3 僅能對取樣頻率 32、44.1、48 kHz 的音樂編碼，ISO/IEC 組織後來又訂定 MPEG-2 的標準。MPEG-2 為 MPEG-1 的延伸，它能對取樣頻率為 16、22.05、24 kHz 的音樂訊號編碼，亦可支援多聲道的編碼。

參考資料

- [1] ISO/IEC JTC1/SC29/WG11 MPEG, IS11172-3 “Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5Mbit/s, Part 3: Audio” 1992.
- [2] ISO/IEC 13818-7 “Information Technology – Generic Coding of Moving Pictures and Associated Audio Information, Part 7: Advanced Audio Coding” 1997.
- [3] Davis Pan, “A Tutorial on MPEG/Audio Compression,” IEEE Multimedia, Vol. 2, No. 2, Summer 1995.
- [4] Ted Painter, “Perceptual Coding of Digital Audio,” Proceedings of IEEE, Vol. 88, No. 4, April 2000.
- [5] Charles D. Murphy, and K. Anandakumar, “Real-Time MPEG-1 Audio Coding and Decoding on a DSP Chip,” IEEE transactions on Consumer Electronics, Vol. 43, No. 1, FEBRUARY 1997.
- [6] Noll, P., “MPEG Digital Audio Coding,” IEEE Signal Processing Magazine, Volume: 14 Issue: 5, Sept. 1997
- [7] TMS320VC5416 Fixed-Point Digital Signal Processor Data Manual, Ti, 2001.
- [8] Hung-Chih Lai, “Real-Time Implementation of MPEG-1 Layer 3 Audio Decoder on a DSP Chip,” National Chiao Tung University, June 2001.
- [9] E. Zwicker and H. Fastl, *Psychoacoustics Facts and Models*. Berlin, Germany: Springer-Verlag, 1990.
- [10] E. Terhardt, “Calculating virtual pitch,” *Hearing Res.*, vol. 1, pp.155–182, 1979.
- [11] Lin Pei-Ying, “Audio Coding Using Multi-Subband Hybrid Trees,” National Chiao Tung University, June 1998.
- [12] M. Schroeder, B. S. Atal, and J. L. Hall, “Optimizing digital speech coders by exploiting masking properties of the human ear,” *J. Acoust. Soc. Amer.*, pp. 1647–1652, Dec. 1979.
- [13] J. Johnston, “Estimation of perceptual entropy using noise masking criteria,” in *Proc. ICASSP-88*, May 1988, pp. 2524–2527.
- [14] TMS320C54XX Evaluation Module Technical Reference, Ti, 2001.