逢甲大學

資訊工程學系專題報告

MP3 浮水印

黃培忠(四丙)陳世軒(四丙)

林立軒(四丙)李光燿(四丙)

指導教授：李維斌

中華民國九十二年十二月

摘要

MPEG-1/Audio 標準規格提供了對於高品質數位音樂的編解碼方法，其中又以 MOEG-1/Audio 的第三層編碼法(MP3)在相同的位元傳率下的效果最佳，其複雜度亦最高，而在資料數位化的時代來臨時，人們開始重視了智慧財產權的保護，於是便產生了浮水印技術。

本專題主要是探討關於 MP3 的浮水印技術，並期望能以更多有效的方法來實作 MP3 浮水印技術，文中介紹了何謂浮水印與 MP3 的架構與編碼，以及各種 MP3 浮水印的實作方法。

特別感謝我們的指導老師李維斌老師這一年來的指導，這一年來我們遭遇了許多困難，也換過題目，但是過程中我們學習了不少東西，像是 MP3 的一些相關知識和小組研究的經驗，或多或少對於我們以後有很大的幫助。

目錄

圖目錄

表目錄

第一章 簡介

1.1 研究背景與動機

在現今這個數位的年代，由於電腦的逐漸普及化，加上資料數位化的趨勢以及網際網路的進展，資訊的傳播變得更非常容易與快速，資訊的取得也更方便了。但是數位化資訊極易被竄改與複製，這其中便牽涉到智慧財產權侵犯的問題。為了遏止非法資料的快速傳播所造成日益嚴重的侵權問題，數位資訊保護的相關問題便因應而生，也成立了許多相關的保護智慧財產權團體，如唱片業者為了抵制 MP3 而出資贊助的團體「安全數位音樂提案」（SDMI），而這類問題成為最近幾年來相當熱門的研究課題。

有人會有疑問說，為何以前數位化較不普及時，智財權的侵權問題不及今日嚴重與受到重視呢？主要原因在於類比(analog)資料與數位(digital)資料先天性質的差異。類比資料如複印件、錄音帶、與錄影帶等經過一次複製後的品質即與原始資料有異，若再經多次複製，所獲得資料品質將大不如前。因此，盜版者較無意願複製與再散佈。相反地，數位資料如 MP3 、JPEG images、CD、與 DVD 等，其每一次複製品質皆一模一樣，而且還與原始資料分不清何者才是真正的

原版。再此情況下，盜版者便有強烈意願複製與再散佈並從中獲利。最後，將導致資料擁有者不願再提供資料(如滾石唱片將不發行 CD 等)，這也會是消費者的不便與損失。另一方面，當使用者在瀏覽某些網頁(如科學博物館)時，都希望即使不能身歷其境，也能夠經由一些數位化資料詳實地的介紹，便能獲取所需的資訊。然而，資訊提供者若是有智慧財產權侵害的顧慮，而無法利用數位資料提供較詳盡和精準的解說，這將是雙方面的不便與損失。而最近國內較顯著的智財權侵權例子是成大 MP3 事件。

根據以上的事實，數位資料智慧財產權的保護可說是不容忽視。相關問題的研究在最近幾年可說是蓬勃發展，吸引許多研究者的注意。我們這組經由老師的推薦而以此為我們的專題題目，我們的題目原本是資訊隱藏，但由於資訊隱藏的範圍廣泛，所以當初我們的方向便不能馬上決定，後來經由老師的介紹才了解何謂浮水印，而我們這組經過討論後決定以”MP3 浮水印”為我們的專題題目，原因是因為 MP3 是我們學生日常生活中可以說是幾乎天天都在聽的，加上本來我們對於何謂 MP3 就有點感興趣了，所以便一致決定以 MP3 的浮水印為我們的研究方向。

1.2 章節安排

在我們小組的初步討論下，我們覺得要研究 MP3 浮水印的話就必須先由 1.何謂浮水印? 2.何謂 MP3? 這兩個方面先去了解，所以在下面兩章便先初步的介紹一下何謂浮水印與 MP3。而我們的章節安排如下:

第一章說明研究背景與動機

第二章介紹何謂浮水印

第三章介紹MP3的發展與編碼解碼

第四章MP3浮水印實作

第五章為結論與未來展望

第二章 何謂浮水印

首先我們先來初步了解何謂數位浮水印以及它的功能與目的

2.1資訊隱藏與加密有何差異

「密碼學」和「資訊隱藏」是兩種隱藏訊息的方法；雖然可以相輔相成，兩者卻是不同的。

密碼學 (Cryptography)：密碼學可改變檔案或訊息的內容，讓除了目標收件者以外的每個人都無法閱讀。目標收件者握有「金鑰」，可以解開加密檔案的鎖，並以傳送者計劃的方式來檢視檔案。加密的訊息並不是隱藏的，而且訊息送進送出都有可能受到偵測或監視。一旦找出加密的方法，仍然需要破解密碼的人來找出金鑰，才可能將訊息解密。

資訊隱藏：您可以把資訊隱藏想成一種非常堅固的加密方式。而它隱藏訊息的方式，會讓觀察者也許根本察覺不到有訊息的交換。它和密碼學不一樣，資訊隱藏是無法偵測到的。人們常常利用資訊隱藏來輔助加密。透過加密，再結合加密資料隱形，就能徹底保護訊息而不讓資料間諜得手。

而隨著多媒體與電腦網路技術的發展，人們可以輕易的藉由網路傳送訊息，包括文字、語音、靜態圖片與動態影像等，不但加快了訊

息的傳播，也刺激了新的技術的產生；然而，這其中包含有兩個問題，其一為儲存大量的數位資訊，尤其是影像或視訊資料，需要很大的電腦記憶空間，此外，傳數位資訊的時間也與影像的大小有密切的關係，有必要將儲存的資料做壓縮，使資訊的儲存與傳輸更加有效率。另一個問題是網路安全性，在網路上傳輸時，希望能夠確保不被非法者竊取並解讀。對此，除了在網路系統的安全性加強外，另一個方式就是將傳送的資料隱藏及加密，使資料無法為非法者察覺或解讀。

2.2資訊隱藏的運用--浮水印

隨著網際網路（Internet）的普及和它的便利性，使得資料的取得已經變得十分容易且得以快速的複製與傳播。尤其是當圖畫、音樂、影像等等經過數位化的處理後，任何人都可以藉由網際網路輕易取得他人的原創作品，若未經作者的同意而任意覆製、修改，而侵害到原作者的智慧財產權，為了能夠保護原創作者的權利，可利用浮水印的技術在數位資訊裡頭加入宣告擁有者的（owner）一些資訊，而浮水印便是一個跟資訊隱藏有關係的技術了。

何謂浮水印技術？簡單的說，就是一種可以將一些額外的資訊（如文字、影像、聲音等）藏到一份媒體的技術。例如像文字、、聲音、影像、多媒體影片等，都可稱作為媒體。 然而，浮水印又分為可看得見和看不見的，各有其功能在。另外浮水印技術可以用來傳遞

密秘訊息、表達言外之音、智慧財產權的保護、以及其他種種的用途等等，而這也就是為何浮水印近來會這麼受歡迎的原因之一。

2.3浮水印介紹

保護著作權的首要課題就是原創作者如何去證明某項電子文件的全部或其一部份，確實為其所創作。而最原始的作法就是將作者的電子簽章 (Digital Signature)，放進其創作裏。這裏所指的電子簽章，也就是所謂的電子浮水印 (Digital Watermarking)，用以和一般文字檔案的電子簽章區別，通常用於影像(image)、聲音(audio)、視訊(video)等資料中。

電子浮水印可分成可見(visible) 和不可見(invisible) 兩類，其做法與目的也各不相同。前者最常見的例子就是有線電視(CATV)頻道上所傳送的視訊資料角落通常會有屬於該頻道所特有的半透明商標(logo)，其最主要目的乃在於嚇阻作用，防止非法的使用，雖然減低了該資料的商業價值，卻無損於擁有人的使用。

相反的，不可見的浮水印 (invisible watermark)藉由將屬於原創作者的電子浮水印隱藏於影像資料中的不顯眼處，做為將來起訴非法使用者的舉證，因此其最主要的目的乃是增加起訴非法使用者的成功率，以保障原創作者的著作權。

當然，如果非法的使用者發現了浮水印，便會嘗試將其去除，因

此一個好的 電子浮水印必須要滿足以下五點要求：

1)透明性(Transparency):

加入的浮水印，需肉眼看不見或聽不見難以被人察覺，這樣能對於原影像或聲音品質的影響應減至最低。

(2)不可移除性(Nonremovable) :

在儘量不影響影像品質的前提下，浮水印不可輕易被輕易剪下或移除。

(3)強健性(Robustness) :

在經過一些影像處理的動作或壓縮後，浮水印仍能存在於影像資料中維持其功能或浮水印能忍受各種數位處理與攻擊，至少在其被破壞前原始影像已嚴重失真。

(4)可解碼性(Decodeability) :

對於經過授權的合法使用者，不需要原始的影像檔案，必須要能輕易地將不可見的浮水印抽取出來。

(5)安全性(Security)：

即使使用者知道加入的浮水印的程序也無法讓未經授權者移除所加入的浮水印。

但目前要做到使上面五點通通做到還有困難，所以還有努力的空間。

我們小組也是以這五個方向為努力的目標來做我們的MP3浮水印。而MP3之檔案格式，因為定義了許多檔頭（Header）及Side Information，故有許多空間可存放浮水印資料，且不會影響聲音品質，但由於可輕易地被了解MP3檔案格式的人移除，不符合上述之Nonremovable性質，故並不適用。所以一般來說是將浮水印資料，直接隨機地加在聲音資料中。所應用的原理是另一種人體聲學現象：Masking Effect（遮蔽效應），即兩個頻率相近的聲音，其中聲音較大者（Masker）會將聲音較小者（Maskee）蓋過去，造成人耳幾乎完全聽不到較小聲的聲音。故若將聲音來源先作快速傅立葉轉換，再將Sound Press Level（聲壓）對頻率作圖，則可得到Masking Threshold曲線，而在此曲線之下的聲音就幾乎聽不到了。利用此特性，則可以將適量的浮水印資料加入聲音中，形成控告非法侵權的證據，而不會被發現了。

2.4 常見的浮水印的技術

Spatial domain

最低位元法 Least significant bit (LSB)：

Least significant bit embedding 是藉由 M_sequence 來改變電子

浮水印的表現，作法如下：第一步：將影像資料轉換為 8bit 的二進位明文，舉例如果灰色圖素的值是 90 轉換結果值將為 01011010。第二步：選擇圖素的 Least significant bit 插入 watermark 一個位元轉換成新的資料，舉例說明如果我們想要嵌入一個”1”在灰色值 90 的圖素裡我們將得到一個新的圖素資料 91 它的二進位是 01011011。

當我們執行 ＬＳＢ數位浮印技術我們可以得到的好處是非常的簡單快速而且容易製作，加入浮水印的位元設在圖片區塊位元的最低位元,是不易被人眼所觀察出來的。但是相對的它的缺點是容易被雜訊及幾何改變的破壞，容易被刪除，安全性不高。

頻域轉換法

(frequency transform)

Discrete cosine transform (不連續餘弦轉換) ：

不連續餘弦轉換是屬於區段基礎的處理，自從大部分影像及圖像壓縮藉由 JPEG， MPEG 或 H.261/263 標準來處理，這些方法是

利用 DCT 的基礎技術，使用 DCT 來作電子浮水印處理是相當的合宜，當我們利用 DCT 轉換作浮水印嵌入，一個圖檔會被分割成兩個變動大小的同質性區塊，使用 DCT 作浮印處理的好處是具有非常強韌性的資料而且不會降低視覺的品質。作法如下：1. 計算 DCT 包含於 N*N 個 BLOCK。2.選擇一對互相作用的 KEY 依照下列的規則嵌入 watermark

嵌入 1 當 X>Y，嵌入 0 當 X<Y

目前最新進的展頻技術是利用 DCT watermarking。

Wavelet transform(微波轉換)：

在應用數位微波轉換之後原始資料將被轉換成數個波段，這些波段包括低-低波段，低-高波段，高-低波段，高-高波段，利用微波轉換來作浮印處理我們可以得到的好處是強化 MPEG 的程式及重作和多重嵌入浮印處理的能力。

展頻 Spread spectrum

目前在電信通訊及人造衛星系統都使用展頻技術來增加資料安全的能力及消除雜訊人為干擾，自從資料能夠應用延展技術相對的資料復原比率也相對增加，舉例：我們傳送一個”1”如果我們接收到的也是

一個”1”，我們可以說這傳送是正確的，如果將”1”延展 3 次變成”111”，這時接收到的可能是”011”，”101”，”110”，”111”，藉由多數或最大可能性的原理我們得到原來正確的資料也是”1”，最近幾年來展頻技術已經應用在浮水印處理技術上，無論如何最常使用的一個方法是利用直接展頻，在這裡我們建議兩個新的處理方法叫做區段展頻，主要是應用在區塊的處理，另外一個叫做重複展頻，是利用複製的特性，藉由以上兩種方法我們可以增加嵌入資訊的品質，減少嵌入的體積，更能增加復原的能力。

Direct sequence spread (直接序列展頻) ：

下面的例子直接說明序列展頻的作法，步驟一：假設 watermark 的資料流是”1011001”，原始圖檔資料流是”00101001(41)，01010100(84)，00111010(58)，10000111(135)，00011111(31)，10001000(136)，00000000(0)，11111111(255)，10101111(175)，…..”，在此藉由展頻技術將 watermark 嵌入在影像的資料裡面。步驟二：將 watermark 的資料流”1011001”延展 3 次可得”111000111111000000111”。步驟三：如果選擇 least

significant bit 的方法去嵌入將可得到”00101010(42)，01010101(85)，00111011(59)，10000111(135)，00011111(31)，10001001(136)，00000001(1)，11111111(255)，10110000(176)，…”。步驟四：將嵌入的資料取出我們可得到”1，1，1，0，0，0，1，1，0…”。步驟五：應用多數理論的原則來還原原始的 watermark 得到的是”1，0，1，1，0，0，1”。

Block spread spectrum (區塊展頻技術) ：

區塊展頻技術是應用在區塊及周期性的遞歸處理上，如果利用 least significant bit 將 logo X 嵌入在圖檔 Y 裡面，這個資料的容量等於圖檔圖素的總合，典型的區段展頻 least significant bit 資料嵌入的作法如下步驟，步驟一：假設 logo X 是 I*J，圖像資料 Y 的容量是 M*N，I，J << M，N(最好>32 次)。步驟二：展開 X 的容量 I*J 變成二位元的資料流，得到 I*J*8 位元資料 X’。步驟三：將圖像資料 Y 分割成(M*N)/(I*J*8)個區塊，我們叫做 Y’區塊陣列。步驟四：每一個 Y’的區塊陣列圖素的 least significant bit 位置上嵌入 X’資料，一次一個 bit，順序是 Y’[1，1]，Y’[1，

2]…Y’[k，n-k]。

以上的方法是利用區段展頻結合 least significant bit，圖檔的容量剛好等於圖素的總合，在這裡，圖檔將被區分為多少個區塊呢? 這個答案將根據不同的 logo 和不同的圖檔特性來決定。

Duplication spreading (重複展頻) ：

重複展頻的基礎只是應用反覆的嵌入處理，將 logo 展開的二位元資料流嵌入圖檔的每一個圖素裡，重複執行，直到圖檔的最後一個圖素。作法如下，步驟一：假設 logo X 的容量是 I*J，圖檔 Y 的資料容量是 M*N，在這裡 M，N >= I，J。步驟二：展開 logo X 成為二位元的資料，得到結果是 I*J*8 位元資料流 X’。步驟三：將 X’嵌入圖檔 Y 順序是 Y[1，1]，Y[1，2]…Y[M，N]，按照順序重複 X’直到 Y 檔案結束為止。

應用重複展頻技術可提高資料還原及安全性，此方法僅較優於直接序列展頻。

展頻方法的比較 ：

在直接序列展頻，浮水印的資料是不能回歸而且沒有周期性的性質，因此可靠度是比較令人不滿意的。在區段展頻方面，最重要的好處是不論 logo，watermark 或嵌入資料將很平均的被展開在原始影像檔裡，而且分佈的很完美。應用重複展頻資料的安全度可被增加，但是嵌入的資料不能分散的很均勻，它的效果比區段展頻技術較差。

第三章 何謂MP3

在初步了解浮水印後，在試著在MP3裡加入浮水印之前，我們必須了解MP3的檔案架構以及解壓縮原理，才能找到適合我們存放隱藏資訊的地方。

3.1 MP3 歷史：

MP3 所使用之演算法乃 1987 年由德國的整合研究發展機構 Fraunhofer IIS 與 University of Erlangen 合作研發出來。至今，Fraunhofer 的編碼程式，依舊被視為 MP3 編碼的工業級標準。MP3 其實是 MPEG1 Audio Layer3 之縮寫，即 MP3 原本是定義在 MPEG（Moving Picture Expert Group）中的一項聲音壓縮標準。

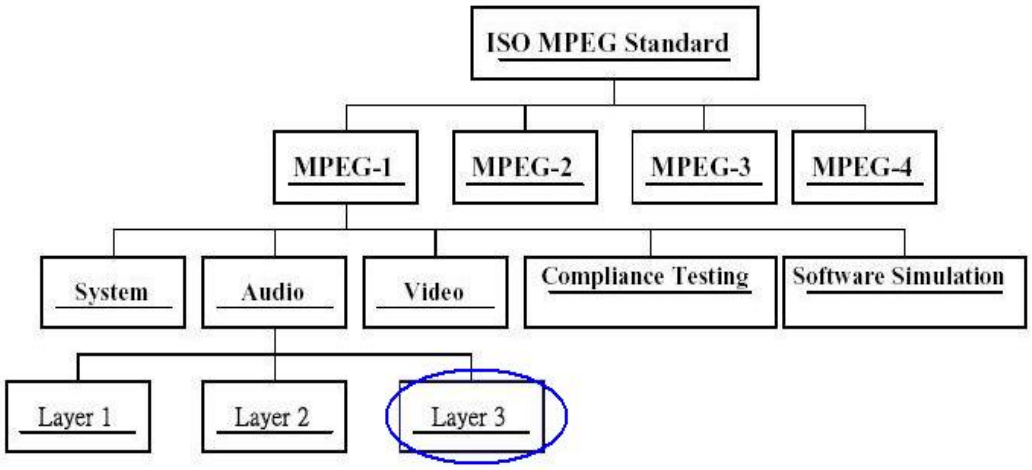


Figure 2.1 The hierarchy of the ISO MPEG Standard

圖一 ISO MPEG 組織圖

如圖所示，ISO MPEG 的標準包含了四種壓縮標準：MPEG-1、MPEG-2、MPEG-3、MPEG-4。而其中 MPEG-1 又分為五個部分：namely system，vidio，audio，compliance testing，software simulation。其中 MPEG-1 Audio 演算法是其中一種國際標準的數位壓縮技術，壓縮過程並不會破會自然音樂的音質。他不但可以獨自應用在音樂方面的應用軟體上也可以和其他視訊軟體結合使用。MPEG-1 依照不同的應用需求分成了三層：Layer I，Layer II，Layer III。這三層分別支援 32kHz，44.1kHz，48kHz 取樣頻率和四種撥放模式：Single Channel －單聲道模式；Dual Channel－雙重聲道，利用一組資料流位元表示兩種獨立的聲道；Stereo Model－立體聲模式，利用一組資料流位元表示左右兩個聲道左結合而成的立體聲。Joint Stereo Model－結合立體聲，此模組考慮到立體聲在兩個頻道之間不重要的部分和多餘的部分作修

改。

表一 MPEG1 中各 Layer 之比較：

MPEG	Layer-1	Layer-2	Layer-3
取樣率	384 kbps	192~256 kbps	<128 kbps
CD 音質輸出時壓縮比	1 : 4	1 : 6~8	1 : 12

可以發現，隨著 Layer(相當於版本之意)之更新，聲音之壓縮比愈來愈高，而品質卻未隨之下降。

MPEG 的第一層主要適用於數位錄影帶產品，此壓縮法要求較高的 384kbps 聲音數值取樣率，因為取樣率高，所以壓縮效果不大。第二層要求的取樣率為 256~192kbps，效果比第一層好，但是當取樣率低於 192kbps 時音質則明顯變差，但三層改善了前兩層的缺點，針對低取樣率的聲音所設計，此層大幅提高了音頻的解析度 18 倍，若以 CD 音質的輸出作比較，MPEG 聲音壓縮標準第一層的壓縮效率為 1:4，第二層約為 1:6~1:8，第三層的壓縮效率約為 1:10~1:12，壓縮率有大幅的提升，當然以上的比較是就 CD 音質的壓縮品質而言。

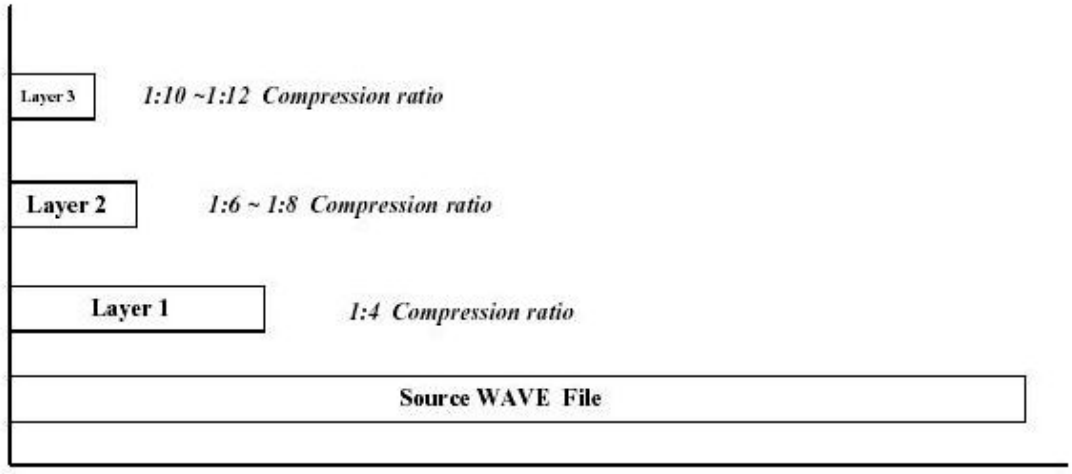
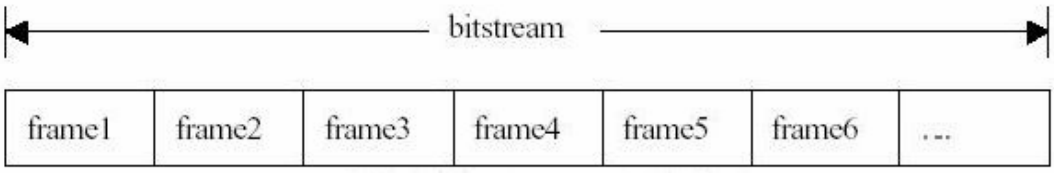


Figure 2.2 The comparison of the ISO MPEG Standard

圖二 ISO MPEG-1 Audio 各層壓縮比

3.2 MP3 檔案格式

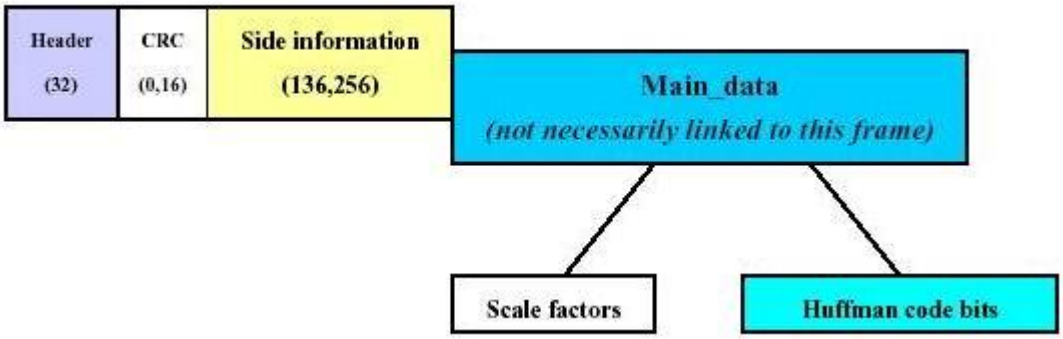
MP3的資料流(bitstream)是由一個一個frame所組合成



圖三 MP3資料流

而一個frame有固定的長度且細分為四個部分

1. 檔頭(header)： 4 bytes
2. 錯誤檢查(error check)： 0or2 bytes根據檔頭資料來決定
3. 副資訊(side information)： 17or32 bytes一個或兩個聲道(chammels)
4. 主要資料(main data)

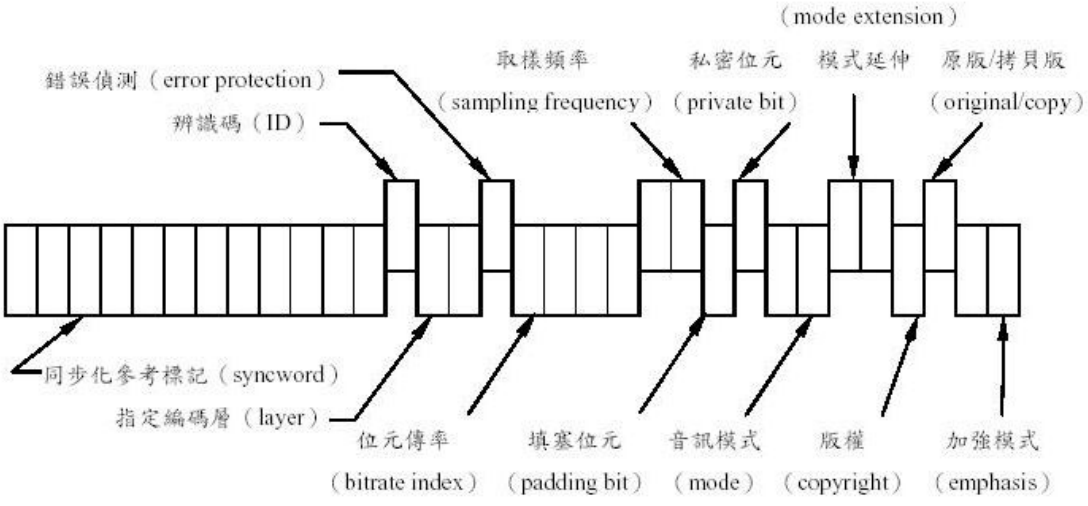


圖四 MP3 訊框格式圖

3.2.1 檔頭資料

MPEG對於Audio的資料有著很細微的規定，它共分三個層級(Layer1、Layer2、Layer3)，每個層級所用的解碼運算以及解碼器的設置都不一樣，所以它的標頭資料就變的很重要。接下來就解釋一下MPEG-1的Audio Frame Header

圖五 MP3 檔頭位元配置圖



圖五 MP3 檔頭位元配置表

<table>
<tr><th>Bit
位置</th><th>Bit number</th><th>代表意義</th><th colspan="7">分別表示的代表意義</th></tr>
<tr><td>0-11</td><td>12 bit</td><td>同步字元</td><td colspan="7">1111 1111 1111</td></tr>
<tr><td>12</td><td>1 bit</td><td>演算法 flag</td><td colspan="7">1:MPEG-1 0: MPEG-2</td></tr>
<tr><td>13-14</td><td>2 bit</td><td>Layer flag</td><td colspan="7">11：layer I 10：layer II

01：layer III 00：保留</td></tr>
<tr><td>15</td><td>1 bit</td><td>糾錯 flag</td><td colspan="7">1：未添加多餘度

0：添加了錯誤碼保護多餘度</td></tr>
<tr><td rowspan="18">16-19</td><td rowspan="18">4 bits</td><td rowspan="18">碼率 flag

(與立體聲，聯合立體聲，雙通道，單通道等具體編碼模式無關)</td><td>索引</td><td colspan="3">MPEG-1</td><td colspan="3">MPEG-2</td></tr>
<tr><td></td><td>Layer I</td><td>Layer II</td><td>Layer III</td><td>Layer I</td><td>Layer II</td><td>Layer III</td></tr>
<tr><td>0000</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0001</td><td>32</td><td>32</td><td>32</td><td>32</td><td>32</td><td>8</td></tr>
<tr><td>0010</td><td>64</td><td>48</td><td>40</td><td>64</td><td>48</td><td>16</td></tr>
<tr><td>0011</td><td>96</td><td>56</td><td>48</td><td>96</td><td>56</td><td>24</td></tr>
<tr><td>0100</td><td>128</td><td>64</td><td>56</td><td>128</td><td>64</td><td>32</td></tr>
<tr><td>0101</td><td>160</td><td>80</td><td>64</td><td>160</td><td>80</td><td>64</td></tr>
<tr><td>0110</td><td>192</td><td>96</td><td>80</td><td>192</td><td>96</td><td>80</td></tr>
<tr><td>0111</td><td>224</td><td>112</td><td>96</td><td>224</td><td>112</td><td>56</td></tr>
<tr><td>1000</td><td>256</td><td>128</td><td>112</td><td>256</td><td>128</td><td>64</td></tr>
<tr><td>1001</td><td>288</td><td>160</td><td>128</td><td>288</td><td>160</td><td>128</td></tr>
<tr><td>1010</td><td>320</td><td>192</td><td>160</td><td>320</td><td>192</td><td>160</td></tr>
<tr><td>1011</td><td>352</td><td>224</td><td>192</td><td>352</td><td>224</td><td>112</td></tr>
<tr><td>1100</td><td>384</td><td>256</td><td>224</td><td>384</td><td>256</td><td>128</td></tr>
<tr><td>1101</td><td>416</td><td>320</td><td>256</td><td>416</td><td>320</td><td>256</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>1110</td><td>448</td><td>384</td><td>320</td><td>448</td><td>384</td><td>320</td></tr>
<tr><td rowspan="5">20-21</td><td rowspan="5">2 bit</td><td rowspan="5">取樣頻率</td><td>Frequency value</td><td colspan="4">MPEG-1</td><td colspan="4">MPEG-2</td></tr>
<tr><td>00</td><td colspan="4">441000Hz</td><td colspan="4">22050Hz</td></tr>
<tr><td>01</td><td colspan="4">480000Hz</td><td colspan="4">24000Hz</td></tr>
<tr><td>10</td><td colspan="4">320000Hz</td><td colspan="4">16000Hz</td></tr>
<tr><td>11</td><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td>22</td><td>1 bit</td><td>緩衝 flag</td><td colspan="8">1： frame 內包含將平均碼率調至取樣率的附加 Slot (layer I 4 bytes， layerII 和 layer III 1 bytes，僅用於 44.1 kHz 取樣率)

0： 不包含：</td></tr>
<tr><td>23</td><td>1 bit</td><td>專用位元</td><td colspan="8">留待私用(此位元將來 ISO 也不用)</td></tr>
<tr><td rowspan="5">24-25</td><td rowspan="5">2 bit</td><td rowspan="5">模式 flag</td><td>Mode value</td><td colspan="7">mode</td></tr>
<tr><td>00</td><td colspan="7">Stereo(立體聲)</td></tr>
<tr><td>01</td><td colspan="7">Joint stereo(layer I，layer II 為強度立體聲，layer III 可以為和差立體聲)</td></tr>
<tr><td>10</td><td colspan="7">Dual channel</td></tr>
<tr><td>11</td><td colspan="7">Single channel</td></tr>
<tr><td rowspan="4">26-27</td><td rowspan="4">2 bit</td><td rowspan="4">模式擴充</td><td colspan="8">僅用於聯合立體聲模式。

在 layer I，layer II，指示哪些子帶用強度立體聲編碼 (其餘為立體聲編碼)：

00：4-31 01：8-31

10：12-31 11：16-31

在 layer III，若上面的 flag 採用聯合立體聲編碼方法，那這兩個 bits 就用來指出是採用哪一種立體聲編碼。而強度立體聲及和差立體聲的頻帶範圍則隱含在演算法中。</td></tr>
<tr><td>索引</td><td colspan="3">強度立體聲編碼</td><td colspan="4">和差立體聲編碼</td></tr>
<tr><td>00</td><td colspan="3">No</td><td colspan="4">No</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="4"></td></tr>
</table>

			01	Yes	No
			10	No	Yes
			11	Yes	Yes
28	1 bit	版權 flag	0：無版權要求 1：有版權保護		
29	1 bit	原版 flag	0：bit stream 是複製的 1：bit stream 是原始的		
30-31	2 bit	加重標識	指示應用哪一種去加重措施。 00：編碼時未加重 01：50/15 微秒加重 10：保留 11： CCITTJ.17		

以下我們來看一個例子，假設我們開啟某個MP3檔案，其檔頭為：FF FB 90 44 00 00 .....

我們先將它轉成二進位格式

1111 1111 1111 1011 1001 0000 0100 0100 0000 0000 0000 0000 .....

再對照表，我們就可以很明白的看出這個MP3 file的基本資訊。

1111 1111 1111 是同步字元，1011 表示是 MPEG-1，layer III encoding，未添加任何糾錯資料，1001 表示這個檔案是以 128 kbps 來 sample 的，1001 則說明瞭檔案 sample rate 是 32 kHz，0000 表示是立體聲模式，0100 表示無版權的原始檔案，編碼時未加重。

3.2.2 副資訊

副資訊(side infoernation)是用來控制解碼時所必須要的資訊，包含main_data_end指標、專用位元(private_bits)、窗口型態(window type)、Huffman表格號碼及他們應用的區域、安全因子(scale factor)描述符號的敘述等等。

在MPEG-1，單聲道(Single Channel mode)的Side information有17bytes，而雙聲道(Dual channel)則有32bytes。

最先出現在Side information的九個bit是MainDataBegin。他是用來指向每個Main Data開始位置的指標，利用一個帶有正副號的偏移量來紀錄與第一個Header的同步位元(Synchronization word)的相對位置。

緊連在MainDataBegin後面的是Private bits。若為Single Channel的話，Private bits是五個bits，Dual Channel則為三個bits。

接下來的數值所代表的就是Scalefactor Selection Information(SCFSI)。每個Channel有4bits的SCFSI，若為1則表示每個granule是用同一個scalefactors；0表示每個granule是由不同的scalefactor band來表示。

在granule中所代表的第一個數值是12bits的Part23Length。

Part23Length是用來紀錄saclefactors和Huffman encoded data的長度。

Part_2代表saclefactors的長度而Part_3代表Huffman encoded data的長度。

在granule中接下來的九個位元是Big Value。Big Value紀錄著在Main Data中Big Value的長度，

Golbal Gain有八個位元，他是用來紀錄經過量子化的音階大小。在Golbal Gain後面的四個位元是Scalefactor Compress。他是用來紀錄在Main Data中Scalefactor的位元數。

接下來的是Window Switch Flag。這個數值是用來決定接下來的22/44個位元(單聲道/雙聲道)所代表的意義。若Window Switch Flag等於0則接下來的位元所代表的就是TableSelect0、TableSelect1、TableSelect2、Region0Count和Region1Count。若Window Switch Flag等於1則接下來的位元所代表的就是Block Type、Mixed Block Flag、TableSelect0、TableSelect1、SubblockGain0、SubblockGain1和SubblockGain2。

TableSelect是用來決定不同的Huffman table。一共有32個Huffman table，每個TableSelect有五個位元。

在Main Data中Big Value後面還有兩個部分，region0和region1。其長度分別由Region0Count(4 bits)和Region1Count(3 bits)來決定。Block Type是用來決定視窗型態。0代表一般視窗，2代表是利用三個

小視窗，1代表由一般視窗轉為小視窗，3代表由小視窗轉為一般視窗。每個Block Type有兩個位元長。

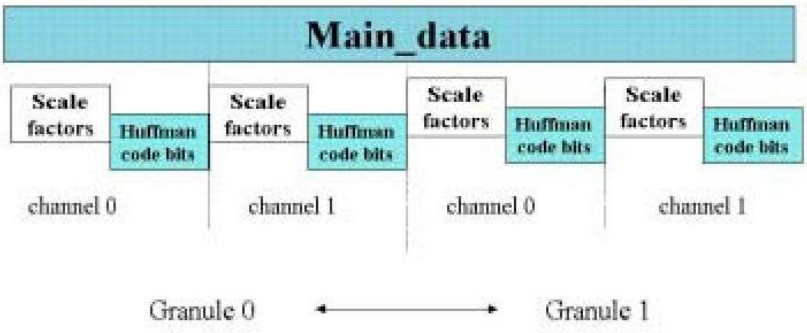
Mixed Block Flag標是出要用哪一種視窗型態來紀錄高頻和低頻的部分。

在Mixed Block Flag後面的是三個位元的Subblock Gain。Subblock Gain是只在短視窗型態下才有的變數(Block Type=2)。

在granule後面的三個位元分別是Preflag、ScaleFactorScale和Count1TableSelect。

3.2.3 主要資料

主要資料(main data)是用來儲存主要音樂資料的地方，它的編碼包含安全因子與Huffman資料且Huffman編碼值是由低頻往高頻排序的。一個Main Data中有兩個Granlue，而每個Granule包含了兩個Channel，若該Frame的編碼方式為Single channel則其Granule的兩個Channel為相同的。

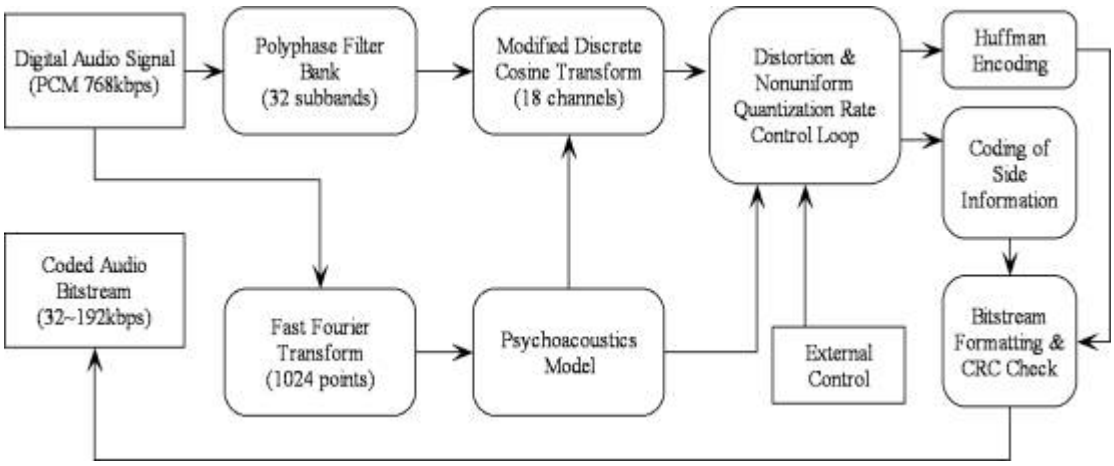


圖六 MP3 主要資料圖

3.3 MP3 壓縮原理

MP3 的基本原理是利用的人爾遮蔽效應，將人耳聽不到或是不易辨認的部分省去，只針對我們可辨認的部分作壓縮，如此一來才能使資料量大幅降低，又不至於影響音樂品質太大。

編碼流程：



圖七 MP3 編碼流程圖

上圖是 MPEG-1 Layer3 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) ，每個聲音取樣為 16 位元。MP3 編碼時，首先將原始輸入的 16-bits PCM 音訊經過濾波器排的分析(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)將每個子頻帶訊號，在細分為 18 個次頻帶，然後根據第二聲響心理模型(Psychoacoustic Model II)所提供的遮噪比(SMR,Signal-to-Mask Ratio)，對每一個子頻帶訊號，作位元分配及量化編碼。最後只要將編碼後的資料依照 MPEG-1 定義的位元串的形式輸出即可。

（1）Polyphase Filter Bank（32 subbands）：

這一節將藉由詳細的測試與分析讓我們來了解 MPEG/Audio Ployphase（多相）filter bank 的運作，也就是一個類似 decoder 的綜合 filter bank 之應用，這種多相的 filter bank 對於一般 MPEG/Audio 壓縮的三層協定 layer 1、layer2、layer3 是一個重要的關鍵組成因素，它將 audio 信號分成 32 個等寬的 frequency subbands，這些 filter 相當的簡單並且能隨著合理的頻率分析對時間來提供一個良好的解答，這種設計有幾點值我們來探討。

第一點，等寬的 subband 並沒有準確地反映出人類聽覺系統的聽覺特性。許多心理聲學上的影響都始終和 critical band 的頻率規模大小有關。例如，有兩個聲音較大的信號，其聲音清析度在一個遮罩信號的存在下會不同於那些只有一個或更多 critical band 的信號。比較多相 filter 的帶寬與那些 critical band 的寬度可知在低頻下，單一 subband 會遮蓋到數個 critical band，在這種環境下，量化的 bit 數無法明確的調諧為可獲得各別 critical band 的 noise masking。換言之，最少 noise masking 的 critical band 說明了對於整個 subband 是需要量化 bit 的數目。

第二點，filter bank 與它的反向過程並不會喪失其轉換的形態，甚至不用量化，這反向轉換也無法完美的由原始信號中找到。然而，

在前面介紹 filter bank 中，我們可知道這錯誤是很小的並且幾乎是聽不出來的。

第三點，鄰近的 filter band 會有較多重疊的頻率部份，在單一頻率下，一個信號可以影響兩個鄰近 filter bank 的輸出。有關於這些將在下面作更詳細的說明。要了解多相 filter bank，最好的方法就是去理解它的工作原理。ISO MPEG/audio 標準敘述了計算與分析多相濾波器輸出的過程，這種過程很相似於 Rothweiler [8]所提出的方法。以下是 filter bank 輸出等式(1)：

$$S_t[I] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][j] * (C[k + 64j] * X[k + 64j] \quad \text{.....(1)}$$

等式(1)中，i 是subband index，範圍從 0 至 31。S_t[I]是在時間t時subband I 的filter 輸出取樣，其中t是audio取樣區間整數 32 的倍數，C[n]是標準下定義的分析視窗之 512 因素之一，X[n] 是從 512 取樣buffer讀取的audio 輸入取樣，而M[i][k] = cos[(2*i+1)*(k-16)*p/64]是分析矩陣因子。上述的部份方程式已經最佳化來達到減少運算的時間。因為這在括弧中的方程式是與i值獨立的，而且M[i][k]與j值也是獨立的，32 filter輸出只需 512 + 32*64 = 2,560 相乘與 64*7+32*63 = 2,464 相加，或粗略地 80 加乘每個輸出。實際上，在運算乘和加的減少是可行的，例如，一快速非連續的cosine轉換 [9,10]或是一快速的傅立葉轉換完

成過程。要注意到filter bank的完成是一臨界取樣：對於每 32 輸入取樣，filter bank會產生 32 個輸入取樣。其中可運用方程式(1)轉換成較熟悉的filter迴旋方程式：

$$St[t] = \sum_{n=0}^{63} X[t-n] * Hi[n] \quad \text{.....(2)}$$

X[t] 是在時間 t 時的 audio 取樣，Hi[n] = H[n]* cos[(2*i+1)*(k-16)*p ⁄64]其中如果 n/64 的整數部份為奇數，則 H[n]=-C[n]，不然的話對於 n=0 至 511 內 H[n]=C[n]。在這種形式下，每個 filter bank 的 subband 都有自己的 band-pass 濾波響應，Hi[n]。雖然這種形式對於分析來說更方便，但很明顯的不是一個有效率的解決方法：這直接完成的方程式需要 32*512 = 16,384 相乘與 32*511 = 16,352 相加來計算 32 個 filter 的輸出。H[n]，對於多相 filter bank 符合低通濾波響應的標準。C[n] 使用了部份的最佳化方程式(1)，對於 M[i][k]來說 H[n]的 64 個係數群組每個都會有奇數。M[i][k]的 cosine 形式的範圍 k 只從 0 至 63，且含蓋了半週期的奇數部份，其中 Hi[n] 的 cosine 形式範圍 n 從 0 至 511 並包含八倍半週期的數。Hi[n]的方程式明顯地表現出每個標準響應的調整運用 cosine 形式來使低通響應移到適當的頻率帶，因此這些就叫作多相濾波器(polyphase filter)。這些濾波器在 p⁄(64T)的奇數倍數有中心頻率，其中 T 是 audio 的取樣週期，每個都有 p ／(32T)名

義上的帶寬。標準濾波響應在其帶寬並沒有尖銳的近路，所以當濾波輸出用 32 來取樣時，會產生相當大的膺頻效應(aliasing)。標準濾波器的設計和在 cosine 形式下的適當相位位移，可避免在 decoder 的綜合 filter bank [8,12]完全的發生膺頻。另一個用比帶寬還寬的濾波器所產生的結果是會使鄰近多相濾波器在頻濾覆蓋範圍會重疊，這現象會不利於高效率的 audio 壓縮，因為在 subband 邊界附近的信號能量將會出現在兩個鄰近的多相濾波輸出。

（2）MDCT(Modified Discrete Cosine Transform)：

Layer3 的演算法是由 ASPEC (audio spectral perceptual entropy coding) OCF (optimal coding in the frequency domain) 兩種演算法來加強，雖然 layer3 使用的 filter bank 和 layer1、layer2 是相同的，但是有些頻帶的濾波器輸出卻經由 modified discrete cosine transform(MDCT)來補償。MDCT 進一步將頻率輸出的 subband 做細分，以得較好的頻譜解析度。更甚者，由於將頻率的 subband 做更細分，layer3 的編碼器能將一些經由 polyphase filter bank 所產生的重疊

消除掉，當然，解碼器會將消掉的重疊部份恢復，所以 inverse MDCT，經由 filter bank 可以重建原來的聲音取樣和重疊。

Layer3 分成兩個不同長度的 MDCT 區塊：一個 18 sample 的長區塊和一個 6 sample 的短區塊，因為連續的轉移窗口有百分之五十的重疊，所以窗口長度是分別是 36 和 12。在聲音信號有穩定的特性時，長窗口有較高的頻率解析度，而較短的窗口則提供較好的時間解析度給暫態。注意短區塊是長區塊的三分之一，在短區塊的模式，可用三個短區塊來替帶一個長區塊，所以 MDCT 的取樣區塊大小對 frame 的取樣數大小不會有影響。對於一個聲音信號的 frame，MDCT 的區塊大小可以全是單一一種(長的或短的)。也可以是長短混合式的，在混合的形式中，MDCT 對兩個較低頻的 subband 使用長區塊的模式，而對其他以上的三十個 subband 用短區塊的模式。這樣為需要較高解析度的低頻提供了較好的解析度，而又不犧牲較高頻的時間解析度。

長短區間的轉換不是立即的，一個長區塊會有特別的長變短窗口或短變長窗口，來作長短區塊間的轉換。因為經由 MDCT 來處理 subband，提供較好的頻率解析度，也就是說提供了較差的時間解析度，因為 MDCT 運作在 12 或 36 polyphase filter sample，所以實際處理的時間窗口是 12 或 36 倍大。這個 MDCT 的量化值會引起整個時

間窗口的錯誤，所以這個量化就很像會產生聲音的失真。這種失真可以從 pre-echo 中發現，這是由於信號之前部的雜訊遮罩比信號後部的雜訊遮罩還微弱。

Layer 3 混用了多種方法來避免 pre-echo，第一，layer 3 的 psychoacoustic model 為了偵測出 pre-echo 而做了修改。第二，layer 3 使用保留位元(bit reservoir)的 bit 來減少 pre-echo 情形存在時的量化雜訊。最後，還可以轉成較小區塊的 MDCT 大小，來減少實際的時間窗口。

Layer 3 有特別的方法將處理 MDCT 值，以除去 polyphase filter bank 產生的重疊。為使整個的量化值能有一個一致的 signal-to-noise ratio，Layer 3 的量化器將其輸入的信號功率乘上 3/4 因子，而最後輸出時乘上 4/3 因子還原原來功率。Scale-factor band. 不像 layer1、layer2 有不同的數量因數，layer3 使用 scale-factor band，這個頻寬包含了許多 MDCT 的係數和適當的 critical band 的寬度。在 layer3 中，數量因數使量化雜訊適合 masking threshold 的頻率輪廓變化，數量因數的值被調整成雜訊分派處理的一部份。

（3）FFT（Fast Fourier Transform）：

即使用快速傅立葉轉換將訊號從 Time Domain 轉換到 Frequency Domain。因為分析訊號時，頻譜上之數值可清楚地展現出訊號的特性，所以一般會將訊號轉換到頻率軸上分析。又 FFT 是等時間間隔取樣的離散時間傅立葉轉換，可以大幅減低計算上之複雜度。此步驟主要將波形資料轉換為若干種不同的頻率，其中主頻帶有最多之資料特性，愈後之頻率則因較不重要，可以被捨棄以增加壓縮比。

（4）Psychoacoustics：

即 MP3 之最主要技術的中心思想，Perceptual Audio Coding（覺聲編）所謂的知覺編碼就是以人類耳朵可感覺到的聲音來作為編碼的基礎，在 M3 內指的就是聽覺訊號。

以下是知覺編碼有關的方法:

1.最小可聆聽範圍(Minimum audition threshold)

人類的聆聽範圍其實並不是線性的，其實有不少聲音是聽不到的，例如從 2Kz 到 5Kz，耳朵聽不到的聲音便可剔除。

2. 掩蓋效果(Masking effect)

掩蓋效果是針對人類聽覺的特色『聲音掩蓋』。

當有爆炸的聲音時，我們只會聽見爆炸而聽不見其它的聲音，因為爆炸聲音的能量太大，其它聲音被蓋過了的緣故。例如在演奏交響樂時,一些聲音大的樂器會把其它遮蓋了,在編碼時只要使用一種『心理聲學(Phycoaucostic Model)』的方法令電腦可以模擬人類耳朵，把能量最大的聲音編碼,而其它被蓋過了的聲音除掉就可以省去去多儲存空間。為人體聲學模擬的部分，主要原則即將人耳在聽覺上較不敏感或是聽不到的部分（高頻與極低頻）過濾刪除，或是利用遮蔽效應，也就是在一陣非常大的聲音之後，人的聽覺會短暫的聽不見叫細小的聲音，像是這種對我們而言，比較不清楚或是聽不到的聲音，便可將其省略，雖然會使聲音選擇性的變差，但卻可以減少資料量，達到我們壓縮資料的目的。

（5）Huffman Encoding：

霍夫曼在 1952 年所提出的一種無失真壓縮技術，其原理是將欲壓縮之字串，先讀一遍，將字串中的每一相異單字元（Single Character）的出現頻率，做成統計，依此建構霍夫曼樹（Huffman’s Tree）。每一相異單字元，用 0 與 1 予以編碼，出現次數逾多者，給予較少的位元編碼，最後將這些位元串組合起來，並加上

Huffman’s tree ，就成為壓縮檔案。

霍夫曼編碼法的特點在於所編碼出來的檔案具有唯一碼性質的即時碼。也就是各個相異字元所編碼出所位元串並不相同，解碼時能立即解出，為何如此呢，請看下列說明：

1.輸入一字串　YAHOO

2.統計每一相異字元出現次數　　Y:1　A:1　H:1　O:2

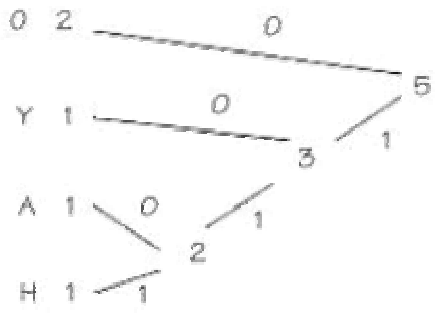
3.出現次數由大到小排列　O:2　Y:1　A:1　H:1　　並令其為各節點

4.找出加權比重小的兩個節點，為這兩節點做父節點，並將兩節點之權值相加給予此父節點。

5.重複第四步，直到找到樹根(root)。

6.建構完成霍夫曼樹，樹枝右邊給 0，左邊給 1。

7.完成編碼：



圖八 霍夫曼編碼圖

編碼: O－＞0 Y－＞10 A－＞110 H－＞111

YAHOO －＞ 10 110 111 0 0

而在 MP3 的編碼應用中，乃先將 MDCT 輸出的 576 個點依頻率大小做排序，則高頻部分會被排在最後，且數值均為零或極接近零之數，故不需編碼。其餘數值較大的低頻部分，也可依其資料之分布特性，切割為數個區域再做 Huffman encoding。如此經過 encoder 的結果，不但能減少資料量達到壓縮的目的，而撥放出來的音樂，”聽起來”亦不會與原來的音樂有太大的出入。

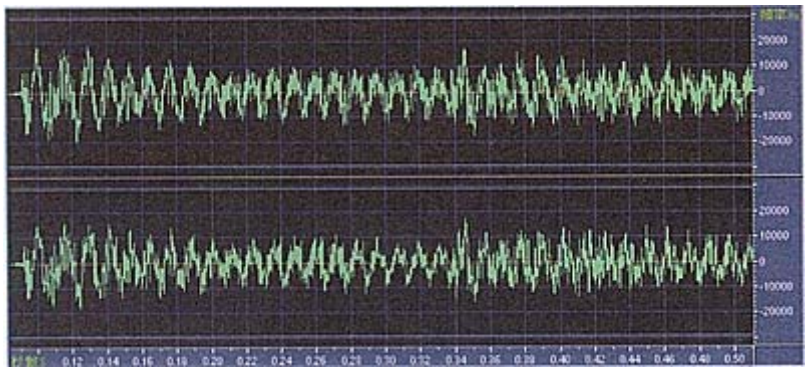
解碼流程：

即將編碼過程反過來運作，主要流程為 Huffman Decoding，Descaling，IMDCT（Inverse Modified Discrete Cosine Transform），Inverse Filter Bank。執行上比起壓縮時簡單許多，目前已經可以達到 Real Time 撥放之解碼速度。

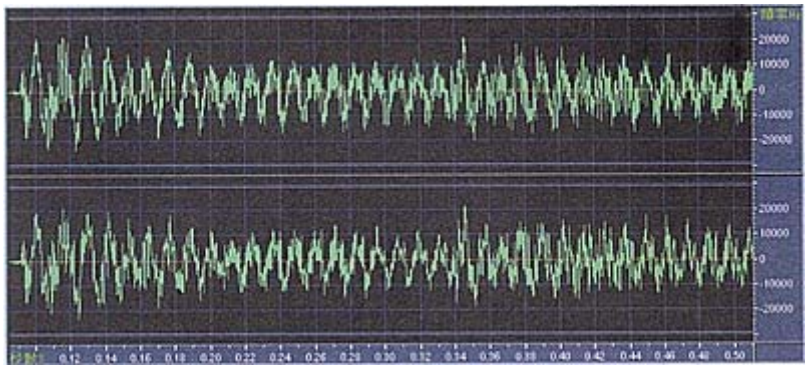
表三 MP3 之壓縮比與品質間關係：

聲音品質	聲音頻寬	取樣率	模式	壓縮比
電話	2.5 kHz	8 kbps	單聲道	1 : 96
無線電	4.5 kHz	16 kbps	單聲道	1 : 48
AM 調幅廣播	7.5 kHz	32 kbps	單聲道	1 : 24
FM 調頻廣播	11 kHz	56~64 kbps	立體聲	1 : 24~16
近似 CD 音質	15 kHz	96 kbps	立體聲	1 : 16
CD 音質	>15 kHz	112~128 kbps	立體聲	1 : 12~14

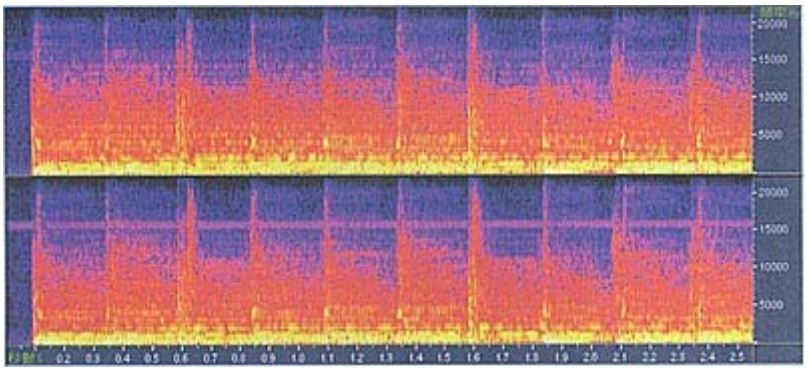
圖九 聲音原始波形：



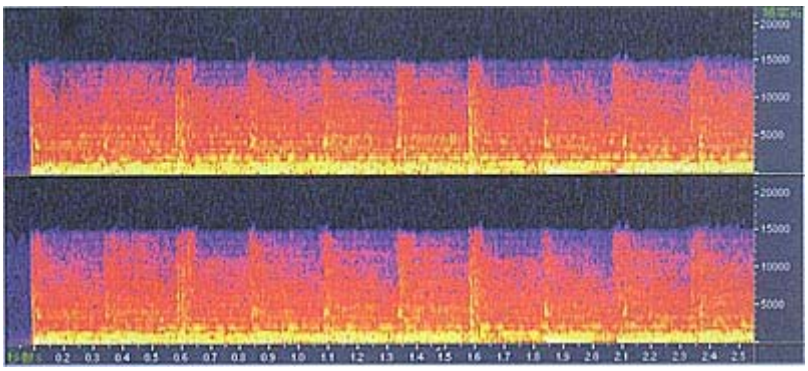
圖十 MP3 編碼後波形：



圖十一 MP3 編碼前聲音資料頻譜：



圖十二 MP3 編碼後聲音資料頻譜：

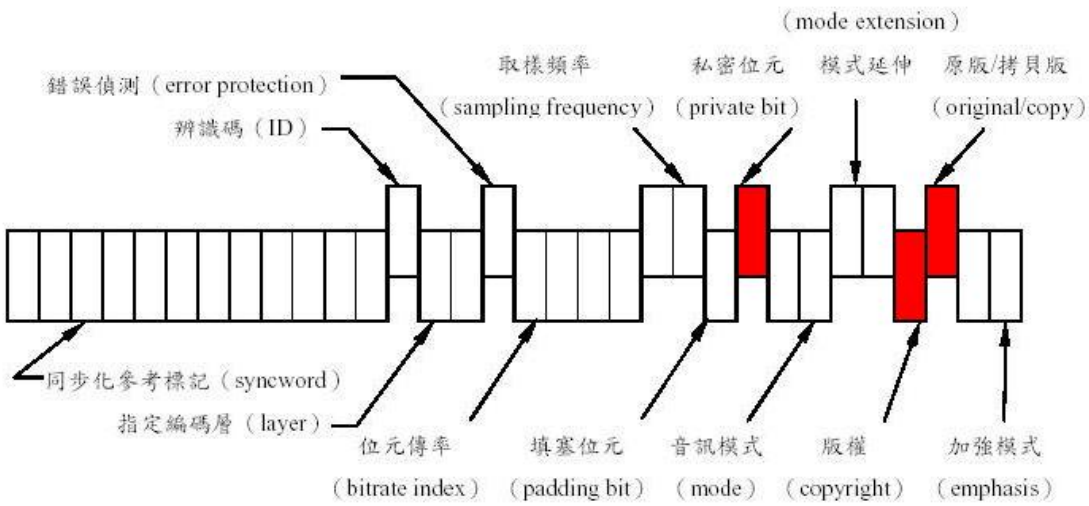


（可以看見高頻之訊號被已過濾刪除）

第四章　浮水印實作

我們嘗試下列五種方法試著去實做浮水印

（1）將 watermark 藏在 header 裡：



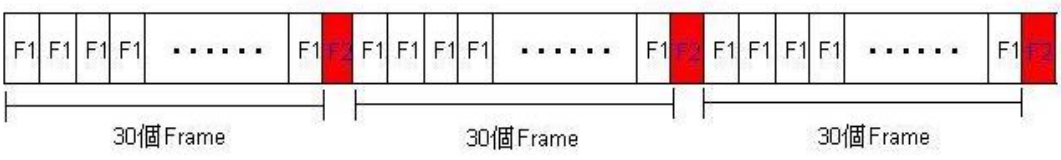
圖十三 浮水印藏檔頭

把 watermark 藏在每個 frame 的 header 中第 24bit 私密位元(private bit)、第 29bit 版權位元(copyright bit)、第 30bit 原版/拷貝位元(original bit)。其中 private bit 是 ISO 用來留待私用的位

元，copyright bit 是用來宣告此 MP3 版權的位元，original bit 是用來表示此 MP3 是否為複製品的位元。所以將 watermark 藏在這三個 bit 並不會影響到 MP3 的編碼和音質。但是因為這三個位元很的功能是固定的，所以很容易被對於 MP3 有初步了解的人在不影響音質的情況下，拿掉我們所藏的浮水印。而且由於每個 frame 的 header 就那麼三個 bit 而已，即使全部都藏，卻也藏不了多少資料，用處不大。就拿一個 MP3 檔案為例：假設有 8000 個 frame 我們所能夠存的資料量只有(8000bits÷8)×3=3000byte=3k 資料量。

（2）兩個 Frame 的交叉組合

我們曾將兩個大小不同的 MP3 以 Frame 為單位交替組合，利用撥放速度的不同讓使用者只能聽到其中一首 MP3 的聲音，而另外一個因為 Frame 與 Frame 間的間隔太大而被忽略。例如一首有 9000 個 Frame 的 MP3_1 和一首有 300 個 Frame 的 MP3_2。在 MP3_1 中每 30 個 Frame 中插入一個 MP3_2 的 Frame。假使每 0.5 秒撥放一個 Frame，因為 MP3_2 每 30 個 frame 才會有一個，會因為時間太短而被忽略。但若增快撥放速率到每 0.5 秒撥放 30 個 Frame 的話，MP3_1 的聲音就會因為撥放太



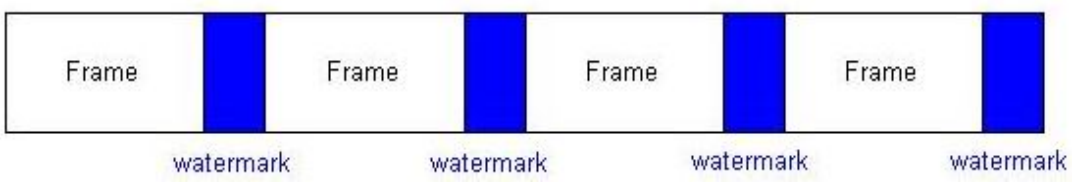
快而不清楚，相對的 MP3_2 的聲音就會還原成正成。

圖十四 相異 frame 交替

實作時因為撥放軟體在一開始只會對一種聲音的 Header 作解碼，所以即使更改撥放速率，兩首 MP3 的歌曲也只會有一首被撥放出來，而另外一首歌被視為雜音。而且在撥放軟體加速的時候並不是真正的加快了讀取 Frame 的速度，而是利用跳躍的方式由原本的每一個 Frame 讀取一次改為每三個或五個 Frame 讀取一次，如此的話所藏進去的 MP3 也無法被讀取出來。

（3）frame 的末端

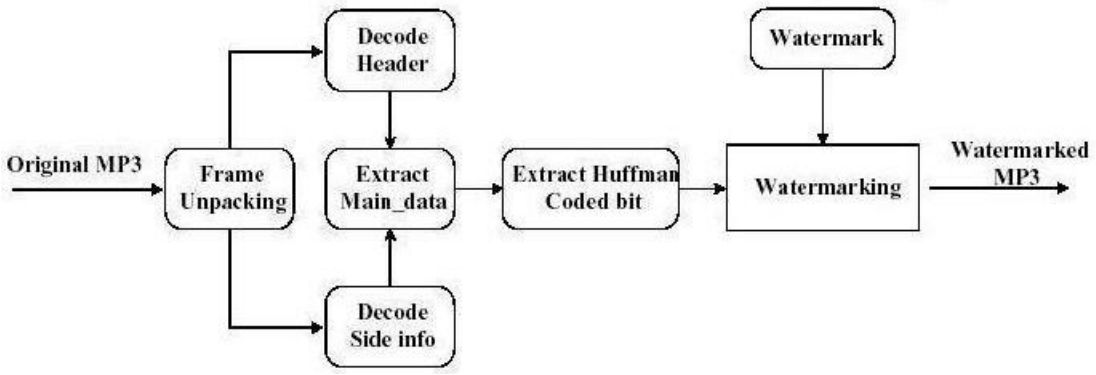
因為 MP3 中每個 Main Data 的長短不同，而決定下一個 Main Data 的開始位置是利用 Side information 中的 Main_data_begin 的指標所標示的相對來標明。Main_data_begin 是利用上一個 Frame 的結束位元再加上相對位置來決定下一個 Main Data 的開始位置。所以，我們將 watermark 藏在每個 Frame 的末端也不會影響到主要音樂區的音質。



圖十五 藏於每個 frame 的末端

但是因為 watermark 並不是藏在 Frame 裡面,所以只要利用 Side information 中的 Part2_3Length 等位元算出每個 Frame 的大小，所藏的 watermark 也很容易被拿掉。

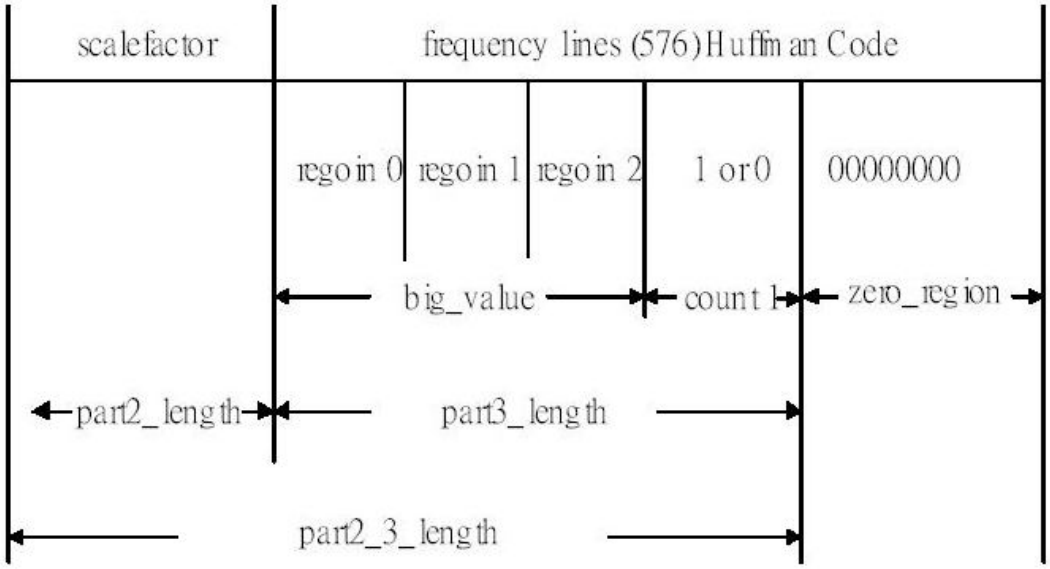
（4）利用 Decoder 藏在 Main Data 的中頻區



因為 MP3 在編碼時市將紀錄音樂的二維陣列經過 MDCT 轉換區分出高頻，中頻和低頻，然後再經過 Huffman table 的編碼才變成 Main Data 的主要音樂區，並且把所有編碼資訊紀錄在 Side information 和 Header。所以我們就利用 MP3 的 decoder 將原始的 MP3 檔經過 Header、Side information 的解碼，利用這些編碼資訊把 Main Data 音樂區的資料經過 Huffman table 還原成原本二維陣列。此二維陣列沒有經過 IMDCT 的轉換，所以

還有紀錄著高頻、中頻和低頻的區域。因為高頻區容易再經過壓縮的時候被刪除，而低頻區會嚴重破會音樂品質，所以我們取出其中中頻的部分利用 LSB 技術將 watermark 加上去。LSB 是利用人類對於八位元資料的最後一個 bit 不靈敏的特性，將 watermark 加在每個八位元資料的第八個 bit 上。

（5）直接藏入 Main Data 的中高頻區



圖十六 main data 分析圖

Main Data 再編碼時又分為 Scalefactor、big_value、count 1 和 Zero_region 等部份。音樂經過 MDCT 會將一連串的 0 是唯一個區間，稱為零區間(Zero_region)，此區間是不需要經過編碼的。Count 1 和 big_value 是由一連串的 0 與 1 組成，分別紀

錄著音樂的中頻區和低頻區。

我們再藏 watermark 時，先將每個 frame 的 Main Data 所包含的區域找出。然後從 Main Data 中第 200 個 byte(大概中頻區的位置)開始每 20 個 byte 利用 LSB 技術塞一個 bit 的 watermark。如此所涉及的範圍就是從音樂的中頻區(count 1)到高頻區(Zero_region)。

第五章 結論與未來展望

5.1 結論與困難

經過實際的去實作 MP3 浮水印時才發現並未向當初想的那樣容易，在實作浮水印之前首先必須作的便是要對於 MP3 的編碼解碼有相當的了解，如此一來才能進行我們藏浮水印的動作，我們必須要決定我們在 MP3 編碼的哪個步驟裡去進行我們藏的動作，而我們原本想做的是在先將 MP3 解碼，在將它重新編碼一次，在分析出頻率的高低範圍後的過程來進行我們藏浮水印的動作，但是對於 MP3 編碼這部分實在是我們這組的障礙，大家花了許多時間去研究與討論到如今都還未能很懂，更不用說用語言來實作了。

而在一開始的時候我們小組也嚐試將浮水印藏在 MP3 frame 的各個不同欄位，也嚐試過隨機藏在 main data 的方法，這幾種方法相較之下就比較容易多了，而我們這些簡單的方法都實作出來了。

5.2 未來展望

起初我們遭遇了許多的困難，其中最大的部分便是關於 MP3 編碼解碼的部分，這部分對於第一次碰觸這方面的我們實在是非常大的麻煩，所以我們在這部分花了非常多的時間，但是遺憾的是最後並未能將它了解並實作出來，但是我們也獲得了一些關於 MP3 的知識，在未來繼續研究 MP3 浮水印的話一定要更了解 MP3 的結構與編碼，如此一來才能做出更符合現代要求及經濟效益的浮水印技術，而相信浮水印技術在資料數位化的時代一定會越來越熱門，因為每個人都想保有自己的智慧財產，所以浮水印技術一定會繼續發展下去，甚至於應用到其他各項技術之中，比如軟體與硬體廠商相互合作，以後可能某廠商的 player 只能撥放含有某些合格浮水印的 MP3，而我們相信只要人們重視智慧財產權的一天，浮水印技術應該會繼續發展改良下去才是。

心得感想

這次專題算是第一次接觸到較大規模的團體研究，所以一開始大家都顯的沒什麼方向，因此題目遲遲不能決定，以至於浪費了不少時間，但有過這次經驗後，相信以後對於類似方式的小組研究定能作的更好。而這次的專題實驗讓我體驗到了不少的事，除了一些專業的知識以外，也體驗了分組研究的經驗，也獲得了不少東西，第一個是關於 MP3 的部分比以前還更要了解何謂 MP3 了，包括它的發展與基本編碼原理，第二個是學習了和別人一起研究的方法，第三個便是對於自己更加了解了，而在經歷了這幾個月來的努力之後，真正的知道了自己從前所學的不足，尤其是在寫程式的部分，在這段的過程中真的感到很難過，有種心有餘而力不足的感覺，而這短暫的幾個月時間內大部分的時間只能盡力的去想有什麼方法能實作出我們所期望的成果，並將自己的想法與培忠一起討論，自己卻沒能力去做出來，顯示了我這方面的不足，而我也知道程式實力不是一天兩天就能養成的，所以我日後一定會一步一步的厚實自己的程式實力的，而因為是團體合作，自己這方面較弱便從其他方面多去努力付出，最後在這段期間也獲得了不少，也看了許許多多的論文，我相信這對於我以後做研究會有很大的幫助。

陳世軒

一開始面對專題的時候，就是想利用這次機會學習如何去做一個團隊性的研究，並且希望可以讓我在大學四年所學到的東西充分應用在這次專題中。選擇了 MP3　watermark 這個題目是希望可以做些技術性的研究，以便是應付未來研究所的學習環境和出了社會以後的工作挑戰。

在這次專題研究的過程中，讓我充分的學習到如何將一個書面上的理論利用程式實做在電腦系統中。在撰寫程式時也遇到了許多的阻礙，像是在 C 語言的環境中如何從檔案中一次讀取和寫入一個位元的資料，如何將一 byte 的資料轉換成二位元的 0 與 1，如何利用指標在副程式之間做一般變數和結構變數的資料傳遞，如何寫出一個可讀性高且容易修改維護的程式等。最重要的，這次的專題開檔的檔案資料量都很大，在這幾千幾萬個 byte 做 debug 也是這次的成就之一。

第一次團隊的研究問題，一開始在工作的分配和方向的決策上都出現了很多的問題，使得我們這組一開始等於是在原地踏步毫無進展。到了後來卻又因為技術上的頻頸遭受到不少的挫折。這一次雖然由於時間的關係不能成功的將 watermark 隱藏在 MP3 當中，不過我相信下次要是再有一次研究的機會一定可以成功。

黃培忠

在一開始接觸專題的時候.是希望利用這個機會.來接觸一些課程外的東西.並將自己在大學這幾年學的東西學以致用.並且體會一下小組的研究.在這次的專題中發現.關於專業知識的方面固然重要.小組間的分工合作及溝通更是不可忽視的一點.或許是很少有機會接觸像這樣的討論.一開始我們顯的十分混亂.不過在經過一次又一次的討論之後.我們組員間的默契也漸漸的成型.這讓我體會到了小組討論共同研究的重要性.我想.在一個團隊裡.如果沒有建構出屬於他們的之間的默契.要能成功.應該是一件困難的事.

我們的專題.是將浮水印藏於 MP3 裡.因此.當然需要對 MP3 要有一定的認知.所以一開始我們是將重心放在找尋相關資料上.接下來再來實作.在找尋資料方面.我們就碰到了許多釘子.因為很多資料都是我們第一次接觸.一切對我們而言都還滿陌生的.因此我們組員間常常會將自己的看法或意見提出來給彼此參考.然後集思廣益.慢慢的.我們更加的了解一些關於 MP3 較專業的部分.接下來.便是藏浮水印.我們用了許多方法來嘗試將資訊藏進去.在這方面.我們有將簡單的部分做出來.像是從檔案或是 frame 方面下手.可是由於太過簡單.所以我們又另尋別的方法.有試著從 MP3 編解碼下手.但關於這地方.實在是我們最頭痛的一點.一些相關的知識過於艱深.我們無法克服.所以只好退而求其次的以其他部分著手.最後.則是採用了 LSB 的方式來實作我

們的系統.

在這幾個月當中.覺得所學到的東西還滿多的.像是小組間的協調.專業知識方面的補強.遇到事情如何共同去解決等等.而且.也更進一步的了解自己所欠缺不足的.在這段時間裡.由於程式部分的能力偏弱.所以比較將重心放在其他部分.最後再將成果作整理.雖然說最後的成果並不能稱得上是很完美.但是.我卻得到了更多額外的東西.了解自己不夠的地方.可以進一步加強.跟組員間的分工合作的機會.我相信.這對於我以後的幫助一定會很多.在這段時間裡.也十分感謝老師和助教對我們不厭其煩的提點.讓我們可以在摸索中很順利的進行.這幾個月下來.雖然不能稱得上是過的昏天暗地.但卻也是吃了不少骨頭.不過一切都很值得.相信以後對於這方面.一定能比現在做的更好.

林立軒

參考資料

[1] 逢甲大學資訊工程學系專題報告 MP3 保護與 Smart card 的應用

[2] 逢甲大學資訊工程學系專題報告 WEB 化的 Watermark 安全性測試平台

[3] 逢甲大學資訊工程學系專題報告 MP3 加浮水印

[4] 交通大學電機與控制學系碩士論文 MP3 編碼之研究與實現

[5] 第三波雜誌、逢甲大學資訊工程學系專題報告

[6] MPEG Audio Standard.,ISO/IEC 11172-3,1994
[7] MPEG Audio Standard.,ISO/IEC 13818-3,1994
[8] http://www.cs.berkeley.edu/~aswan/cs252
[9] http://www.subco.org/MP3/
[10] http://www.cl.cam.ac.uk/~fapp2/steganography
[11] http://members.tripod.com/gia_5/fractal/hard.htm
[12] http://tds.ic.polyu.edu.hk/mtu/hict/mas/t1/p2.htm
[13] http://members.tripod.com/gia_5/m_akustik/ch3.htm
[14]http://roger.ee.ncu.edu.tw/chinese/pcchang/course98b/comsp/MP3/3.html
[15] http://www2.ee.ntu.edu.tw/~b8901157/MP3.htm