具備車載背景音濾除機制之哼唱查詢式音樂播放系統

黃大育
國立屏東科技大學
資訊管理系
碩士生
M9756038@mail.npust.edu.tw

吳育澤
國立屏東科技大學
資訊管理系
碩士生
M9856030@mail.npust.edu.tw

劉寧漢*
國立屏東科技大學
資訊管理系
副教授
gregliu@mail.npust.edu.tw

摘要

隨著音樂播放器的進步，讓車用娛樂得以更為實用，但既有的車用播放器仍須手動選歌，為提升便利與維持行車安全，本研究使用哼唱查詢式音樂檢索法提供更為便捷的選歌方式，並針對車載環境設計一個對車載背景音做兩層過濾的機制，本研究以快速傅立葉轉換求取頻譜，並與背景音樣本頻譜進行相似度計算，先以第一層過濾移除相似車載背景音頻譜的頻譜資料，接下來以倒頻譜求基頻，由第二層排除人聲頻率外的基頻，進而得出較為純淨的半音語句，再接續以 Dynamic time warping 法進行音樂樣本比對，找出具相近旋律者並播放之。經由實驗數據分析，以本研究設計的機制進行車載背景音過濾，並對系統做最佳設定，使其在有車載背景音之狀況下維持不錯的查詢準確率。

關鍵詞：哼唱查詢式音樂檢索系統、音高追蹤頻譜、Dynamic time warping

Abstract

Due to the developments of music players, the vehicle entertainments for music are more useful. However, most of current vehicle music players must depend on manual control or key in information to select object songs.

In this work, Query by Humming is used by the music play system in order to provide more convenient way for select songs. The designs of the system are focused in vehicle background noises, and there is a mechanism provides two-layer filters. In layer 1 of the mechanism, its work is main to use FFT to convert audio signal frames to spectrums, and makes similar measure with background noises samples.

In the layer 2, it converts FFT spectrums to Cepstrums to find fundamental frequency, and then it filters out some fundamental frequency values which exceed of vocal frequency range. By these two layer's filter's process, it can improve the input be more pure for querying. Finally, the system measures the similarity of the query melody string and sample melody strings by Dynamic Time Warping algorithm, and find the most similar one to play it from the music database.

According the result of experiment, this research optimizes the QBH (Querying by Humming) system by our setting and the design of background noises filter. It has shown that this system can maintain well accuracy of querying in the vehicle environment.

Keywords: Querying by Humming System, Pitch Trace, Spectrums, Dynamic Time Warping

1. 前言

拜數位音樂播放器，與無線網路的技術發展之賜，聆聽音樂已不再受歌曲數量、設備容量空間、播放地點等限制，縱使在行車當中，車內使用者依然可透過手機、手提電腦，甚至是可攜式音樂播放裝置，透過當今發達的無線網際網路，立即進行下載與播放音樂。雖然音樂網站或資料庫提供便利的查詢服務，卻因現存歌曲數量為數過於龐大，若沒有掌握該歌曲的充分資訊，例如歌曲名稱、歌手資訊，或專輯名稱等歌曲相關資訊，此時使用者想要立即尋找與播放就顯得相當困難，而就以上看來，即便是使用方便輸入資訊的電腦設備，欲進行立即的歌曲找尋時，也不甚快速便利，更遑論使用者在行車當中，能輸入資訊並輕鬆與立即地找到想聽的音樂，而在另一方面，當使用者在行車當中，若頻頻以手動操作來選歌，也容易造成注意力不集中，而嚴重影響到行車安

*通訊作者

全。本研究鑒於安全上之考量，於設計之車載音樂播放系統使用較為直覺的操作方式--哼唱查詢式(Query By Humming)音樂檢索法。近年來，哼唱查詢式音樂檢索法，逐漸受到研究者的重視，QBH 是一種基於內容式之音樂檢索系統，其檢索方式有別於關鍵字音樂檢索需手動輸入音樂歌曲資訊，哼唱查詢僅需使用者哼出的一段旋律，即可找出具有相對應的旋律之歌曲並播放之。

在第一套哼唱查詢研究問世以來，各種加快檢索方法不斷地被提出與改進，儘管比對方法日漸繁多與精進，然而多數哼唱查詢式音樂檢索之相關研究仍有相同之處，大致上皆是對查詢語句做音高追蹤，並產生旋律資料，與資料庫中的半音語句樣本，或與旋律變化輪廓，做逐一配對以進行近似度計算，進而找出相似程度高者，最後從資料庫取出對應歌曲資料並播放之。大致上而論，哼唱查詢僅以某片段旋律做為檢索依據，便可找尋使用者欲聆聽的歌曲，故使用者不需輸入太多歌曲相關的資訊，即便忘記歌曲名稱，只需對著麥克風哼唱出腦中記憶的音樂片段，便能讓系統從音樂資料庫中，找出與旋律相互對應的歌曲，達到不需動手操作就能找尋歌曲，故此音樂檢索方式可符合車載音響系統操作需具安全性之需求。

然而，在行車時的車載環境並非安靜無聲，綜觀目前哼唱查詢系統，大多實作於較無噪音干擾的環境中，即便實作於其他環境，也尚未有針對車載環境所做的系統設計，若以既有哼唱系統在行車環境中運作，將因為哼唱時不免因停頓而產生一小段空隙，使得哼唱查詢語句填入車載背景音，從中產生錯誤旋律資料，造成旋律中存在不少錯音及贅音，而對查詢的結果造成可觀的影響，故本研究將以車載環境背景音過濾為主要目標，進行哼唱查詢系統之設計。

為設計一個適用於車載環境的哼唱查詢系統，就必須具備一個有效的濾除背景音機制，以讓系統產生一個較為正確的查詢旋律，故本研究先以 Fuzzy c-means 對純人聲與車載背景音頻譜進行分群，其結果證實車載背景音彼此具有相似頻譜之事實，再根據這個事實，設計一個以濾除非人聲資料為基礎，並能有效濾除車輛背景音之兩層過濾機制，此機制分別於第一層過濾車載背景音近似頻譜資料，於第二層過濾人聲頻率外的基頻值。由實驗結果顯示，若將先後順序調換，對背景音處理效果與查詢結果準確率幾乎沒影響，但執行效率則不然，若第一層過濾優先執行，其效率確實優於第二層優先執行者。而經過兩層過濾，伺服端將半音查詢語句做前後半音差計算，轉換為半音差字串(或稱為音高輪廓)，並將其傳送至伺服端，以時間及取樣率容錯性高的 **Dynamic time warping 演算法**與音樂旋律半音樣本做相似度比對，並找出前幾名具有相似片段之歌曲以供使用者確認。

而在降低背景音干擾程度方面，由實驗結果顯示，相較於實作在較無噪音的室內環境，本研究中的哼唱查詢系統在車載環境下，準確度僅比一般無噪音環境稍差 4%，顯示我們設計的車載背景音過濾機制，即使在充斥著車載環境背景音的環境中，仍顯示能維持一定程度的準確率。

本文分為五個章節，在第 2 章節探討哼唱查詢系統之相關研究與音高追蹤與相似度比對方法，而第 3 章節為背景音過濾機制與系統設計，第 4 章為實驗數據與結果，第 5 章節則是結論與未來展望。

2.相關文獻

2.1 哼唱式音樂查詢系統

隨著音樂播放器與相關數位音樂科技的進展，音樂資訊檢索系統逐漸受到許多的研究與音樂播放應用改進與探討。MIR(Music Information Retrieval)主要分為內容式音樂檢索與關鍵字檢索兩個主要類型。基於關鍵字的音樂索引，僅以文字做為查詢的依據，其方法查詢動作並無參照音樂本身資訊，而有別於關鍵字音樂索引，內容式檢索[1]方法，基於音樂音訊資料本身內容，即運用音訊本身內容的資訊，如音色、音高、節奏，等數個音訊特徵值，以此作為系統檢索音樂時的依據，內容式檢索方法通常以多維度，多個不同型別的音訊特徵，對其做初步運算處理，以作為檢索動作的依據。近年來，內容式音樂檢索逐漸成為研究者所青睞的研究議題，其研究趨勢可說方興未艾，並且在網路與資料庫技術取得巨大的進展之際，因應資料庫日漸龐大，針對在歌曲找尋上逐漸不易的議題上，研究者們開始探索與提出新型態、直覺、具備互動，與人性化的操作方式，哼唱查詢式音樂檢索[2]便具備上述數個特點，此檢索法為了使一個哼唱音樂查詢系統可以更為客觀與準確，目前主要運用音訊資料裡的音高與旋律資料。哼唱查詢式音樂檢索系

統[3][7]，在查詢時，使用人聲作為輸入資料，並以使用者輸入哼唱查詢資料，盡可能在資料庫內最精確的音樂檢索與查詢動作，其主要基於音樂所具備的特質，使用者只記得某段旋律，即便是在使用者遺忘音樂大部分的資訊情形下，哼唱查詢仍可在大型音樂資料庫中搜尋，以提供使用者更為自動與便捷的音樂搜尋。Ghias[4]提出一個使用 Autocorrelation Function (ACF)追蹤音高，搭配音高旋律輪廓法的哼唱查詢系統，此系統並非音頻既有格式，而是使用一組符合來描述音高的變化情形。在 Zhu[9]等人的研究中使用動態視窗規劃法，以進行哼唱查詢字串比對，研究中使用 Dynamic time warping[6]作為近似度比對法，可以將兩個時間序列資料作伸縮配對，以達比對此類非線性資料，與最小誤差之目的。但待處理資料眾多時，若使用 ACF 會造成可觀的運算量，為了即時與快速辨識音高，Goffredo Haus[5]等人所發表的哼唱查詢系統，以 FFT 轉換求頻譜，並用倒頻譜求音高以達及時且精確的目的。

2.2 音高追蹤

音樂旋律是由一連串音高變化所組成的一長串語句，故多數哼唱查詢系統在查詢時，即依據查詢語句的音高變化，以找出相對應旋律，音高是人腦經由人耳聆聽與判斷，即聲音在一定時間內，所震動之頻率變化，一段音樂經過人耳聆聽後，人腦會將週期內的聲音片段，做出聲音高低的認知與主觀判斷，一般來說，要以人工方便辨別音高，是較為主觀且容易的，然而，若要以人工智慧或者電腦程式作客觀的音高判斷，這就必須使用找出一段聲音內的主頻以求取音高，一般來說，要得知音高大致上都使用基於頻域，或基於時域的音高追蹤法[8]來取得音高變化情形。

2.2.1 Autocorrelation Function(ACF)

ACF 演算法是一種基於時域的音高追蹤方法，依照週期長度來取決音框的時間，並決定此 ACF 運算的平移量為 m 次，在限定的平移量內對長度為 n 的音框做 n-m 次內積運算，以下為其運算公式:

$$ACF\ (m) = \sum_{i=0}^{n-1-m} x(i)x(i+m) \qquad (1)$$

ACF 每次對該音框做內積一次，就將音框平移一點，並在不斷地重複運算 n-m 次後，便求得到 n-m 個內積值。

$$Frequency = Fs/Index \qquad (2)$$

Frequency 代表所求基頻，*Fs* 代表每秒取樣頻率，*Index* 為位於音框串列，能量最高的位置，並如上式所示，選出其中最大的時間序列位置，將其除以每秒取樣頻率，以選出基頻即代表此取樣週期的音高之主頻，並將其除以每秒取樣頻率，進而得出基頻，由於演算法步驟較少，故 ACF 在實作上為較容易，在音高追蹤上也有不錯的準確率，然而計算的方式是對連續音框資料，做內積運算，造成運算量與運算時間隨著資料量成正比，因此會耗損大量設備運算資源與運算時間，所以 ACF 並不適合實作在需要執行即時音高追蹤，且處理能力有限的可攜式設備上。

2.3.2 Cepstrum

Cepstrum[9]是一種基於頻域的音高追蹤方法，一般流程是先將音訊資料切割成音框，將音框做前處理後，以 Fast Fourier Transform 做轉換運算，再對其取對數產生頻譜能量參數，最後將其值使用反傅利葉轉換，找出時域上最高能量位置，倒頻譜的一連串的處理程序如下所示：

$$Cepstrum(t) = rfft(\log(fft(|x(t)|)) \qquad (3)$$

Cepstrum 為倒頻譜值，*fft* 為快速傅立葉轉換，*rfft* 為反快速傅利葉轉換，$X(t)$則是單位時間函數，也就是單位時間內的音訊資料值，經過此過程求出一個倒頻譜 w_{ceps}，再求得倒頻譜中最大值以得基頻。

2.4 Dynamic time warping(DTW)

Dynamic time warping[7]為一種動態視窗規劃法，其演算方法是為 Time warping distance 的變化型態，常用在語音辨識上，主要目的是為了加速運算與高容錯度，藉由拉扯兩筆須比對的序列資料，此演算法以遞迴的方式，將查詢字串與樣本字串從頭至尾做比較，並與目前累積路徑成本相加，形成一個最佳路徑之最小成本值，以尋求一個最佳化的時間序列配對，才得以從兩任意非線性資料間求得正確的近似度。

2.5 分群法

在音訊相關研究中，為了做後續有效分類，或事先觀察資料是否符合實驗預期，而使

用分群法做預先評估，計算資料點彼此的相似度，並將數個相似者置於同一群體中，以產生相近者同群，資料相異者則於不同群，並顯現於分群分佈結果中。但用於多維度資料辨別應用中，例如在界定資料類別時，其目標資料為其界線上較為模糊音訊資料，若使用 k-means 將多維度音訊特徵值分群後，點與群心具備了絕對關聯的特性，造成無法進行更為細部的觀察與進階的分析。為解決上述問題，就必須使用模糊分群法，以讓分群結果提供更多細節，以下為 Hard clustering 與模糊分群之比較圖：

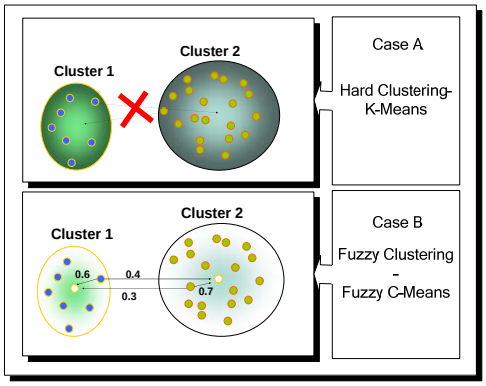

圖 2 Hard clustering 與模糊分群比較圖

如圖 2 所示，Case A 中的分群法，屬於 Hard Clustering 類別，在其對資料分群後，僅能知道其分布到哪一群，而無從得知與另一群的接近程度，而 Case B 則顯示，模糊分群對資料分群後各點仍與各群心存在著隸屬程度，而非僅能知道分到哪一群，故有利於後續資料細部分析與延伸運用。

2.2.1 K-Means

k-Means 是一種由 J.B.MacQueen 在 1967 年所提出的群聚法，其基本型態是隨機產生各群組中心點，並計算群中心與其他資料點的距離，遞迴地作疊代與重新產生群中心，直到聚合度無法再縮小為止，以下為目標函式：

$$J = \sum_{i}^{n}\sum_{j}^{k} dist(x_i, c_j) \tag{4}$$

$$dist(x_i, c_j) = \sqrt{(x_i - c_j)^2} \tag{5}$$

在以上式子中，$dist(x_i, c_j)$為 x_i 資料點與群心 c_j 間的歐幾里得距離，k-means 目標函數 J 可使用歐幾里得距離作全部資料點到各群群心距離的累加，進而得出其聚合度。

2.2.2 Fuzzy C-Means

有別於一般硬式分群，1981 由 Bezdek 所提出模糊分群法 Fyzzy C-Means 主要特性在於具有一個歸屬度矩陣，矩陣內的歸屬程度值能充分表達出資料點與群心的關聯性，故能達到模糊分群的目的。大致來說，FCM 之運算式與 k-means 相似，分群訓練流程亦是遞迴數次至聚合度無法再縮小為止，FCM 中的 Wt_{ih} 值，即單一資料點 h 之於 i 群心隸屬度，其隨機值介於 0~1 間，並如式子 6 所示，其對 c 群隸屬程度之加總值為 1。

$$\sum_{i=1}^{c} Wt_{ih} = 1 \tag{6}$$

將 Wt_{ih} 代入下式，以計算各群之群心值，由於代入值屬隨機產生的性質，故群心的產生規則亦具隨機性質，而計算群心的公式如下所示：

$$V_i = \frac{\sum_{t=1}^{n}(Wt_{ih})^F x_t}{\sum_{t=1}^{n}(Wt_{ih})^F} \tag{7}$$

上式中，X_t 為資料點數值，F 為模糊程度，F 值通常設為 2，V_i 則是所求之群心值，FCM 演算法接續利用隨機群心值計算出一個新的 Wt_{ih} 值，公式如下：

$$Wt_{ih} = \frac{1}{\sum_{r=1}^{c}\left(\frac{dist(x_i, v_h)}{dist(x_i, v_r)}\right)^{\frac{2}{F-1}}} \tag{8}$$

接下來的步驟，將 Wt_{ih} 代入 FCM 目標函式，求取聚合度的目標為下：

$$Min\ J_m(Wt, v) = \sum_{i=1}^{c}\sum_{h=1}^{n}(Wt_{ih})^F dist(x_h, v_i) \tag{9}$$

在上式中，$dist(x_{h,}, v_i)$為群心與資料點間的距離，是以歐幾里得距離公式作為運算式，求出兩者間的差距。在求取歐幾里得距離，經由上式作與 Wt_{ih} 之後，並累加 c 群與 n 筆資料值，而得到目標函數所求之值，也就是所謂的 FCM 聚合度。在第二次後的遞迴運算後，演算法將比較此次聚合度值是否較上回來得小，若無法再次縮小，即停止運算，並以此次分群結果做為分群數據。

3. 系統架構

本研究所開發的哼唱查詢式車載音樂播放系統，首先錄下哼唱歌聲，以一個非人聲資料為過濾為主軸，並針對車載背景音的兩層過濾

機制做車載背景音過濾動作，此機制分別於第一層過濾車載背景音近似頻譜資料，於第二層過濾人聲頻率外的基頻值，接著進行特徵轉換成中介格式，並經由 3.5G 無線網路傳輸資料至哼唱查詢伺服器內，最後由伺服器進行將音樂索引與查詢樂句做比對，以找到與查詢樂句相對應歌曲，並回傳至客戶端播放。

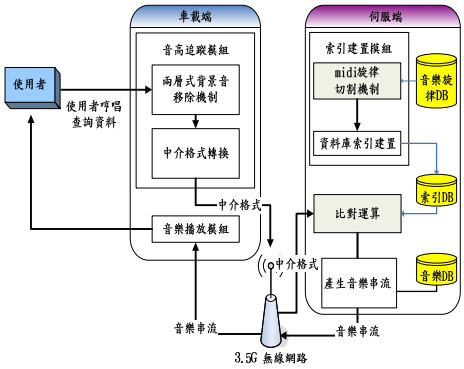

圖 3 系統架構圖

3.1 頻譜分群與過濾範圍設定：

以人耳聆聽，來判別車載背景聲與人聲是相當容易且明確，但若想要以電腦系統作辨認，就必須利用到聲音頻譜，因為每種不同的聲音皆有不同的頻譜，故以音訊頻譜資料作為分辨的依據，即可辨別出不同特質的聲音。一般行車時，車輛運行時所產生的車載背景音，主要組成為車輛頻繁的引擎震動聲，而具有特定且重複的聲音特質。為了取得其相關數據作為濾除車載背景音的依據，在設定過濾機制前，我們先要驗證車載背景音頻譜是否與人聲頻譜間是否具有不小的差異度，方能定義濾除門檻範圍。在資料探勘領域中，分群法能夠依照資料的差異程度，清楚顯示資料的分佈狀況，而分群法中，以模糊分群概念為基礎的 Fuzzy C-means(FCM)分群法有別於一般 Hard clustering，例如:k-mean 那樣，在分群結束後，某一群內的成員資料點，就與它群的群心，毫無任何關聯性可言，相較之下，FCM 有較為詳細的分群細節，即便是已將資料分群，但系統仍能掌握各個資料點，某資料點與多個群的群心之隸屬程度與關係，如此便可得知其差異及相似程度。以下我們使用 FCM，進行兩種聲音的分群，並對其分群結果作觀察。

本研究找來十個不同錄音來源者，其中男女各佔一半，取樣率為每秒 8000 次的純人聲錄音樣本，與同樣取樣頻率，在上述所提及的 10 輛車，在車輛行進時所錄下的背景音錄音檔，以長度為 256 的音框為分群之單位資料點，取此長度以確保人聲頻率範圍會落在其週期範圍內，其它設定如下表所示：

表 1 音訊資料設定

資料特徵	人聲	車載背景音
取樣率	8k/s	8k/s
音框點數	256	256
重疊點數	128	128
FFT 頻譜點數	256	256
樣本取樣時間	12 秒	12 秒

在做分群動作前，先將樣本資料以音訊常用的預處理，並以音框資料與漢明窗作相乘動作，接續將以快速傅立葉轉換(FFT)，目的是將原始資料轉換為頻域為主的頻譜資料，以方便人聲/車載背景音分群。為了觀察其分群狀況，我們將人聲與車載背景音分別轉成 FFT 頻譜，將 10 輛車的車載背景音，與 10 個人聲頻譜的 FFT 能量頻譜做平均值，並依照 1~10 與 11~20 將 10 輛車分別在暫存置入不同車載背景音平均能量頻譜與純人聲錄音頻譜資料，以方便觀察其分群情形，其實驗數據如下表所示：實驗數據如下表所示:

表 2 各頻譜與各群的隸屬程度

	對群一之隸屬度	對群二之隸屬度
1	1.0000	0.0000
2	1.0000	0.0000
3	1.0000	0.0000
4	0.9985	0.0015
5	1.0000	0.0000
6	1.0000	0.0000
7	0.9993	0.0007
8	1.0000	0.0000
9	1.0000	0.0000
10	1.0000	0.0000
11	0.9888	0.0112
12	0.0055	0.9945
13	0.0063	0.9937
14	0.0017	0.9983
15	0.0024	0.9976
16	0.0024	0.9976
17	0.0000	1.0000
18	0.0000	1.0000
19	0.0000	1.0000
20	0.0006	0.9994

於上表中，車載背景音平均數所在的資料頻譜資料(1~10)，全都落於同一群，而暫存檔中，純人聲頻譜資料部分，(11~20)頻譜資料代表點幾乎也落在同一群，由此分群結果所顯示，車載背景音震動聲較為固定，其頻譜穩定變化小，而人聲頻譜變化程度則稍大，導致仍

有一個資料頻譜平均值落於它群，此頻譜為代號 11 的人聲頻譜，其平均頻譜值最接近人聲，由上述可知，儘管在廠牌與車況不同的情況下，車載背景音頻譜與人聲頻譜具有明顯的不同，若以最接近背景音的人聲頻譜作為交界處，如此既不會遺漏過濾背景音，又能避免誤判而刪除人聲頻譜，此交界處的定義式如下所示:

$$Hum_{\min} = Min(\sqrt{\sum_{j=1}^{256}(Hum_{kj} - Noiseavg_j)^2} \quad (10)$$

本研究蒐集 n 個人聲頻譜樣本，Hum_k 為人聲樣本的頻譜值，k 可代入 $1\sim n$，而車載背景音頻譜樣本因為彼此相當接近，故以車載樣本頻譜能量平均值 $Noiseavg$ 替代其他車載背景音頻譜值，我們使用上式來計算，找出與 $Noiseavg$ 頻譜距離值最小者，即 Hum_{min}，並以 Hum_{min} 作為在車載背景音樣本與人聲樣本之間的初步的邊界點，而在定出了邊界點後，我們便能初步設定出一個濾除的範圍大小，此範圍大小如下所示:

$$Threshold = \sqrt{\sum_{j=1}^{p}(Hum_{\min,j} - Noiseavg_j)^2} \quad (11)$$

$Threshold$ 為 Hum_{min} 與 $Noiseavg$ 間的歐幾里得距離，當任何頻譜與 $Noiseavg$ 頻譜間的距離小於 $Threshold$ 值，此頻譜將被判定是車載背景音頻譜並移除，判別示意圖如下:

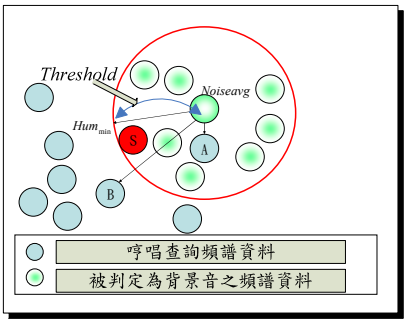

圖 4 背景音判別範圍示意圖

圖 4 中的圓半徑為 $Threshold$，其值是 Hum_{min} 與 $Noiseavg$ 間距離，圖中的背景音過濾以 A,B 兩頻譜值為例，A 頻譜與 $Noiseavg$ 間距離小於 $Threshold$，系統將其視為背景音並濾除，而 B 頻譜與 $Noiseavg$ 頻譜值之間距離大於 $Threshold$，故系統將其保留。縱使上述的過濾方法，能夠初步濾掉部分背景音，但此過濾範圍並非最佳設定值，為了使過濾效果更佳，我們對式子 11 加入了一個 w 數值，並改寫為式子 12，使其可調整判定標準權重:

$$Thresholds = W * \sqrt{\sum_{j=1}^{p}(Hum_{\min,j} - Noiseavg_j)^2} \quad (12)$$

w 其預設值為 0.5，更動其 w 值即可調整判別寬鬆程度，若其值越大，右式值就越大，此時，在判定上的門檻標準就越寬鬆，換句話說就是有可能因此濾除更多非人聲頻譜，但也可能將更多非背景音頻譜誤判並濾除，故 w 值並非越大越好。根據以上計算式，並以下列之不等式作來對濾除範圍作規範:

$$Dist(Query_x, Noiseavg) \le Thresholds \quad (13)$$

$Query_x$ 為哼唱查詢語句的頻譜，$Noiseavg$ 為車載樣本頻譜能量平均值，若左式小於右式，依此判定該頻譜值 $Query_x$ 為車載背景音頻譜並濾除。在加入權重 w 後，我們就能得到一個可變動的過濾範圍，其變動情形如下圖所示:

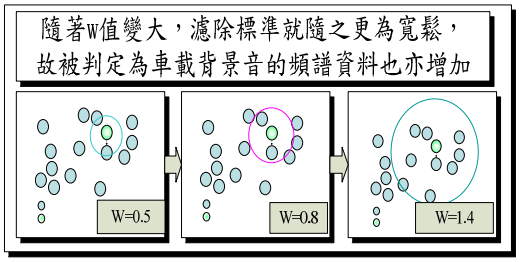

圖 5 過濾權重調整示意圖

如圖 5 所示，隨著 w 值變大，可能原先被判為人聲者，就有可能會被判為背景聲;而相反的，若 w 值越小，原先應該被濾除的背景音頻譜仍舊殘存，若適當地調整 w 值，便能夠具後續辨別調整空間，以判定標準最佳化來降低誤判率。

3.2 車載端

車載端部分主要用於收錄哼唱聲與做前處理，並轉換使用者所哼唱的音訊資料為中介格式，以供系統做比對。音高追蹤模組包含背景音移除機制以及半音與中介格式轉換運算兩個部份，背景音移除機制主要是設計用以移除部份非人聲頻率與車載背景聲，而中介格式轉換將已初步處理的音訊資料進一步轉換成能與音樂樣本比對的中介格式(半音差字串)，以下將介紹音高追蹤模組的兩個主要單元。

3.3 背景音移除機制:

為移除影響哼唱查詢的車載背景音，我們設計一個分別以兩層過濾，車載背景音與非人聲基頻的背景音過濾機制，此設計之主要目的是為了產生一個較為正確哼唱查詢字串，此兩層式過濾機制示意圖如下所示：

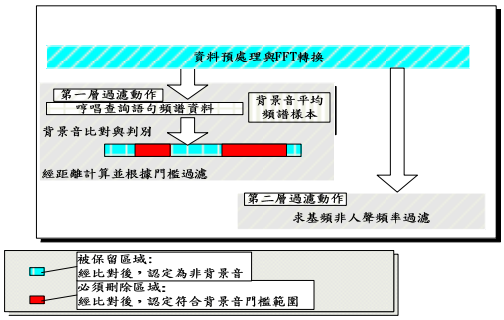

圖 6 兩層式車載背景音過濾機制運作示意圖

如圖 6 所示，本研究中的兩層式背景音過濾機制先對資料做預處理以及進行快速傅立轉換(FFT)之後，可選擇執行第一層過濾機制，或者選擇第二層過濾機制來做優先過濾動作，其機制分別為針對頻譜資料與基頻資料做過濾，且不論何者優先執行，皆必須在一層過濾完成後，執行另一層過濾動作，方能將人聲頻率內外的車載背景音做最大限度的濾除，並藉此取得較為純淨的哼唱查詢語句。

第一層過濾機制:

此層的過濾機制主要步驟為:先對原始音訊資料作音框化，並利用快速傅立葉轉換後產生的頻譜，再以本文中 3.1 節所定義車載背景音頻譜過濾門檻與過濾機制設計來進行過濾與移除，最後將與車載背景音樣本頻譜相近者移除，達到降低部份車載背景干擾的效果。

第二層過濾機制:

一般在音高追蹤常用的 ACF 法，在資料量大時會消耗大量運算效能，因此我們先使用倒頻譜(*Cepstrum*)法建立頻譜，處理流程為:先將音框資料以 Fast Fourier Transform 做轉換運算，再對其取對數產生頻譜能量參數，最後將其值使用反傅利葉轉換，完成倒頻譜處理流程，其流程如下式:

$$Cepstrum(t) = \frac{1}{N}\sum_{t=0}^{N-1}\log(\sum_{n=0}^{N-1}x(t)W_N^{tn})W_N^{-tn} \quad (14)$$

$$W_N = e^{-j(2\pi/N)} \quad (15)$$

Cepstrum 是資料序列中 *t* 的倒頻譜值，*j* 為虛數值，*N*為音框長度，首先將資料做音框化，產生音框長度為*N*的 *x(t)*時間函數，再以 W_N 將 *X*(*t*)與之相乘做視窗化，以求得傅立葉係數，經由以上過程算出音框內的倒頻譜值，得出一個 ω_{ceps} 頻譜參數，再求得音框的倒頻譜參數中的最大值，並求得基頻，如下所示:

$$Index = Maximum(\omega_{ceps}) \quad (16)$$

$$Frequency = Fs/Index \quad (17)$$

其中利用倒頻譜所產生的頻譜，將 w_{ceps} 頻譜中最大值視為 *Index*，*Frequency* 為基頻值，*Fs* 為每秒取樣點數頻率，由此算式便能求得該音框的基頻值。接續進行第二層針對人聲門檻外基頻的濾除動作，將 40*HZ*~1000*HZ* 之外的基頻值濾除，以完成第二層濾除動作。

過濾優先安排策略:

由於兩層過濾皆以 FFT 將音框資料轉為頻譜再進行後續處理，故由哪一層先執行過濾，再交由另一層繼續處理，對哼唱查詢結果也較無影響，但由於倒頻譜的時間複雜度，至少為 FFT 的運算次數(*nlogn*)的兩倍以上，若車載環境中的背景音較多屬於於人聲頻率範圍內的，那麼以第二層過濾為優先者，將付出龐大的運算浪費之代價，並拖慢執行效率，反之則不然，還是得端看實驗環境所造成的影響。

3.4 中介格式轉換機制設計：

由於查詢的對像為音樂資料，使用基頻比對不僅運算量大且較無意義，另考量查詢處理需於伺服器上執行，因此哼唱查詢語句之基頻將轉換為中介格式，以降低傳輸資料量並減少錯誤率。

Step 1 :半音轉換：

由於音樂大都是以固定音高組成(例如樂譜)，並非為任意音高，因此我們將查詢語句之頻率序列轉換為對應鋼琴琴鍵之 MIDI 半音值序列，下式將週期基頻值進行半音值轉換，以求得每個單位音的半音值 S，半音轉換公式如以下式子:

$$S = Round(\log_2(\frac{Freq}{440})\times 12 + 69) \quad (18)$$

Step 2 :轉換成音高差異字串並傳送

由於大部分的使用者不具有絕對音準，也就是起音之音高往往與欲查詢之音樂有差距，因此，我們將進一步把半音字串轉換為音高差異字串，也就是將一個音符之半音數值與前一個音符之半音值相減，所以半音值將轉為前後音高變化量，以提高比對容錯性。轉換後的音高差異字串將傳送至伺服器段進行查詢比對。

3.5 旋律索引建置模組設計：

由於主旋律是一般人記憶音樂的主要方式，如果能將主旋律及歌曲中令人印象深刻的片段

摘錄出來，並建立相關索引，將能減少資料庫搜尋比對之時間。由於在我們預設的音樂資料庫是包含 MIDI 格式檔及其對應的音訊檔案，為降低系統複雜度，我們將針對 MIDI 格式擷取其資料內容的旋律變化，並將其轉為半音差格式以方便比對。

3.5.1 查詢處理設計：

由於 Dynamic Time Warping（DTW）在用於近似度比對上，由於其具有較高的容錯性，可跳過某些錯誤的字串部份，故整串旋律字串中若有少音多音的情形，也對其比對結果較無影響，例如某位哼唱者本身對節拍不甚敏感，容易跟不上節奏，既使其音準不差，對哼唱系統哼出存有 Fa Me Me 三個音符的旋律，系統並以數字形式做轉換，將此字串轉為「4 3 3」，但因單音符停留時間長度與樣本旋律單位時間不對等，可能因哼唱者停留某幾個音符較久，造成「4 3 3」被記錄成「4 4 3 3 3」，故與樣本中的正確答案「4 3 3」做比對，會有時間序列上無法對配的問題，若使用 DTW 近似度比對，便能以其路徑規劃，對時間序列作最佳的調整，達到不同等長度的時間序列，不同時間點出現的資料可彼此做配對的目的，比對示意圖如下所示:

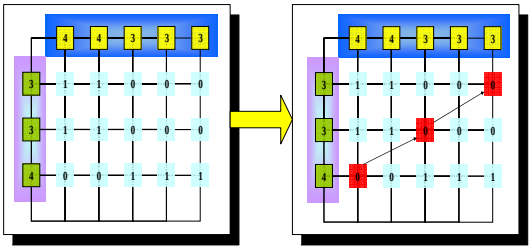

圖 7　DTW 比對示意圖

在示意圖的左半圖中，由兩字串間各元素的差距產生一個距離表格，右半圖顯示 DTW 由距離表格中，找出最佳路徑的情形，經近似度比對後得出查詢字串中的「4 3 3」與樣本字串「4 3 3」配對，並且由於哼唱音準方面沒問題，得出兩字串最短距離為 0，故即使音符時間配對不同時，仍能比對出近似者。

Step 1：DTW 比對哼唱字串與索引查詢結果字串將哼唱之中介格式串列與索引查詢出之音樂片段中介格式字串作 DTW 運算，DTW 比對關鍵旋律片段，我們將兩個向量查詢樂句字串與樣本句字串進行比對，使用 DTW 將查詢樂句與關鍵片段從頭至尾開始比對，距離公式使用歐幾里得距離，路徑選擇條件式如下所示:

$$\text{Distance}[i,j] = D[i,j] + \min\begin{Bmatrix} D[i-2,j-1] \\ D[i-1,j-2] \\ D[i-1,j-1] \end{Bmatrix} \quad (19)$$

上式當中，Distance[i, j]代表表格中目前 $D[0,0]$ 移動到 D[i,j]的必需花費，且 D[i, j]與 $dist(x_i, x_j)$ 等價，即 x_i，x_j 間的歐幾里得距離，計算距離公式如下:

$$dist(x_i, x_j) = \sqrt{(x_i - x_j)^2} \quad (20)$$

Step 2 :依近似程度排名

利用 DTW 法則計算出之兩字串總編輯成本；計算各備選歌曲之相似度後，求出前幾名，備選者並將其歌名回傳至車載端，由使用者決定是否為查詢答案及是否需播放。

4.系統介面與實驗

我們在系統設計完成後，實際建置了車載哼唱查詢系統，並將實驗硬體環境與介面分述如下：

4.1 實驗環境

在本研究中，我們所使用的車載端的開發平台和設備如下表：

表 3　車載端環境

名稱	內容
型號	BENQ Joybook Lite U101
中央處理器	Intel® ATOM™ N270 @ 1.6GHz
記憶體	1024MB
作業系統	Windows XP Professional

4.2 系統介面

本研究實作的是車載哼唱系統，力求介面設計著重於便利並易於操作，故僅提供基本的哼唱查詢功能與相關設定，以下為系統介面圖:

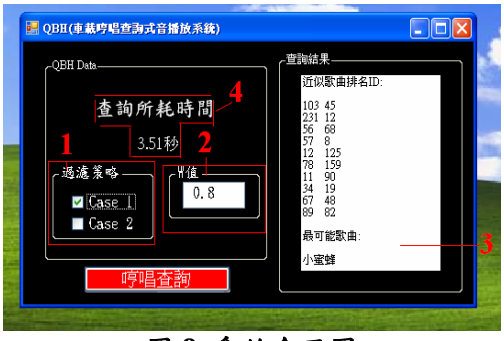

圖 8　系統介面圖

上圖為車載哼唱查詢系統畫面，我們將這 4 區塊分別介紹：

區塊 1–過濾策略設定：

提供過濾策略設定，勾選 Case 1 或 Case 2 來進行背景音過濾，全部勾選被取消，系統則判定為使用者取消兩層式背景音過濾機制，僅過濾人聲範圍外之基頻為主。

區塊 2–w 權值更動設定：

提供輸入框供 w 權值更動，以改變第一層過濾中，對車載背景音頻譜判別的過濾範圍，權值將影響音框頻譜資料的誤判率。

區塊 3–查詢結果：

由於車載音樂播放系統螢幕尺寸受限之故，設計上並不會列出超過最接近的前 15 名，本系統列出所有可能的近似歌曲，依照近似程度排列歌曲 ID，列出近似歌曲排名清單，以及顯示最相似者之對應歌曲的歌名，其查詢結果可藉由前近似幾名的設定來判定準確率。

區塊 4–查詢所耗時間：

顯示從哼唱查詢開始，到系統算出 k 個旋律之近似距離完成為止，經歷此一連串的處理程序所耗用之時間，可藉以判斷出系統與設定所帶來的效能之影響。

4.3 不同過濾策略下的查詢效率與效能實驗

經由以下連續數個實驗。實驗中的查詢準確度就是前 k 名的出現機率，例如我們於查詢 100 次數中，所哼唱的歌曲出現 50 次於近似度排名前 10，則說其準確率為 50%，出現次數越多則準確度越高，依此類推。我們以資料庫中 300 首旋律片段，以做為測試時比對用的樣本，且每項實驗皆各別於五輛不同廠牌的車內中，分別作系統測試。

本文中前面部份有提到，無論是以第一層過濾機制，或者第二層過濾機制來優先執行，其哼唱查詢準確度在理論上是相差無幾，而執行效率則會與之相反，為驗證並確認何者為確最佳的過濾策略，我們請來一位音準不算差，且學過鋼琴的同學，在車載環境中進行兩種過濾機制策略實驗，並對兩種策略各做 20 次的哼唱查詢，而得出以下實驗結果:

表 4 不同優先過濾機制策略下的效果與效率

	排名前 5	排名前 10	排名前 15	查詢完成平均秒數
Case 1	65.8%	71.4%	75%	3.3 秒
Case 2	67%	70.2%	74.5%	6.8 秒

以本系統基頻轉換值來觀察，車輛在運行時會產生連續介於 60~110hz 間的頻率值，我們由此推測，因其頻率介於 60~100hz 範圍之內，故車內多數背景音應為人聲門檻範圍內，若系統將第二層做為優先過濾機制，將會平白浪費運算資源，而且對哼唱查詢準確度毫無幫助。我們以實驗數據加以佐證，以第二層為優先的過濾策略，其準確度較差，且花費較多運算時間。

4.4 設定不同過濾權重 w 值之哼唱查詢實驗

改變 w 值將造成過濾範圍變化，一般來說過濾範圍越小就會遺漏許多背景音頻譜，而 w 值過大時，便會刪除不應該刪除的人聲頻譜，本研究對此做研究，找來了 5 位平常愛唱歌的朋友，對各 w 值設定，每人各進行 10 次的哼唱查詢實驗，得出了當 w=0.8 時，準確度最高的情形，而 w=1.4 時，準確度最低。

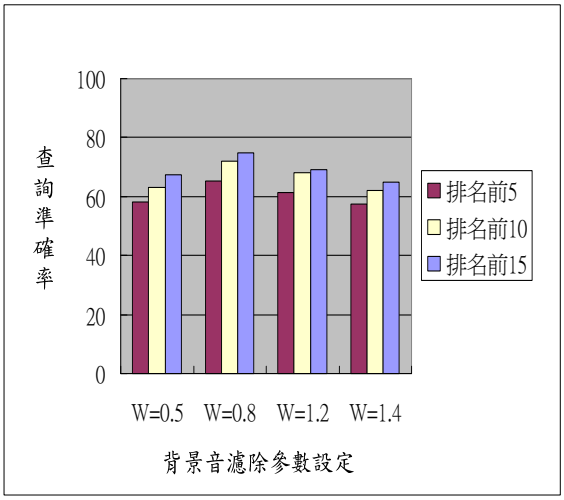

圖 6 在不同車載參數設下之準確率比較圖

圖 6 顯示，w 值並非越大越好，將其值設定過大，反而將應該作為哼唱查詢依據的人聲頻譜部分都刪除，造成查詢正確率不佳。

4.5 在不同環境下之查詢準確度實驗

此實驗是為驗證本研究所設計的車載背景音過濾機制之實際成效而設計，本研究以 10 名測試者，3 名有音樂相關背景，其他 7 名無此背景，受測者分別在室內與車載環境內，並個別進行 20 次哼唱查詢測試。實驗結果內容如下表示:

表 5 哼唱查詢準確度

	排名前 5	排名前 10	排名前 15
開啟過濾機制(室內)	71.4%	75%	82.4%
關閉過濾機制(室內)	69.4%	74.3%	81.4%
開啟兩層過濾機制(車載)	65.8%	71.6%	75%
未開啟過兩層濾機制(車載)	46.4%	49%	67%

由上表可得知，在車載環境下開啟兩層過濾機制，哼唱查詢各項排名項目的準確率部分，與室內環境測試者之成績相距不遠，反之未開啟過濾機制者於車載環境在準確度上便有不小的落差，結果顯示本研究之過濾機制具備一定程度的降低背景音干擾之能力。

5.結論與未來研究方向

相較於以往研究沒有針對車載背景音作過濾動作，其大多僅以人聲基頻範圍作資料過濾，與預處理作一個初步的訊號處理，在數個車載背景環境下，本研究利用不同種類的聲音，皆有不同的頻譜之特性，藉由預先錄製的背景音樣本與人聲樣本，並設定過濾門檻及排除非人聲範圍聲音，以進行背景音頻譜的過濾，利用既有的資料庫 300 首旋律，經過各項實驗的測試，與嘗試數個的過濾範圍，調整過濾規範標準，與車載背景音過濾策略，得出最佳的設定值，既使在車載環境下，各項目平均準確度仍僅下降 4%左右，而在車載環境中，並且無開啟車載背景音過濾功能的情形下，各項目平均準確度下降 17%左右，顯示我們設計的車載背景音過濾機制，即使在充斥著車載環境背景音的環境中，顯示仍然能維持一定程度的準確率。縱然目前是以預錄樣本對哼唱查詢資料作是否要移除的判斷，已能在防止車載背景音對哼唱查詢的干擾上，具有不錯的成效，但車載背景音畢竟是由數種環境音所組成，僅憑目前這混和聲音的頻譜樣本，恐不能將車載背景音濾除做到更為精進與有效，未來若能以基因演算法或其它分類法，將車載背景音混和頻譜細分為引擎聲、風切聲、甚至輪胎摩擦聲，並與配合數種雲端技術，例如網格運算快速將哼唱查詢語句與各種樣本頻譜作快速比對，將可為車載哼唱查詢甚至是語音辨識準確度上帶來更大的效益。

誌謝

本研究承國科會專題研究計畫(編號：NSC98-2200-E-020-004 及 NSC98-2221-E-020-025-MY2)補助經費，謹此誌謝。

參考文獻

[1] Bin Cui, H.V.Jagadish, Beng Chin Ooi, Kian-Lee Tan,"***Compacting Music Signatures for Efficient Music Retrieval*** ", EDBT'08, March25–30,2008.

[2] Erdem Unal, Shrikanth S. Narayanan, Elaine Chew," ***A Statistical Approach to Retrieval under User-dependent Uncertainty in Query-by-Humming Systems***",MIR'04, October 15- 16, 2004.

[3] Erdem Unal, Shrikanth Narayanan, Elaine Chew ,Panayiotis G. Georgiou, Nathan Dahlin, "***A Dictionary Based Approach for Robust and Syllable-Independent Audio Input Transcription for Query by Humming Systems***",AMCMM'06,October 27,2006.

[4] Ghias A., Logan J., Chamberlain D., Smith B. C., "***Query by humming-musical information retrieval in an audio database***", ACM Multimedia '95 San Francisco, 1995.

[5] Goffredo Haus, Emanuele Pollastri,"***An Audio Front End for Query-by-Humming Systems***", Procs ISMIR pp. 65-72 ,2004 .

[6] Jefrey Lijffijt1, Panagiotis Papapetrou, Jaakko Hollmén1,"***Benchmarking Dynamic Time Warping for Music Retrieval***",PETRA'10 June 23 - 25, 2010.

[7] Lei Wang1, Shen Huang, Sheng Hu, Jiaen Liang, Bo Xu1,"***An Effective and Efficient Method for Query by Humming System Based on Multi-Similarity Measurement Fusion***",IEEE 2008.

[8] Roger Jang, Shiu-Yun Cheng,."***An Evaluation of Query By Singing/Humming Based on Various Pitch Tracking Methods***", MS Thesis, National Tsing Hua University, Taiwan, 2008.

[9] Zhu, Y, Shasha D,"***Warping Indexes with Envelope Transforms for Query-by–Humming***", In Proceedings of ACM SIGMOD 2003.