以哼唱方式查詢 MP3 音樂資料庫

游弘明
中華大學資訊工程系
新竹市東香里六鄰東香 30 號
m8902016@chu.edu.tw

劉志俊
中華大學資訊工程系
新竹市東香里六鄰東香 30 號
ccliu@chu.edu.tw

摘要

在本篇論文中，我們提出使用哼唱的方式來對 MP3 音樂資料庫做查詢。在這種查詢架構上我們首先對 MP3 的內涵做深入的分析，在瞭解 MP3 音樂物件的種種特性之後，我們根據人類發聲的特性以及樂理的基礎訂定我們的 MP3 樂句特徵向量，文中並提出背景音樂消除方法，讓 MP3 音樂物件中的背景音樂對於哼唱準確率的影響能夠降低，使得查詢的準確率能提高。在文中同時也將透過實驗驗證我們方法的可行性，以及背景音樂消除方法對查詢準確率的增進效果。

關鍵詞： MP3 databases, query by humming, content-based retrieval, cutoff frequency, background reduction

一、簡介與相關研究

MP3 檔案格式的誕生可謂是數位音樂上的一大革命，它所帶來的高壓縮比與近乎 CD 音樂品質的低失真性，已儼然成為了現在最熱門且普及化的音樂壓縮格式。這種 MP3 壓縮格式的推出，使得原本只能儲存 10～12 首歌的 CD，現在卻可以儲存 100～120 首歌左右，其壓縮比近乎十倍，大大地增加了 MP3 在各個方面上的應用，例如：網路上的音樂檔案傳輸、MP3 音樂隨身聽、甚至用手機都可以來聽 MP3 音樂，其應用無遠弗屆。

由於現在的唱片公司眾多，旗下的歌手又是日益增加，所發行的專輯與歌曲早已不計其數，當我們去 KTV 點歌時，面對厚厚的一本歌本，要找到自己所想要的歌曲真的不是件容易的事，而 KTV 的這些歌曲只不過是茫茫歌海中的冰山一角，還有許許多多沒拍 MTV 的歌曲在人們的電腦中流傳著。在網際網路上隨著寬頻時代的來臨，MP3 音樂物件在網路上的流傳已變成了一個十分熱門的話題，從 Napster 網站的熱門程度來看即可知道 MP3 檔案所受到歡迎的程度。因此找出一種既方便又符合人性化的搜尋 MP3 方法已經是一件刻不容緩的事情。

但是現在的 MP3 搜尋軟體依舊是停留於檔名的搜尋方面，如此方式的查詢如果對於那些忘記歌名的音樂就莫可奈何了，如果我們可以透過哼唱一段旋律的方式來下達我們的查詢樣本，那麼就不需要侷限於一定要記住歌曲名稱才能找到我們所要的 MP3 檔案，因此我們想藉由哼唱的方式來找尋我們所要的 MP3 音樂物件。

在多媒體內涵式查詢的研究領域中，許多的研究學者將焦點都擺在影像或視訊方面，甚少人做音訊的內涵式查詢。以往音訊資料庫的相關研究大多把音訊物件（Audio Objects）看成一個長串的位元組（Bytes）。除了一些描述性的屬性，好比名字、檔案格式、取樣頻率外，基本上一個音訊物件被看成是一個不透明的大物件，人們都忽略了隱藏在音訊原始資料（Raw Data）的音樂內涵。在近幾年的多媒體音訊相關研究中，漸漸地開始重視內涵式的查詢與分類，不過這些研究多媒體音訊的學者，幾乎全部還是把重點都放在 MIDI 的檔案格式上，鮮少有人注意到現階段最熱門的音訊壓縮格式 MP3。

在 Martin[14]一文中提到，以往的音樂理論以及音樂訊號處理的一些方法，至今還是無法建構出成功的音樂多媒體系統，原因就是因為許多音樂多媒體系統都是以艱深的音樂理論為基礎，無法真正讓使用者了解它所分析出來的結果，因此我們希望去了解人類的聽覺，利用人類處理聲音訊號的方式，去作音樂內涵的分析，進而建構出一個以非專業人員為觀點的音樂多媒體系統。

如此，從較低階的音樂訊號的分析，到較高階的音樂心理學分析，我們就可以發展出一

本論文研究為國科會之研究成果。計畫編號 NCS-90-2213-E-216-010

個完整的音樂資料庫。在 Pfeiffer[18]中首先說明了一些聲音分析所具備的基本屬性，並介紹了一些方法來作為分析 Audio 的基礎。在這裡作者對 Audio 的部分做了許多的計算，其中包括了振幅（amplitude）、頻率（frequency）、頻率移調對應圖（frequency-transition maps）等等，利用這些的屬性來針對某些特定聲音作統計上的分析。在此，作者主要是針對暴力（包括哭聲、槍聲、爆炸）來作分析，並判別其所應該採用屬性比例應是如何。其文中還提到了一種利用 fufs（fundamental frequencies）來作為音訊（Audio）的特徵，並當成索引來做搜尋比對，此種方法的原理就在於當音訊（Audio）的旋律產生變化時，其音訊（Audio）的基頻也會隨之變化，因此 fufs 可以當作音訊（Audio）的一組相當重要的特徵值。

而對一位沒有受過訓練的使用者，最好的查詢方法就是去哼一段音樂，然後把這段音樂當成查詢的條件，去音樂資料庫中找出類似的音樂物件。這個作法稱為哼唱式查詢（Query by Humming），[6][13][9][10]等相關研究中皆使用此技術。Ghias[6]將資料庫中每一首 MIDI 音樂與哼唱式音樂查詢物件皆先透過追蹤音高模組（Tracking pitch）轉換成一組由（'U','D', and 'S'）所組成的字串，其中'U'表示現在這個音符的音階比前面高，'D'表示現在這個音符的音階比前面低，'S'則表示現在的音符與前面相等。如此一來，音樂相似度比對的問題就可以轉化成字串相似度比對的問題了。但由於人哼唱的音準與節拍會有誤差，於是字串的比對也採用近似字串比對（Approximate pattern matching）的方式。 這種容忍錯誤的近似字串比對包含了下列三種情形：置換型態錯誤（Transposition error）（如：sabla 與 sbbla）、遺漏型態錯誤（Dropout error）（如：sabla 與 sbla）、重複型態錯誤（Duplication error）（如：sabla 與 saabla）。但如此將音樂的旋律只用三個符號代表他的音調高低變化實在太粗超，因此在[13]這篇文章中，就提出將音樂旋律轉換成不連續邊圖形（broken-edge graph）的方式，使得音樂的音符長度資訊也包含於其中，以及兩個接鄰音符所差異的音調高低程度資訊也包含於其中，用以改進對音樂旋律描述不足的缺點。而在[9][10]這兩篇文章中比起以往針對音調移轉（Tone Transition）作分析之外，還額外提出了對音調分佈情形（Tone Distribution）作分析，分析整首音樂的音符分佈情況、接鄰的音高差異統計以及相對於第一個音符所差異的程度來作分析，在文中對音樂檔案的斷句部分採用滑動視窗（Sliding Window）的方式讓使用者可以針對音樂中的任何一個部分進行哼唱。

在音訊內涵式查詢比對方面，在[2]中，我們嘗試將一個音樂物件的節奏（Rhythm）、旋律（Melody）、以及和弦（Chords），轉換成音樂物件的特徵字串，並且發展出一個叫做 1D-List 的資料結構，有效率的去做音樂近似字串的比對（Approximate String Matching）。在近似字串比對演算法（Approximate String Matching Algorithm）中所做的相似度的測量，是根據音樂理論來設計的。在[2]之中，我們把音樂的物件以及音樂的查詢當作一個連續的合弦。因此我們發展一個索引的結構，去做一種比較有效率的部分比對。在[3]之中，我們提出一種方法，就是藉著音樂的節奏來找尋音樂的物件。在這個方法中，藉著音樂符號（Mubols）所構成的節奏字串（Rhythm strings），將一個音樂物件的節奏模組化。在一個音樂物件中，音樂符號就是一種節奏的樣式（Pattern），我們在兩個音樂符號中去定義五種相似度的關係（Similarity Relationships），然後兩個節奏字串的相似度就可以依照這些相似度的關係的定義被計算出來。

本論文的章節結構說明如下：在第二章中，我們將介紹有關音訊內涵式查詢的相關研究。第三章將介紹 MP3 的編碼/解碼（Encode/Decode）理論。第四章則說明 MP3 音訊的合成原理。第五章則說明如何在 MP3 音樂物件查詢系統中加入內涵式查詢。第六章則詳細說明如何從 MP3 樂句中截取出適當的特徵值來作 MP3 的內涵式查詢。第七章則針對我們所提出的 MP3 內涵式查詢方法進行實驗。最後則對本論文做個總結並說明未來研究的目標。

二、MP3 內涵式查詢

在本章節中將會介紹什麼是內涵式查詢、內涵式查詢的特點與重要性，以及如何使用哼唱的方式來查詢我們所想要的歌曲。

2.1 MP3 內涵式查詢之涵意

在傳統式的查詢結構中，主要的都是針對於資料的名稱（如：檔名）來做查詢，或是採用關鍵字的方式來做資料的搜尋。舉個例子來說：我們每天幾乎都會接觸到的 WEB，在許多的入口網站中都會提供一個查詢介面來讓使用者查詢 WEB 上面的網頁資料，這種的查詢方式即是讓使用者輸入想要找的一些關鍵字，然後再透過這些關鍵字與網路上的網頁與標題做比對搜尋，如此的搜尋方式也許很方便，但並不見得能夠找到我們真正想要的資訊，尤其是在多媒體的資料方面，像是：影像、音訊、視訊等。

在現在這個影音漸漸地數位化的世代中，數位影像、數位音樂、數位的動畫大量地充斥在我們的生活周遭中，而且不斷的在成長，現在的網路又是如此的便利，使得數位的影音已經隨手可得。在這樣大量的多媒體資料當中，我們如何去找到我們想要的多媒體呢？或許可以利用傳統的方式，以文字來當作多媒體資料的索引，提供使用者用關鍵字的方式作查詢。但是，這樣子的方式會有個問題存在，那就是：有些多媒體資料很難用文字去表達，而且每個人對於多媒體所做的描述都不相同，如此一來增加了許多搜尋上的困難度。並且如此龐大的多媒體資料當中，我們又能記得起每個多媒體資料的文字索引嗎？？因此，內涵式查詢在此就突顯出了他的重要性與必要性。但何謂內涵式查詢呢？其就是針對各種媒體實質上的內涵所做的分析，擷取該媒體最具代表性的特質作為此媒體的特徵索引。例如：在影像上，如果我們只記得那張圖是在月光下的小河旁所拍攝的，但我們忘了這張圖的名字時，關鍵字查詢就完全派不上用場了，但是內涵式查詢卻可以幫我們找到想要的資料，此時就可以透過畫一個月亮與一條小河來搜尋這一類的影像檔案。將其應用在其他的多媒體上道理相同，如此使用各個多媒體本身的特質來做查詢的技術，我們稱之為多媒體之內涵式查詢。

由於本篇論文所針對的多媒體資料為MP3音訊資料，因此，我們首先必須對MP3的音訊資料的內涵作分析。現在的MP3音訊資料大部分都是以歌曲為主，而本篇論文也將把焦點放在MP3的音樂上面。在對音樂檔案做查詢方面，最直覺的查詢方式莫過於使用哼唱的方式（Query By Humming），在很多的時候，我們總是常常記得某個旋律很熟悉，但卻苦於不知道他的曲名，因此無法得到我們想要的音樂檔案，而現在我們改用哼唱的查詢方式即可很方便的尋找我們想要的音樂物件。但是，哼唱一整首歌曲來做查詢那將會嚴重的拖垮查詢速度，而且最重要的問題是：很少人能完整的記起一整首的旋律。所以我們將整首歌切割成一句一句的***MP3 樂句（MP3 Phase）***，使用MP3樂句來當做查詢的的單位，以利使用者做查詢。

2.2 哼唱式查詢所遇到的問題

當我們使用哼唱的方式來搜尋我們資料庫中的MP3歌曲時會遇到下列幾項問題：

(1) 背景音樂的干擾：MP3音樂除了歌者唱歌的聲音之外還包含了背景音樂的部分，背景音樂的存在就是使得我們查詢準確率十分不穩定的頭號殺手，因為我們透過麥克風對資料庫中的歌曲下哼唱式查詢時是不可能有樂隊在旁邊幫忙演奏的。

(2) 樂句長度不一：雖然哼唱者所哼唱的MP3樂句與資料庫中的某樂句為相同一首歌的樂句，但可能因為哼唱者趕拍子，或是拖拍子而使得哼唱樂句長度與資料庫中的樂句長度不一。另一個導致樂句長度不一的原因就在於樂句尾音的部分，樂句尾音長度時常是讓人難以準確抓住它的拍子長度，甚至有時會隨著個人喜好去表現尾音。

(3) 哼唱音準偏失：這個原因可能導因於哼唱者對於歌曲旋律的不熟悉，或者是哼唱者本身對於音樂旋律並不敏感，無法抓住原歌曲的旋律變化。

(4) 哼唱節奏失準：這個原因可能導因於哼唱者對於歌曲節奏的不熟悉，或者是哼唱者本身沒什麼音樂節奏感，無法抓住原歌曲的節奏變化變化。

根據上述種種可能遇到的問題，這些都是在做音訊內涵式查詢時很有可能遇到的問題，在本篇論文的後面部分將針對這些問題提出我們的解決方案，讓這些問題對查詢準確率的影響能降到最低。

三、MP3 音訊編碼/解碼原理

MP3的全名為MPEG 1 Layer 3。乃是源自於MPEG（Motion Picture Experts Group）組織所制訂的多媒體影音標準，像是之前相當普及的影音光碟VCD即是採用MPEG 1的多媒體影音標準，目前市面上逐漸流行的DVD則是採用MPEG 2的標準。而我們所著墨的MP3是制訂於MPEG 1 Audio[1]標準中。接下來我們就針對MP3的音訊壓縮原理做個說明。

表 1　Layer1,2,3 特性比較

	Layer 1	**Layer 2**	**Layer 3**
Filter 模組	Polyphase Filter	Polyphase Filter	Polyphase Filter + MDCT Window
頻帶數	32 Subband	32 Subband	576 Frequency Line
頻帶寬度	689.0625 Hz	689.0625 Hz	38.28125 Hz
聲音品質	較差	普通	較好(近似 CD)
壓縮比	1:4	1:6~1:8	1:10~1:12
壓縮複雜度	簡單	普通	複雜

在 MPEG 1 Audio 標準裡分為三層：Layer 1, 2 ,3，根據其階層等級的不同其所需的其壓縮演算法也有所不同，其特性如表 1 所列。

MP3 的音訊壓縮格式可以將一般聲音資訊以 1：10 到 1：12 的壓縮比例進行壓縮，而且其音質可以近似於 CD 的音質，其原因就是 MP3 對音訊多了一道重要的處理，那就是透過人類心理聽覺模組（Psychoacoustic Model）將我們人類耳朵所聽不到的聲音過濾掉，稍後我們會再詳細介紹這個模組。

3.1 MPEG 1 音訊編碼原理

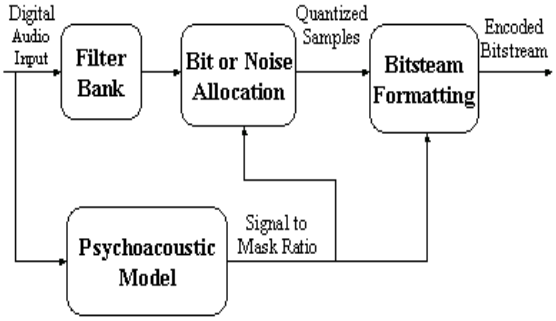

圖 1 MPEG 1 音訊編碼流程圖（參考[7]）

圖 1 顯示 MPEG 1 的音訊編碼架構中最主要的涵蓋了四個模組部分（Block）：當音訊源進入了編碼架構中，首先會將音訊源傳入濾波器模組（Filter Bank）把音源訊號細分成許多的子頻帶，此時音源訊號也會同時地傳入人類心理聽覺模組中，這個模組會根據人類聽覺的一些特性來決定哪一些音訊資料是可以不需要記載的，因為人類的耳朵感覺不到。之後便將這些資訊傳給位元/雜訊配置模組（Bit/Noise Allocation），此模組會根據心裡聽覺模型所傳來的資訊及濾波器組所分割出不同頻帶的音訊資料，做適當的資料配置（將人類聽不到的音訊資料去掉）、量化的動作（也就是決定要用多少個位元去表示），最後將經過量化的樣本，經過位元流封裝（Bit Stream Formatting）模組將其包裝成一個一個的 MP3 框架（Frame）格式，最後輸出整個編碼後的音訊格式。

3.2 濾波器組與修正式餘弦轉換

我們知道濾波器的用意在於將一個訊號裡的某些頻率的訊號過濾出來，例如：將低頻的訊號過濾出來的我們稱為低頻濾波器（Low Pass Filter）；將高頻的訊號過濾出來的我們稱為高頻濾波器（High Pass Filter）；如果將一些固定頻帶的訊號過濾出來我們稱為頻帶濾波器（Band-Pass Filter）。

所謂的濾波器組便是由一堆的濾波器所組成的，濾波器組的功用為做時間領域-頻率領域的轉換（Time-Frequency Mapping），Layer 1,2,3 的音源訊號首先都將經過一個多相位濾波器組（Polyphase Filter Bank），他是由 32 個頻帶濾波器所組成的，所以多相位濾波器組可以將輸入的訊號分割成 32 個相同寬度的頻帶（Frequency Bands）信號。圖 2 為其示意圖。

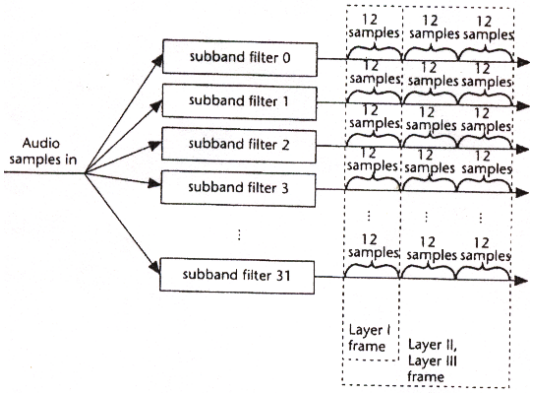

圖 2 濾波器組示意圖（參考[17]）

Layer 3 除了用到多相位濾波器組，他還用到了修正式餘弦轉換（Modified Discrete Cosine Transform）。DCT（Discrete Cosine Transform）為一種將時間領域轉換成頻率領域的轉換，MDCT 的用意在於將 Polyphase 濾波器組的輸出訊號經過再一層的解析，進而分成更細的頻帶信號。經由 Polyphase 濾波器組所分割出的頻帶訊號經過 MDCT 的解析可以再細分成 18 個頻帶信號，這樣的用意在於可以提供較好的頻譜解析度（Frequency Resolution），進而找出因為只用 Polyphase 濾波器組所造成的訊號重疊誤差。圖 3 為其示意圖：

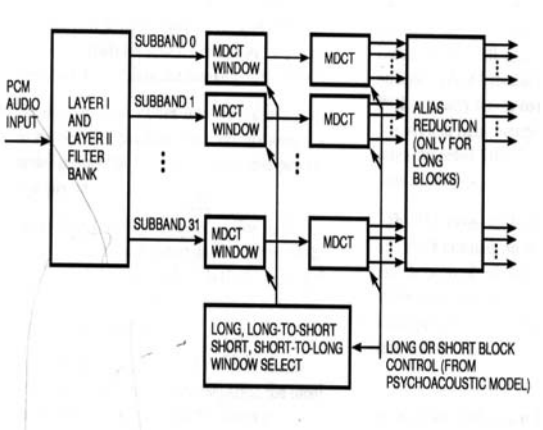

圖 3 MDCT 架構圖（參考[17]）

在本論文中，我們所使用的 MP3 特徵值擷取來源便是取自於經過 MDCT Window 解析輸出後的頻率能量係數。

3.3 人類心理聽覺模組

MP3 如此的高壓縮比又低失真的特性，對此最大貢獻的核心部分莫過於人類心理聽

覺模組，因為透過了人類心理聽覺模組，MP3 的音訊資料中可以移除掉許多人類耳朵所無法感覺到的聲音訊號，如此一來，所要記載的音訊資料頓時大量的減低，也就造就了 MP3 如此高壓縮比的特性。

人類心理聽覺模組提出了下列幾點人類耳朵聽覺的一些特性：

(1) 人類耳朵聽覺的範圍大約在 20Hz 到 20KHz 左右，並且對於頻率的不同，敏感度也相對的不同，最敏感的範圍大約在 2~4KHz 之間。

(2) 頻率遮照（Frequency Masking）：當有一個聲波頻率產生時，其周圍的頻率若是能量低於某個臨界值（Threshold）時，我們將聽不到其周圍頻率的聲音。

(3) 時間遮照（Temporal Masking）：當有一個聲波頻率產生時，緊接著又出現附近頻率的聲音，此時在一個短時間內我們無法聽到其附近頻率的聲音。

整合以上這種種人類心理聽覺的特性，我們可以得到一個聲音遮照的圖形，如圖 4。在圖中我們可以清楚的看到陰影的部分就是被遮照的聲音，也就是我們人類耳朵所無法聽到的聲音。

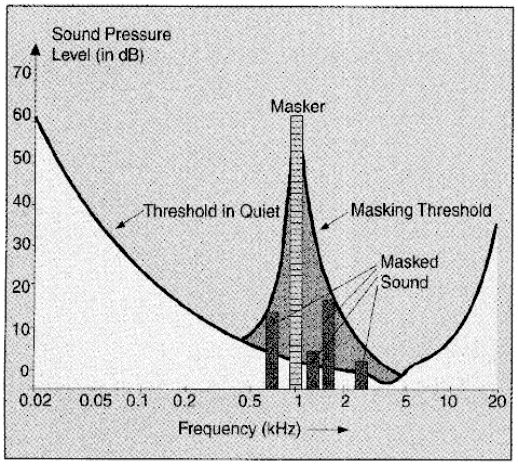

圖 4　Threshold in quiet and masking threshold（參考 [16]）

四、MP3 音訊解碼係數之線性原理

在第三章當中我們已經詳細的介紹了 MP3 的音訊編碼原理，也提到了我們將對 MP3 完全解碼過後的 MDCT 頻率能量係數來進行我們的 MP3 內涵分析，但在進行 MP3 內涵分析之前，我們必須對 MP3 音樂檔案與 MDCT 頻率能量係數之間的關係做更深一層的瞭解。

在前一章節我們提過，MP3 的音訊資料在經過多相位濾波器之後會將 Raw Data 分成了 32 個等寬的頻帶（Subband），而每個頻帶會再經過 MDCT Window 細分成 18 個 MDCT Frequency Lines。因此一個 MP3 音訊框架經過解碼模組之後會得到 576 個 MDCT 頻率能量係數，這 576 個 MDCT 頻率能量係數所分割出來的頻帶是等寬的。一般我們 MP3 音樂檔案取樣頻率都是採用和 CD 音樂取樣頻率一樣的 44.1Khz，但是因為取樣頻率理論提到若是要達到不失真的取樣，必須採用欲取樣頻寬的兩倍，故採用 44.1Khz 所取樣出來的頻帶介於 0Hz ~22.05Khz 之間，因此我們可以計算出來一個 MDCT Frequency Line 的頻寬大約為 38.28125Hz。為了觀察 MDCT Frequency Lines 等分聲波頻帶的特性以及線性的特性，我們使用了 Cool Edit 2000 來產生純 Sine 波，並轉換成 MP3 壓縮格式，從中擷取出 MDCT 特徵係數，來驗證此兩種特性。

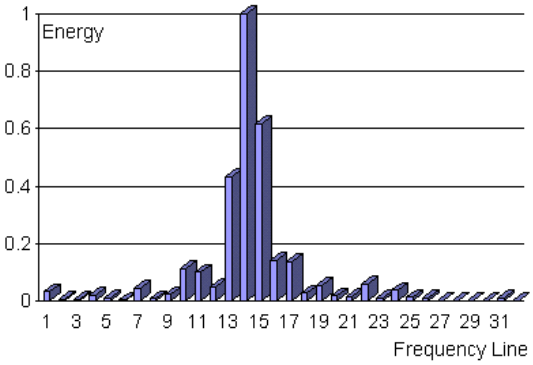

圖 5　Sine 波 -- 523Hz（C5）

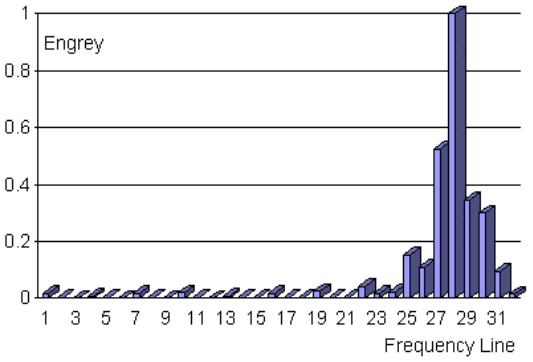

圖 6　Sine 波 -- 1046Hz（C6）

首先我們使用 Cool Edit 2000 產生一個頻率為 523Hz（中央 Do 的音，也就是樂理上的 C5）的純 Sine 波，然後將之轉換成 MP3 的壓縮格式，並且做正規化（Normalized）的動作，我們預估此波形的 MDCT 係數能量值應該會落在接近第十四條 MDCT Frequency Line 上，圖 5 中 MDCT 係數能量值的圖形也證明了此理論。由於係數只集中在第十四條 MDCT Frequency Line 左右，所以我們圖上只擷取前三十二個 MDCT 係數能量值來觀察。

我們同時也可以看到在第十四條 MDCT Frequency Line 的左右兩邊都還有一些些微的係數值，這個情況乃是因為每個 MDCT Window 與接鄰的 MDCT Window 會有重疊（Overlap）的情況產生，所以當產生一個單一頻率的純 Sine 波形，此波形的 MDCT 係數能量值也不會單單的只落在某一個 MDCT Frequency Line 上。

再來我們就要針對 MDCT 係數能量值的線性理論來做個觀察，首先我們個別產生兩個 Sine 波，頻率分別為 523Hz（C5）與 1046Hz（C6），也就是我們一般所稱的中央 Do 的音與高音 Do，並分別取出其 MDCT 特徵值能量係數，將此兩個波形的 MDCT 特徵值係數以 1：0.5 的比例加在一起，然後將此合成係數值與使用 Cool Edit 2000 把此兩個聲波以 1:0.5 的能量比 Mix 起來所產生的 MDCT 係數相比較，如圖 7 與圖 8，我們可以發現幾乎一模一樣，因此，我們可以發現 MDCT 特徵值能量係數是具有線性的關係的。

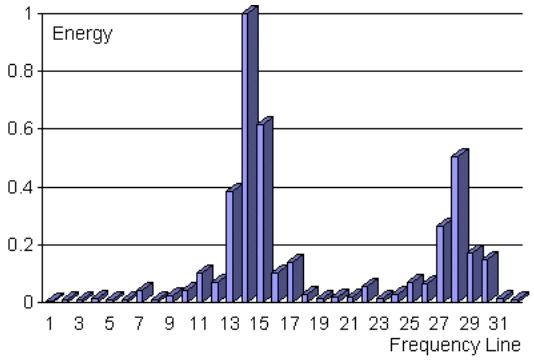

圖 7　以 1：0.5 用 CoolEdit 合成 C5 與 C6

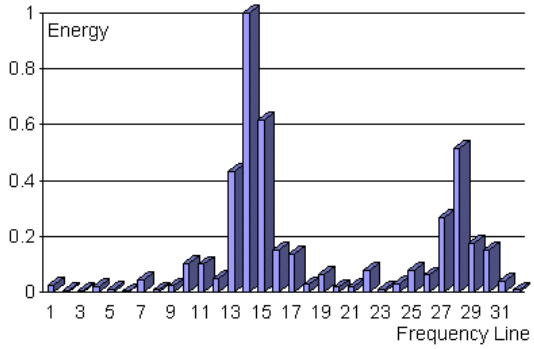

圖 8　C5 與 C6 頻率能量係數以 1：0.5 相加

有了 MDCT 頻率能量係數的此種特性之後，接下來我們就利用此特性發展出一些讓背景音樂影響減低的方法，以使得我們的 MP3 哼唱式搜尋系統準確率能夠提高且穩定。

五、MP3 音樂物件特徵值的擷取方法

在第 4 章中提到了從 MP3 音樂物件所擷取出來 MDCT 頻率能量係數的一些特性之後，在本章節中我們將更加地深入 MP3 音樂物件的特性，並對流行音樂作分析，擷取出更具體且具代表性的特徵值來作為 MP3 音樂物件的資料索引。在本章節中同時也會提出一個***背景音樂消減（Background Reduction）***的方法來解決背景音樂所造成哼唱查詢不準的問題。

5.1 流行音樂之特色分析

在現在坊間的流行音樂當中，我們可以很清楚的發現那些流行歌曲排行榜中的歌曲有個很明顯的共通點，那就是有個很固定的節奏來讓人們加深對歌曲的印象，也因此讓人們能夠很容易的朗朗上口。而流行音樂節奏呈現於歌曲上最常使用的方式，就是使用 Bass 使得人們對於該歌曲產生明顯節奏的感覺。現在的音樂錄製過程中所採用的 Bass 產生方式，大都是以電子鼓來產生其效果，不論是採用真鼓或是電子鼓，其所產生的頻率範圍都差不多。在此我們就以實際上的"定音鼓"來分析流行音樂中 Bass 的主要成分。定音鼓與一般鼓的差別就在於它所產生的頻率可以固定，它具備了調音裝置，可以調整鼓音的高低，但音域仍脫離不了鼓的特色，那就是頻率很低，實用的音域一般都不會超過純五度的音程。圖 9 就是低音定音鼓的音域。

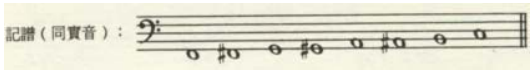

圖 9　低音定音鼓的音域

我們可以發現定音鼓的音高大約在低兩個八度的 La 上，發聲頻率約在 220Hz，因此我們取流行音樂的 220Hz 以下的頻帶來作為分析 Bass 節奏的依據。我們以錢櫃 KTV 國語新歌排行榜（2001/08/29~2001/0904）的冠軍歌曲為例來作分析，此首歌歌名為"綠光"，歌手為"孫燕姿"，此首歌已經連續兩週冠軍，我們取出此首歌中副歌的第一句來作為分析數據。圖 10 就是這句" Green Light I'm searching for you"樂句在時間域（Time Domain）上的波型圖：

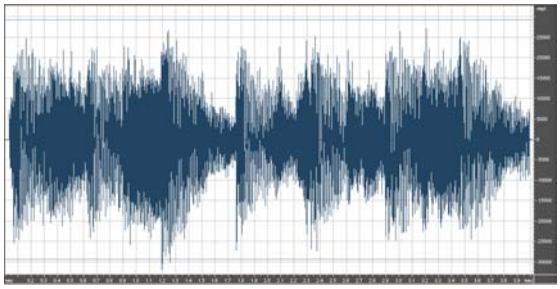

圖 10　"Green Light I'm searching for you" 在時間域上的波形

此句 MP3 的波形或許我們不容易清楚地看出他的節奏，但是我們經過一個低通濾波器（Low Pass Filter）把 220 以上的頻率過濾掉只留下低頻 Bass 部分之後，我們可以很清楚的看到節奏很明顯的突顯出來了，如圖 11。此時若播放此波形，我們也幾乎聽不到人聲的部分，只聽得到 Bass 的節奏聲音而已。由此我們就可以發現那些搖滾流行音樂的一大特色，那就是：節奏有一定的規律。

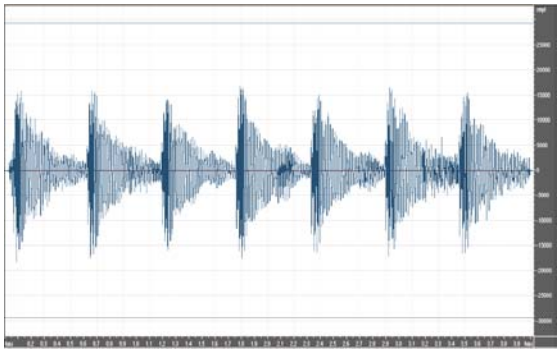

圖 11 “Green Light I'm searching for you” 頻率低於 220Hz 的波形

5.2 背景音樂影響之削減

在我們做 MP3 之哼唱內涵式查詢時所面臨到最大的問題就是背景的音樂部分，因為當我們透過麥克風哼唱歌曲時我們並沒有背景音樂的伴奏，但是 MP3 的歌曲當中很少是沒有背景音樂的，而背景音樂中所造成最大影響查詢結果的莫過於是鼓聲之類的節奏性樂器，因為流行音樂為了要讓人們能夠朗朗上口刺激銷售量，因此大部分都會加入明快的節奏，但是這些節奏性的樂器的頻率並不會隨著歌曲旋律的高低起伏而有所變化，所以在同一首歌當中每一小間隔的時間就會產生一組節奏樂器的頻率能量。

例如在 5.1 節當中我們所提到的“Green Light I'm searching for you”這個 MP3 樂句，我們可以很清楚的發現很有規律的約每 0.57 秒中就會有一次的鼓聲，一共產生了七次的鼓聲，我們若是把這個 MP3 樂句分成了七個***樂音（Slots）***，在這七 Slots 中，每一個樂音 Slot 都包含了一個完整的鼓聲的頻率能量，現在我們把緊鄰的 Slot 做相減（Difference）的動作，我們稱此動作為***背景音樂削減（Background Reduction）***，而此頻率能量差值的特徵係數我們稱之為***背景音樂削減特徵向量（BFV，Background Reduction Feature Vector）***，如此一來我們可以得到六個 BFV，此六個 BFV 已經幾乎完全沒有鼓聲的能量在其中，因為透過背景音樂削減會把接鄰 Slot 中頻率與能量相同的聲波給移除掉，所留下來的 BFV 的意義就是各個 MDCT Frequency Line 在時間軸（Time Domain）上的能量變化情形，簡而言之就是代表著前後音調的變化。在我們一般的音樂當中，音調都是會有高低起伏變化的，因此透過背景音樂削減的方法只會移除掉一些頻率、能量不變的一些節奏特徵，對於歌曲原本旋律的特徵影響並不大，故可以維持一般歌曲的準確率又能夠加強背景音樂很強之歌曲的準確率。

5.3 MP3 特徵向量

在這一小節中，我們將定義 MP3 樂句的特徵向量，此特徵向量乃是根據前面所介紹的 MP3 編碼原理，以及我們所介紹的 MDCT 頻率能量係數的特性，還有 MP3 音樂物件的特性所產生出來足以代表整個 MP3 樂句的內涵。

5.3.1 MP3 音樂物件的組成結構

如圖 12 所示，每一首 MP3 音樂物件都可以透過切割（Segmentation）的方式將其切割成一句一句的 MP3 樂句，此 MP3 樂句也就是我們之前所定義的哼唱式搜尋的最小單位。在 MP3 的標準中，MP3 音樂物件是由許許多多的 MP3 框架（Frame）所組成的，並沒有明確的定義圖中所謂的 MP3 樂句與 MP3 樂音（Slot），但以人類聽覺的角度來看 MP3 的音樂物件，我們直覺的會把他分成一句一句的 MP3 樂句，而每一句樂句當中會有數個至數十個音符，因此我們可以將一個 MP3 樂句細分成若干個 MP3 樂音，而此每個樂音當中又包含了若干個 MP3 框架，這若干個數目取決於 MP3 樂句所切割的長度以及我們將此樂句分成幾個 Slot。

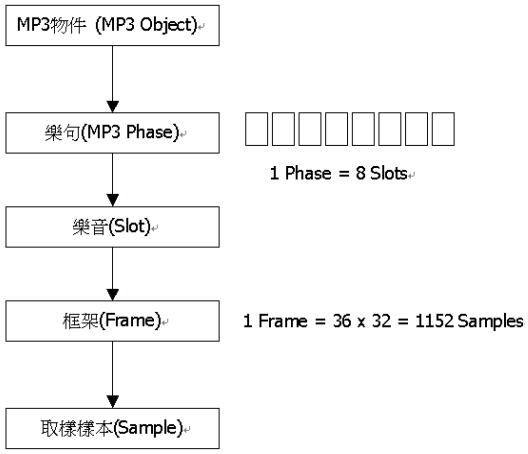

圖 12 MP3 物件組成架構

5.3.2 哼唱頻率之組成

在這一小節中我們將介紹人類哼唱歌曲所產生的頻率結構。當我們在哼唱某一個音符時，所產生的頻率並不會只是單一的頻率，事實上是一個複合頻率，這一個複合頻率乃是由一個基頻（Fundamental Frequency）與許多的

泛音（Overtone）所組合而成，泛音的頻率都是基頻的整數倍，這樣子所產生的聲音才會具有和諧度（Harmonic）。樂理上還有一個特色，那就是低頻的聲音會產生較多的泛音，且能量較大，而高頻的聲音基本上泛音很少，而且能量也很小。

圖 13 為我自己本身哼唱中央 Do（C5）的音，中央 Do 的音實際頻率為 523Hz，理論上應該落在第 14 條 Frequency Line 左右，我們可以看到在第 14 條 Frequency Line 有最大的能量值，但我們也可以從圖中發現基頻大約是發聲在第 3 條到第 4 條 Frequency Line 之間，在基頻的整數倍的頻率上我們也可以發現泛音的能量存在，但在高頻部分的泛音就不明顯了。

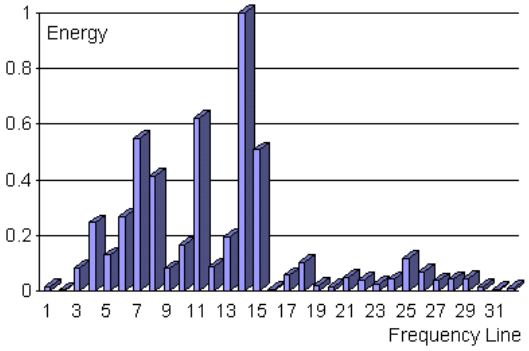

圖 13 哼唱中央 Do 的音

5.3.3 人類歌聲頻帶之擷取

前面我們提過透過擷取 MDCT Frequency Line 所輸出的係數我們可以得到 MP3 樂句中每個框架在各個頻帶上的頻率能量值，所有頻帶的頻率總範圍從 0Hz~22.05KHz，這是根據人類聽覺模型所訂定出來的範圍，但並不是我們所聽得到的聲音我們就能夠透過聲帶發聲出來，我們人類所能夠發聲的頻率範圍乃取決於每個人聲帶的構造，例如有些人聲帶較富有彈性，因此他的音域就相對的較廣，可以發出比較高的聲音，但一般人的聲帶都是大同小異的，能夠發出低頻的聲音大約在 200Hz~300Hz 之間，而高音部分大約在 2K~3K 之間。不過我們這裡所要關心的並不光是人類聲帶所能發出的頻帶區域為何，更進一步的我們要知道一般我們在哼唱歌曲的頻帶範圍，並且以此特徵來設定高頻與低頻的***截止頻率（Cutoff Frequency）***。在觀察了很多流行音樂的樂譜之後，我們發現大部分的流行音樂作曲在人歌唱的部分頻率都不會超過高音的 Re，因此我們將把***高頻截止頻率（Cutoff High Frequency）***設在第 30 條 Frequency Line 上。至於低頻部分由於牽扯到基頻通常比我們發出該聲音的音準還低上幾倍，而且其中還具有一些泛音，因此我們低頻部分不能參照樂譜的譜曲範圍，而參考自我們人類聲帶的特性，低頻的聲音大約在 200Hz~300Hz 之間，所以我們將***低頻截止頻率（Cutoff Low Frequency）***設在第 7 條 Frequency Line 上。

比較一下我們所擷取的人類歌聲頻帶的範圍與 MP3 音樂頻帶範圍來比較，我們可以發現我們並不需要完全的擷取 MDCT 每個頻帶上的係數，我們只要擷取一般人的主要唱歌的聲音頻帶範圍即可，如此一來可以減少非人聲部分對哼唱式查詢的影響，而且又可以增進查詢的速度。總之，我們將擷取出 24 條與人類歌聲最相關的 MDCT 頻率能量係數來作更進一步的分析。這高低頻擷取頻率的選取我們將再經過實驗的驗證，讓這個理論方法更具有公信力。

5.3.4 MP3 樂句特徵向量（MFV）

在前面我們曾介紹過把一個樂句分割成若干的 Slot 的概念，而 Slot 的個數主要是取決於樂句中音符的個數，在此我們實驗觀察了許多的 MP3 歌曲，我們將取 8 個樂音來代表一個 MP3 樂句中的 8 個音段，我們可以根據這 8 個音段的特徵值，來找出每個 MP3 樂句內音調變化的情形，以此來當作我們樂句特徵值的一部份。

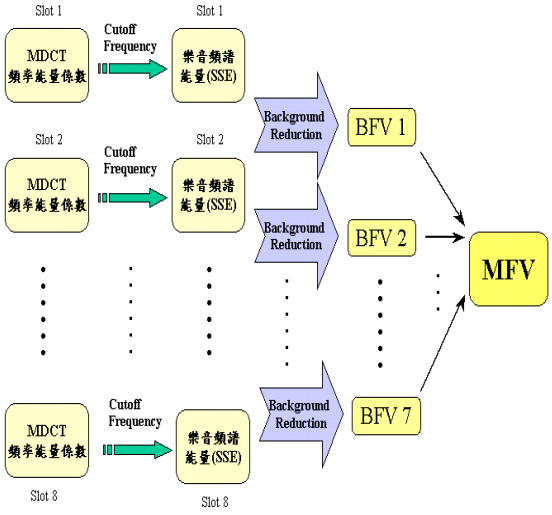

圖 14 MP3 樂句特徵向量產生流程圖

綜合前面的種種描述，我們在此定義了***MP3 樂句特徵向量（MP3 Phase Feature Vector，簡稱 MFV）***。如圖 14 所示，我們首先將 MP3 樂句分割成 8 個樂音（Slot），針對每個樂音中所有框架的 Frequency Line 7 ~ Frequency Line 30 部分做能量的加總與正規化步驟，如此我們可以得到每個樂音的***樂音頻譜能量（Slot Spectrum Energy，SSE）***。之後我們將接鄰的樂音頻譜能量做差值（Difference）的動作，因而得到 7 個***背景音樂削減特徵向量（Background Reduction Feature Vector ，***

BFV)，而我們的 MP3 樂句特徵向量即是由此七個背景音樂削減特徵向量所組成。

六、MP3 內涵式查詢比對

在我們整個 MP3 哼唱式查詢系統主要的架構分成了兩大部分，一個是 MP3 特徵資料庫的建立，另一個部分就是哼唱式查詢比對的部分。

6.1 MP3 特徵資料庫的建立

由於現在的一般時下唱片的發片速度實在是十分地快，幾乎每隔幾天就又有幾張的音樂 CD 要發行出來，因此如果把每首歌都加入了我們的 MP3 資料庫中，那麼我們的資料庫必定十分地龐大，如果我們在欲找尋我們要的 MP3 歌曲時，才對 MP3 資料庫中的 MP3 音樂物件擷取特徵值出來比對，那想必是一件相當耗費時間的事，而且當我們下一次要查詢另一首歌時，豈不是又要重新再對 MP3 音樂資料庫做一次特徵值擷取動作？！所以我們首先所要做的第一件工作那就是 MP3 特徵資料庫的建立。圖 15 就是建立 MP3 特徵資料庫的系統架構圖。

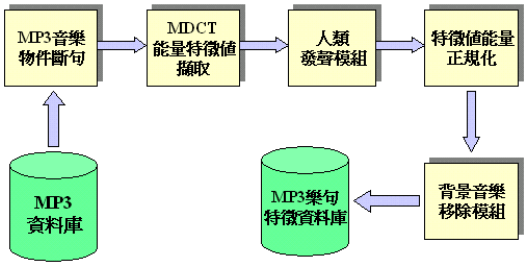

圖 15 建立 MP3 特徵資料庫的系統架構圖。

在整個架構中，首先我們將每個 MP3 音樂物件從 MP3 資料庫中取出來，並將各個 MP3 音樂物件切割成一句一句的樂句，此樂句就是我們先前所提過哼唱式查詢的最基本單位，然後將每個樂句擷取出它的 MDCT 頻率能量特徵值，經過了我們的人類聽覺與發聲模組，並將擷取的頻率能量係數正規化後，再經過我們的 Background Reduction 模組，而得到此樂句的特徵向量，並將之存入我們的 MP3 樂句特徵資料庫中。

6.2 哼唱式查詢比對

在我們系統中有了前面一節所提到的 MP3 樂句特徵資料庫之後，我們就可以對我們資料庫中的歌曲來做哼唱式的查詢。圖 16 就是我們的哼唱式查詢比對系統架構圖。

在哼唱式查詢比對系統架構上，前面擷取哼唱查詢樣本的特徵值部分，和前面擷取 MP3 樂句特徵值的方法是一樣的，在這裡相同的會把哼唱的樂句轉換成 7*24 維的 MP3 樂句特徵向量，然後把此特徵相量與資料庫中的所有 MFV 比對，把最相似的前幾名 MP3 排名出來，這個就是我們所搜尋到的 MP3 歌單。

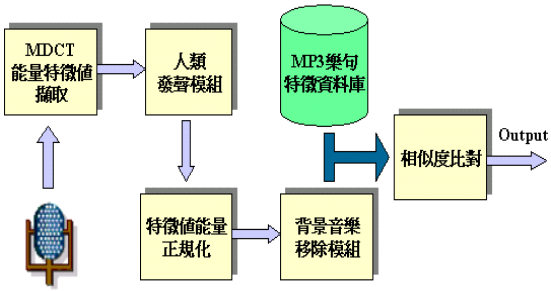

圖 16 哼唱式查詢比對系統架構圖

6.3 相似性測量模組

在比較使用者所哼唱的 MP3 樂句與資料庫中所有 MP3 樂句的相似度方面，我們採用了歐基理德距離（Euclidean Distance）來評估兩個 MP3 樂句之間的相似度。在比較兩個 MP3 樂句間的相似度時，我們會分別計算兩個樂句間各個樂音的歐基理德距離，然後再將所有樂音所計算得到的歐基理德距離加總起來，所得到的數值即為兩個 MP3 樂句間的相似度，數值越小，所代表的涵意即為兩個 MP3 樂句越相似。其公式如下：

$$Similarity = \sum_{i=1}^{7} dist(SSE_{S_i}^{P1}, SSE_{S_i}^{P2})$$

其中 $dist(PCV^{P_1}, PCV^{P_2})$ 為 P_1 的 SSE 和 P_2 的 SSE 的歐基理德距離

$dist(SSE_{S_i}^{P1}, SSE_{S_i}^{P2})$ 為 P_1 第 S_i 個 $Slot$ 和 P_2 第 S_i 個 $Slot$ 的歐基理德距離

其中歐基理 德距離的物理意義為 ：

假設在一個多維 的向量空間 上，有兩個點 α, β

其座標 分別為 $(a_1, b_1, c_1, \ldots)$ 和 $(a_2, b_2, c_2, \ldots)$（其中維度可以無 限延伸）

其兩點間的歐基理德距 離為

$$dist(\alpha, \beta) = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2 + \ldots}$$

我們的 MP3 哼唱式搜尋系統透過了上述的相似度公式即可比較出我們所哼唱的樂句與資料庫中的哪一個樂句相似性最高，並且可以依此排名最相似的 MP3 歌曲前幾名。

七、實驗

在實驗的部分，首先我們會先介紹我們實驗的 MP3 音樂物件資料樣本集合為哪些，還有我們針對資料庫中的哪些歌曲做查詢，而我們哼唱的是哪一句樂句。接著我們也會提到影響我們查詢準確率的因素有哪些，以及最後實驗的結果與探討。

7.1 實驗資料樣本

歌曲實驗資料樣本選取首先要具有客觀性與實用性，於是我們採取了全國最大的連鎖 KTV 業者，錢櫃 KTV 的點播排行榜中的歌曲，由於該業者每一週會統計點播排行前二十名的歌曲公布於網路上，於是我們將 2001 年 8 月 29 日到 2001 年 9 月 4 日這一週點播排行前 20 首的 MP3 歌曲完全加入我們的資料庫中，然後我們再隨意選取 30 首現在熱門的歌曲或是曾經流行過的歌曲加入我們的資料庫中，其中男女聲皆有，也有幾首台語歌在其中。

在我們的資料庫中總共的歌曲有 50 首 MP3 歌曲，總計 1619 句 MP3 樂句。我們將針對前面所提的 KTV 排行榜前 20 名的歌曲做哼唱式查詢。至於要哼唱哪一句 MP3 樂句來做為我們的查詢樣本，我們採用了每首 MP3 歌曲副歌中的第一句，原因就在於副歌是大家較易朗朗上口的一段。而副歌的定義就是 MP3 歌曲中，歌手在整首 MP3 歌曲中重複唱最多次的那一段詞曲。

7.2 影響查詢結果準確率的因素

在我們的實驗中，我們會探討幾個會影響 MP3 內涵式查詢的因素，其中包含了樂音（Slot）的個數、低頻截止頻率（Cutoff Low Frequency）、高頻截止頻率（Cutoff High Frequency）、排名列表（Rank List）的個數等，我們將以這些因素來比較有使用背景音樂削減（Background Reduction）方法與無使用背景音樂削減方法的查詢準確率。以下分別對這幾個因素做一些簡短的說明：

(1) 樂音個數：MP3 樂句乃是由 MP3 樂音所組合而成的，若樂音的個數太少，將會無法完全的表達出 MP3 樂句中音符變化的情形，相反的若是樂音的個數過多，也會因為分割的過細而失去了樂曲旋律的特質，並且效率也會因為特徵值增多而下降。

(2) 低頻與高頻截止頻率：雖然我們的背景音樂削減方法可以讓背景音樂影響減小，但是若我們一開始能針對非人聲的部分做一個移除動作，那將會使我們的準確率提高，而且特徵向量的維度也會下降。

(3) Rank 的個數：Rank 的個數代表著我們對搜尋到錯誤歌曲的容忍度，如果 Rank 值我們定為 3，那代表著如果搜尋到最相似歌曲的前三名有我們想要的歌曲，那即是代表著此次搜尋結果成功。因此，Rank 的大小值設定也是我們準確率的影響因素之一。

7.3 實驗結果

7.3.1 樂音個數的影響

在這裡我們要針對分割樂音個數多寡對於哼唱準確率的影響作分析，我們的實驗中除了樂音個數的改變之外，其餘的參數我們都將予以固定為我們之前理論上所假設的數值，低頻與高頻的截止頻率分別取在 Frequency Line 第 7 條以及第 30 條，Rank 大小為 3，實驗結果如圖 17。

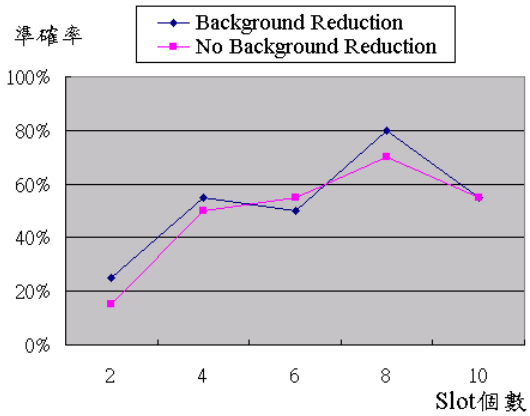

圖 17　Slot 數對準確率之影響

從圖 17 中我們可以看到當 Slot 的個數只取 2 個的時候，準確率相當的低，原因就在於 Slot 個數過少無法表現出樂句中音調的轉變。從圖中可以發現以 8 個 Slot 的個數可以得到較佳的效果。

7.3.2 低頻截止頻率的影響

在這裡我們要針對低頻截止頻率的設定對於哼唱準確率的影響作分析，我們的實驗中除了低頻截止頻率的調整之外，其餘的參數我們都將予以固定為我們之前理論上所假設的數值，高頻的截止頻率設定為第 30 條 Frequency Line，Rank 大小為 3，實驗結果如圖 18。

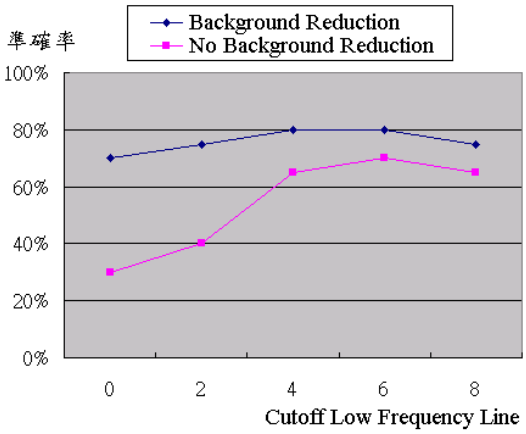

圖 18　Cutoff Low Frequency 對準確率的影響

從圖 18 中我們可以看到當低頻截止頻率取在第 0 條 Frequency Line 時（也就是所有低頻能量係數全部保留），沒有經過 Background Reduction 方法的準確率相當的低，而有經過 Background Reduction 的準確率卻一直保有一定的水準，準確率依然維持在 70%之上，其原因就在於透過了 Background Reduction 之後的特徵向量已經把許多背景音樂（大部分為 Bass 聲）的特徵移除掉了，因此可以維持如此高的準確率。

7.3.3 高頻截止頻率的影響

在這裡我們要針對高頻截止頻率的設定對於哼唱準確率的影響作分析，由於 MDCT 頻率能量係數高頻的範圍高達 22.05Khz，但是我們前面曾提過人類聲音的高頻部分大約在 2～3Khz 而已，因此在此高頻截止頻率的分析部分我們僅呈現出第 100 條 Frequency Line 前的分析（後面的頻帶對於準確率影響極微）。在實驗中除了高頻截止頻率的調整之外，其餘的參數我們都將予以固定為我們之前理論上所假設的數值，低頻的截止頻率設定為第 7 條 Frequency Line，Rank 大小為 3，實驗結果如圖 19。

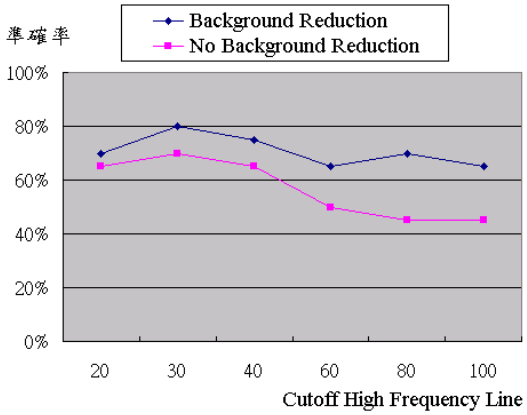

圖 19　Cutoff High Frequency 對準確率的影響

從圖 19 我們可以看出並不是高頻能量係數擷取的越多準確率就會越高，原因也是在於人們在歌唱的時候音域並沒有那麼廣，因此只要取到人們歌唱時發出高音的那個音域即可（大約在第 30 條 Frequency Line 左右），此時的準確率會較高，效率也會提升。

7.3.4 Rank 個數的影響

有了前面這些實驗結果之後，我們可以更加的確定我們的理論與方法事實上是滿正確，而現在我們將調整我們的 Rank 數，看看 Rank 數對於我們提出的方法其準確率影響為何，實驗結果如圖 20。在圖 20 中我們可以發現 Rank 數的增加可以增加搜尋到歌曲的機會，但 Rank 超過 5 之後，就已經趨於穩定狀態了，如果前五名還沒找到該歌曲，那就可能是哼唱失準或是該 MP3 音樂背景過於複雜等因素導致搜尋不到。

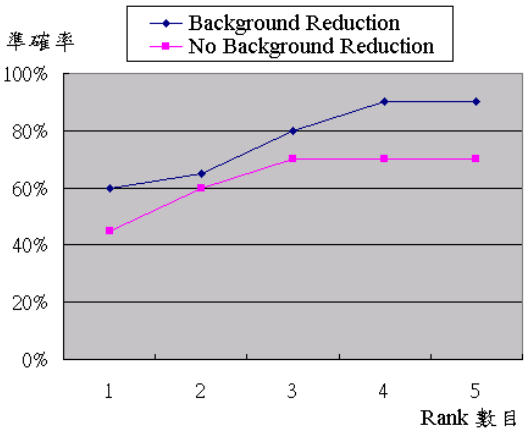

圖 20　Rank 數目對查詢準確率的影響

7.3.5 實驗結果總結

我們從以上的實驗可以發現 Slot 個數的多寡以及高低頻截止頻率的選擇都與我們先前所提出的想法相當接近，也證明了我們方法的可行性與正確性。另一方面，我們也可以看出有經過背景音樂消除在各方面的數據上都比沒有經過背景移除方法的好上許多，當資料庫中的歌曲增多的時候，相信彼此的差距將會拉得更加的大，因此顯示出此方法對於哼唱查詢準確率的提升有很大的幫助。在我們整個系統之下，若將 Rank 數取在 4，我們的準確率將會高達 90%，這樣的準確率應該是相當令人滿意的。

八、結論與未來工作

在 MP3 哼唱式查詢方面，最大的問題就在於 MP3 音樂是包含人聲與背景音樂的，背景音樂會使得 MP3 內涵式查詢的準確率變得很不穩定。在本篇論文中，我們提出了一個對 MP3 音樂物件分析的方法，透過對 MP3 的內涵作分析，擷取出其本身的內涵特徵值，經由我們的分析設定人聲部分的高低頻截止頻率之後，透過我們的背景音樂削減使得背景音樂的影響能夠減到最低，進而得到 MP3 樂句的特徵向量，我們就可以根據此特徵向量做相似度的比對，找到與我們哼唱相似的 MP3 歌曲，並以一種 Song List 呈現出相似歌曲的歌單。

在未來的研究方向，我們大致上有三個目標：

(1)　由於人類的聲音與聽覺的構造十分的複雜，我們將對此做更深入的瞭解，把我們 MP3 的特徵值加以濃縮成更小的特徵向量而不會影響我們的查詢準確率，甚

至尋求更高的準確率。

(2) MP3 的音樂物件基本上在其內涵方面還是保有許多樂理上的特性，我們也同時將針對樂理這部分做更深入的瞭解，運用更多樂理上的特色來增進我們MP3搜尋的準確率。

(3) 當資料庫日漸龐大了之後，我們就需要一個良好的索引結構來加快我們搜尋的時間，所以日後一個良好的索引結構也將是MP3內涵式查詢系統不可或缺的一環。

八、參考文獻

[1] Brandenburg, K. and G. Stoll, "ISO-MPEG-1 Audio: A Genereic Standard for Coding of High Quality Digital Audio," *Journal of the Audio Engineering Society*, Vol. 42, No. 10, Oct 1994, pp. 780-792.

[2] Chou, T. C., A. L. P. Chen, and C. C. Liu, "Music Databases: Indexing Techniques and Implementation," *in Proc. IEEE Intl. Workshop on Multimedia Data Base Management Systems*, 1996.

[3] Chen, J. C. C. and A. L. P. Chen, "Query by Rhythm: An Approach for Song Retrieval in Music Databases," *In Proc. of 8th Intl. Workshop on Research Issues in Data Engineering*, pages 139~146, 1998.

[4] Foote, J. "An overview of audio information retrieval", *ACM Multimedia Systems, Vol. 7,* pp 2-10, Jan 1999.

[5] Foote, J. "Content-Based Retrieval of Music and Audio," *Multimedia Storage and Archiving systems II, Proc. SPIE, Vol.3229,* pp 138-147.

[6] Ghias, A., Logan, H., Chamberlin, D., and Smith, B. C., "Query by Humming: Musical Information Retrieval in an Audio Database," *in Proc. of Third ACM International Conference on Multimedia*, 1995, pages 231~236.

[7] ISO/IEC 11172-3:1993, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s — Part 3: Audio."

[8] IEEE Std 1180-1990, March 1991, "IEEE standard specifications for the implementations of 8x8 inverse discrete cosine transform."

[9] Kosugi, N., Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, "Music Retrieval by Humming," *In Proceedings of PACRIM'99*, pages 404-407, IEEE, August 1999.

[10] Kosugi, N., Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, "A Practical Query-By-Huming System for a Large Music Database," *In Proc. ACM Multimedia 2000*.

[11] Liu, C. C., A. J. L. Hsu, and A. L. P. Chen, "An Approximate String Matching Algorithm for Content-Based Music Data Retrieval," in *Proc. of IEEE Intl. Conf. on Multimedia Computing and Systems,* 1999.

[12] Liu, C. C. and P. J. Tsai, "Content-Based Retrieval of MP3 Music Object," to appear in Proc. of the ACM Intl.Conf. on Information and Knowledge Management, 2001.

[13] Mo, J. S., C. H. Han, and Y. S. Kim, "A Melody-Based Similarity Computation Algorithm for Musical Information," *in Proc. of Knowledge and Data Engineering Exchange Workshop（KDEX'99）*, pp. 114 –121, 1999.

[14] Martin, K. D., E. D. Schrirer, and B. L. Vercoe, "Music Content Analysis through Models of Audition", *ACM Multimedia '98 Workshop on Content Processing of Music for Multimedia Applications*, Bristol UK, 12 Sept 1998.

[15] Melih, K., R. Gonzalez, and P. Ogunbona, "An Audio Rspresentation for Content Based Retrieval," *1997 IEEE TENCON – Speech and Image Technologies for Computingand Telecommunications*.

[16] Noll, P., "MPEG Digital Audio Coding," *IEEE Singnal Processing Magazine*, 1777.

[17] Pan, D., "A Tutorial on MPEG/Audio Compression," *IEEE Multimedia Magazine,* Summer 1995, pp. 60-74.

[18] Pfeiffer, S., S. Fischer, and W. Effelsberg, "Automatic Audio content Analysis," *ACM Multimedia 96*, Boston MA USA.

[19] Rolland, P. Y., G Raskinis, and J. G. Ganascia, "Musical Content-Based Retrieval: an Overview of the Melodiscov Approach and System," In *Proc. ACM Multimedia 99*, pages 81-84, November 1999.