「多模式音樂檢索」系統

李宏儒　許肇凌　王儀蓁　張智星

國立清華大學 資訊工程學系

{khair, leon, evawang}@wayne.cs.nthu.edu.tw，jang@cs.nthu.edu.tw

摘要

數位典藏服務是一門很新的領域，而將各種文化典藏數位化紀錄下來以利傳播及保存已經成為一種趨勢，並且能以簡單而自然的方式進行檢索更是一個重要議題。本研究將以『多模式音樂檢索』為例，提供過去我們在建立音樂資料庫、資訊數位化的處理以及資訊檢索流程時所獲得的經驗分享，期盼透過本研究能提供未來相關設計、開發以及應用數位典藏檢索時的參考，並且能與不同的典藏資料檢索整合，藉此發揮數位典藏在使用者服務功能上的價值以及提高數位典藏檢索的功能。

1.緒論

1.1 研究動機與目的

多模式的音樂檢索方式目前在國內和國外的研究並不多見，目前較具規模的大至分點說明如下：

1.1.1 英國 Southampton 大學 QBH 系統

近幾年來，幾乎所有有關音樂內容搜尋方面的研究報告都引用了英國 Southampton 在 1995 年 ACM 多媒體研討會中發表的論文[1]，這篇論文算是早期最具代表性的一篇報告。他們同時也開發了一套名為 QBH（Query By Humming）的系統，讓使用者可以透過麥克風哼唱來對音樂資料庫搜尋。他們透過自相關演算（Auto-correlation）來求得輸入聲波的基頻分佈圖（Pitch Contour），並將其轉成包含了 U（這個音比前一個音高）、D（這個音比前一個音低）、R（這個音和前一個音相同）的字串用以進行音樂資料庫的搜尋。不過可惜的是，他們並未發展出一套完整的音符切割程序，因此在使用 QBH 時，使用者必須自行分割音符，並未達到真正的自動化。當時在 Sparc 工作站上，整套 QBH 光在基頻分析就需要耗掉 20-45 秒，而音樂資料庫則包含了 183 首歌曲，僅僅只能算是在直覺式歌唱輸入音樂搜尋上邁出了第一步。

1.1.2 紐西蘭 Waikato 大學

紐西蘭 Waikato 大學的 Rodger J. McNab 算是近幾年來對「以歌選歌」發表過最多篇論文的專家，他們和紐西蘭數位音樂資料庫合作開發出了一套名為 MT（Melody Transcription）[2][3]的系統，藉著金-瑞賓勒演算法（Gold-Rabiner Algorithm）[5]找出輸入聲波的基頻分佈，並接著轉成標準音符表示。

接著，他們將 MT 結合數位音樂資料庫，開發成一套名為 MELDEX 的系統 [4]，讓使用者可以透過麥克風哼唱就直接達到搜尋音樂資料庫的目的。他們同時也將音樂資料庫的歌曲數目提升至 9400 首左右，正確辨識率約在 77% - 89%之間。不過 MELDEX 卻仍然帶有致命的缺點，因為他們無法正確地將音符切割開來，因此使用者在哼唱時，在音符與音符之

間，必須自行留下小小的間斷或多加入「滴」「答」聲，對於非專業的演唱者而言仍然相當地不便，也相當地不自然。

1.1.3 SONODA 系統

Sonoda[6]提出同時使用音高及音長來做有效率的檢索，系統先過濾非相似的資料，再將剩下的資料做比對；除此之外，他們的系統需要大量的記憶體運作，因此在實做上沒有很大的效率。

1.1.4 SoundCompass 系統

Kosugi[7]提出同時使用音調轉變及音調分佈來改良他們系統名為 SoundCompass 的效能，其系統中有 10086 首歌曲。但真正使用時需配合節拍器哼唱，這點相當不方便也不適合大部份的一般使用者。

1.2 『多模式音樂檢索』研究目的

Ghias、Sonoda 以及 McNab 所提出的方法都需要從音高向量中擷取出音樂特徵然後再加以比對，特徵的擷取或分段是很容易因為使用者的聲音輸入而造成錯誤；我們的『多模式音樂檢索』系統即針對以前舊有系統的缺失進行改良，期望給予使用者一個更自然、更人性化的搜尋介面，讓使用者能夠透過十分自然的哼唱、敲擊以及語音方式，達到搜尋音樂資料庫的目的。舉例來說，如果使用者的歌唱技巧不是很好，例如節拍可能過快或音調可能不準，『多模式音樂檢索』系統都能以彈性比對的方法，找出最有可能的歌曲，並依相似度高低進行排序；或是利用語音直接講出歌曲名稱、歌手名稱以及歌詞片段等直接檢索搜尋，找出內容相近的歌曲，並依其相近程度進行排序。

『多模式音樂檢索』系統最早研發的重點在於網路 KTV 點歌系統的應用，由於其自然且人性化的介面，提供使用者在資訊檢索上有多種模式輸入的選擇，因此有相當高的評價。有鑑於此，我們將其應用在數位典藏的方向列點說明如下：

1. 本系統應用在數位典藏系統，可以使檢索操作過程更為簡便且人性化，讓使用者能自行選擇最方便、最容易的方式來檢索數位典藏系統。
2. 本系統容易實做在網路系統上，只要放在網路伺服器上，再加上數位典藏的資料庫，就成為一個多模式的數位典藏搜尋引擎。使用者就能輕易地使用多種輸入模式在網路上檢索資訊。
3. 由於系統的優異檢索能力，應用在作曲分析上，不僅僅方便作曲家偵測仿冒，也更加方便作曲家對創作作品與典藏作品進行交叉比對和參考。

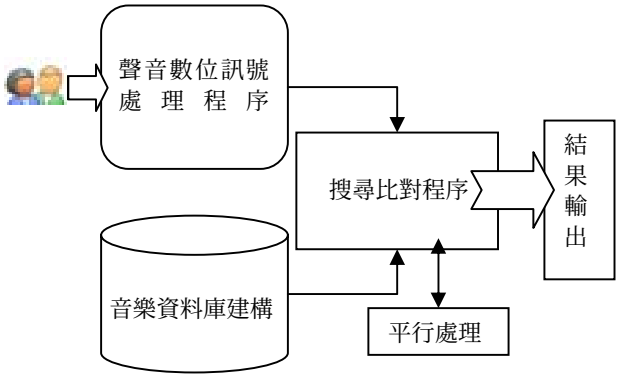
在檢索效率上，本實驗室針對音樂內涵式檢索的改良發表了相關的研究[8][9][10][11][12][13]，並且有相當不錯的成果。對於比對檢索的引擎，我們採用動態程式規劃技術[12][13]，解決了不同使用者在不同環境下容易錯誤的難題，並且針對檢索的效率有很高的改善。因此，對於數位典藏資料庫的增加，檢索系統也都能有效地在短時間內給予回應。

本系統目前已經有相當不錯的成績，經過許多使用者的測試與建議，系統不僅穩定、快速，並能有效地解決使用者在多媒體檢索上輸入的困難。『多模式音樂檢索』系統網址如下：

http://mir.cs.nthu.edu.tw/demo/miracle

2. 設計原理分析

『多模式音樂檢索』乃是國內首見的全新搜尋方式，不僅僅是一種概念上的創新，更是一種方法上的重大突破，整個系統大致可以分為四大部分：音樂資料庫建構、輸入聲音數位訊號處理程序、搜尋比對程序、平行處理。整體架構如圖一所示。



圖一 『多模式音樂檢索』架構圖

2.1 音樂資料庫建構

在說明『多模式音樂檢索』的資料庫建構時，將分成兩部分說明：

2.1.1 以 MIDI 為主的資料庫

音樂資料庫以 MIDI（Musical Instrument Digital Interface）為主，所有的歌曲都以 MIDI 的檔案格式儲存。對於非 MIDI 的其他格式，例如 wave、mp3 等，我們可以採用實驗室發表過的論文[14]，將其專換成 MIDI 格式來建構資料庫。

首先我們從所有的 MIDI 檔案中粹取出旋律和節奏這兩項資訊，並將這些資訊通通保留在一個暫存檔案之中，等到系統被啟動時，即將其轉換成是先定義好的中界格式，以方便隨後使用者搜尋時的比對。

2.1.2 以文字為主的資料庫

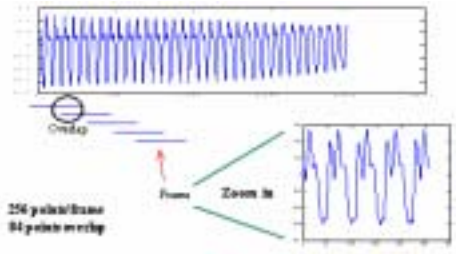
音樂資料庫以文字為主，將每一筆資料的相關資訊，例如曲名、演唱(奏)者、調性、曲目描述等相關資訊保留在一個暫存檔案之中，在系統啟動時即能與使用者輸入的資訊比對。

2.2 輸入聲音數位訊號處理程序

在輸入聲音數位訊號處理程序部分，主要用於將使用者哼、唱以及敲擊的輸入聲音經由一連串的訊號處理程序換變成和音樂資料庫相同的中界格式以方便比對，概略可以分為：聲音取樣過濾、基頻粹取（Pitch Tracking）、節奏粹取(Beat Tracking)、轉換成中介格式。

2.2.1 聲音取樣過濾

系統中我們以 11025HZ 為我們的取樣頻率，輸入的聲音以 8 位元進行量化，並將能量過低的聲音訊號視為雜訊過濾。

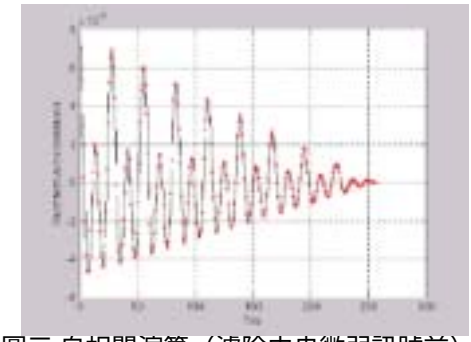


圖二 曲型的輸入歌曲聲波

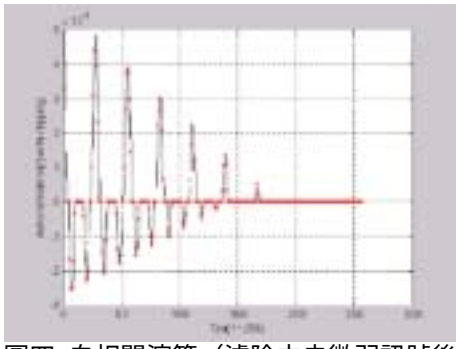
2.2.2 基頻粹取

哼唱輸入的聲波基本上我們可以把它視為一帶有固定週期的波型，我們首先將整個聲波切為數個小音框（Frame），每個音框包含 512 點，兩兩之間有 83 點重合（如圖二）。之後我們針對每個音框經過自相關演算（如圖三）[15]，並將中央原點附近的訊號濾除以求得每一個

小音框的週期（如圖四），並藉以得到每個音框的頻率。最後我們累積所有音框頻率，便可得到整個輸入聲波的基頻分佈圖（如圖五）。



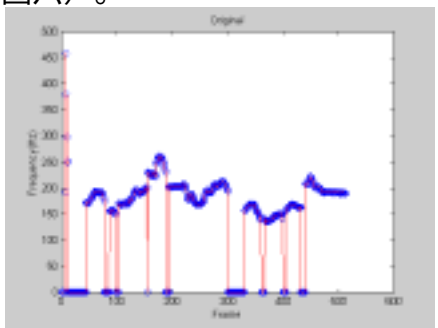
圖三 自相關演算（濾除中央微弱訊號前）



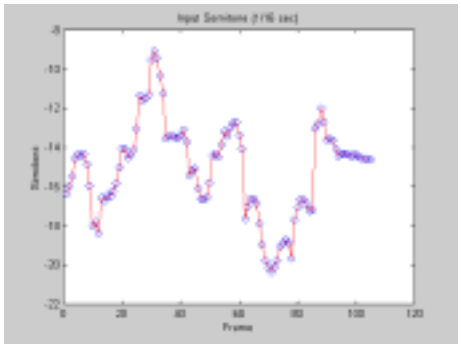
圖四 自相關演算（濾除中央微弱訊號後）

2.2.3 轉換成中介格式

在得到輸入聲波的基頻分佈後，我們可以針對一般人唱歌的頻域，將過高（大於 1043HZ）或過低（小於 82HZ）的頻率濾掉，並經過適當的平滑、刪除錯誤的訊號、降低取樣頻率後，做成時間對半音（Semitone）的作圖，轉換成和音樂資料庫中相同的中介格式，以便比對（如圖六）。



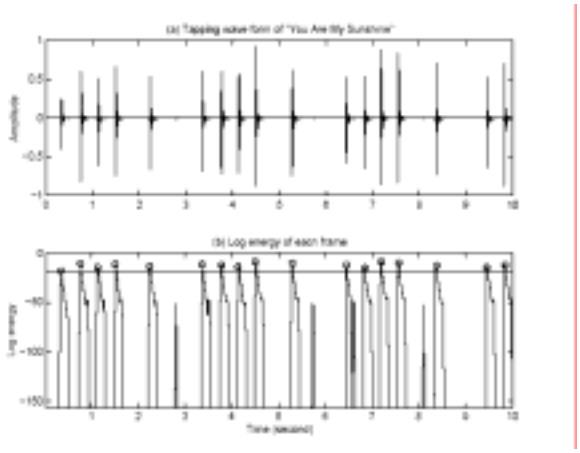
圖五 輸入聲波基頻分佈圖



圖六 中介格式

2.2.4 節奏擷取

圖七中的(a)為『多模式音樂檢索』敲擊輸入的聲波，(b)為其相對應能量取 log 後的圖。從圖(a)中可以明顯發現使用者敲擊了 16 個音，為了計算每個音的拍子長度，我們先做音框處理後，再對每一個音框算出其能量。圖中(b)的圓形標記表示能量在取 log 後的局部最大值(Local Maxima)，接著計算出每一個圓形標記間的距離，即可以算出每一個敲擊音的時間，以方便與資料庫比對。



圖七 (a)敲擊『You are my sunshine』之聲波圖
(b)相對應能量取 log 後的圖

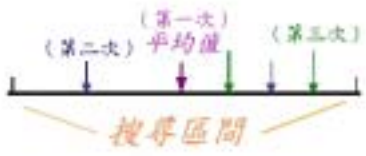
2.3 檢索比對程序

在『多模式音樂檢索』系統中，我們著重難度較高的哼唱比對、敲擊比對以及語音比對。

2.3.1 哼唱比對

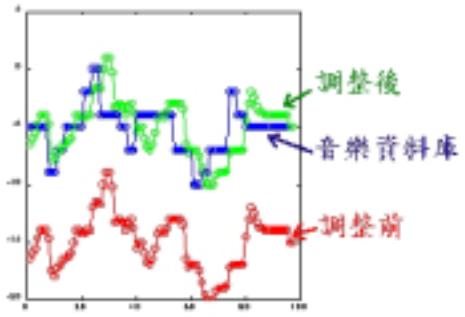
由於每位使用者的音域不同，節奏感不同，我們不可能要求每位使用者都必須如同專業的演唱者一般，哼唱出與音樂資料庫中歌曲完全相同的歌曲。因此，為了使我們的系統具有更大的彈性，在哼唱搜尋比對程序中，我們主要要克服的便是使用者基調及節奏速度不同所造成的問題。

因為每個使用者音域並不相同，在哼唱時，往往造成基調的變異。為了解決基調的變異，我們先將輸入聲波的平均值平移至和歌曲相同，並同時定義出一個搜尋區間，在這個搜尋區間之中，以類似二位元搜尋法藉以在 log(N)的時間內找到最適當的基調（如圖八）。



圖八 二位元搜尋法

圖九為一實際的調整基調範例，最下方的曲線為調整前基頻分佈，深色曲線為正確解答



圖九基調調整範例

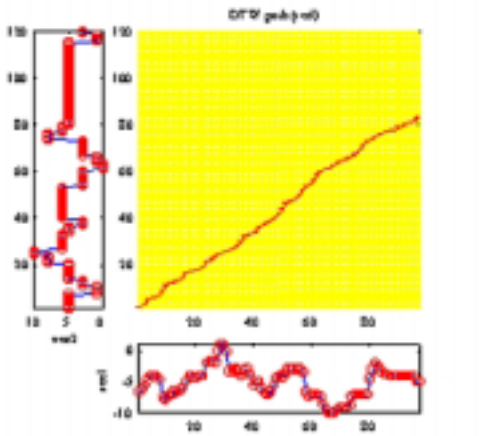
造成使用者哼唱節奏變異的原因，大致可以分為兩種。第一是基於每位使用者習慣的不同，有些人習慣以較快的速度哼唱，有些人習慣以較慢的速度哼唱。第二，由於並非人人都是專業的演唱者，不可能完全精確地記憶歌曲中的每一個音符的長度。因此，同樣的歌曲在不同的演唱者哼唱下，可能造成完全不同的節奏。

為了達到節奏變異的調整，我們採用動態時間扭曲（Dynamic Time Warping）：

$$D(i,j)=\min\begin{Bmatrix}D(i-1,j-2)\\D(i-1,j-1)\\D(i-2,j-1)\end{Bmatrix}+dist(i,j)$$

其中，*i* 表示聲音輸入中界格式的指標，*j* 表示 midi 中界格式的指標，而 *D(i,j)* 則表示兩著間的歐幾里得距離。

圖十一為一節奏變異調整範例，下方為輸入聲波，左方為音樂資料庫正確解答，圖中央曲線即為正確時間變形曲線。藉由這樣的彈性比對，我們可以找出輸入聲音與音樂資料庫歌曲間的最佳距離。



圖十一 節奏變異調整範例

2.3.2 敲擊比對

由於使用者敲擊的速度和資料庫不同，因此我們必須先對其做正規化的處理。假設輸入的為長度 m 的時間向量 t，而資料庫為長度 n 的時間向量 r，通常 n 會大於 m。在使用者不會多敲擊或是漏敲擊音的前提下，我們只需要比對 r 的

前 m 個音。但是我們必須考慮使用者通常在敲擊時是很容易多敲擊或是漏敲擊音，因此我們必須去取得不同長度的 r 來跟 t 來比對。假設取 r 的前 q 個音比對，並且將其與 t 兩者皆正規化至長度為 1000 的向量，公式如下：

$$\begin{cases} \tilde{t} = round(1000 * t / sum(t)) \\ \tilde{r} = round(1000 * r(1:q) / sum(r(1:q))) \end{cases}$$

接下來我們利用改良式的 DTW 來算出其相似性，公式如下：

$$D(i,j) = \min \begin{cases} D(i-1,j-2) + \left| r(i-1) + r(i) - t(j) \right| + \eta_1 \\ D(i-1,j-1) + \left| r(i) - r(j) \right| \\ D(i-2,j-1) + \left| r(i-1) + t(i) - r(j) \right| + \eta_2 \end{cases}$$

其中 η_1 與 η_2 為正整數。方程式中的 $\left| r(i-1) + r(i) - t(j) \right| + \eta_1$ 表示使用者漏敲音，而 $\left| r(i-1) + t(i) - r(j) \right| + \eta_2$ 表示使用者多敲擊音。

2.3.3 語音比對

在語音比對方面，我們採用隱藏式馬可夫模型(Hidden Markov Model，簡稱 HMM)，經由大量語料的訓練，得到一個語者無關（Speaker-independent）的大詞彙語音辨識引擎。在應用上，本系統可依照使用者的語音輸入來檢索歌名、歌詞、歌者等相關文字資訊，系統內部採用 Tree Lexicon 的文法結構，辨識率可以高達 97% 以上。

3.系統流程說明

本系統可以針對數位典藏資料庫的數位音樂資料（如 MIDI）將其轉換成適合比對的中介格式。當使用者進行搜尋時，只需對著麥克風自然地哼唱，系統即會根據不同輸入模式進行一連串的數位聲音處理程序，萃取出相關的音樂曲調資訊，並以其人工智慧進行模糊比對，最後依相似度高低將相關音樂資料列出。

以使用著的觀點而言，他們只需要簡單地哼唱、敲擊，或以語音及文字輸入相關資訊，便能進行在數位典藏資料庫上的音樂資料檢索或搜尋。

2.4 平行處理程序

當歌曲資料庫的資料量增大時，搜尋比對時間也將會隨著增加；有鑑於此點，我們在本系統的伺服器端加入叢集/網格運算（Cluster/Grid Computing）的功能，換句話說，當用戶端的輸入被送到伺服器時，首先有一部主要伺服器（Master Server）負責接收，然後在根據後端隨從伺服器（Slave Servers）的計算能力與當時的計算負載，進行動態的負載平衡（Load Balancing），企圖讓單位時間內的吞吐量（Throughput）為最大。

這個部分的處理程序，曾在 2002 年第一屆國家高速電腦中心的軟體比賽中榮獲全國第一名。

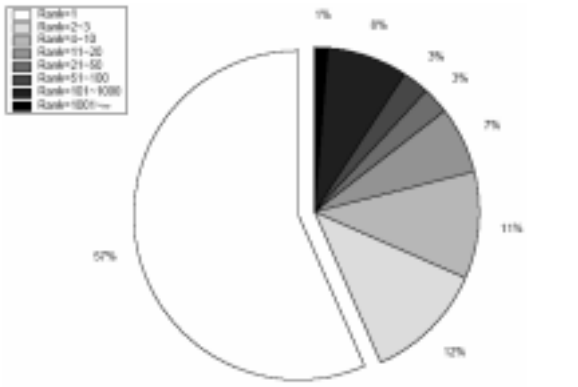
4. 實驗結果與分析

在系統中，不管使用者哼唱、敲擊的速度快或慢，哼唱、語音的音調高或低，系統都能藉著模糊比對的方式將正確的歌曲尋找出。

4.1 哼唱比對實驗結果

在我們的測試實驗中，我們請來 20 位演唱者哼唱 2000 段測試聲音，這些測試者中包含 12 位男，8 位女生，每段測試聲音八秒鐘（取樣頻率為 11025 Hz，解析度為 8 bits）。為了

測試方便，我們目前以 MIDI 當作我們的音樂資料庫指定檔案格式。



圖十二 哼唱辨識率圓餅圖

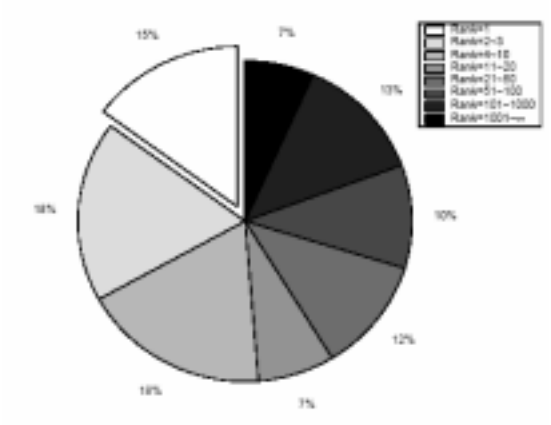
圖十二為 290 段測試歌聲辨識率圓餅圖，在系統的回傳資料中，正確歌曲落於第一名的比率約為 61%，落於前三名的比率約為 72%，落於前十名的比率則為 87%，而資料庫包含了將近 11744 首 MIDI 歌曲，足以證實以連續歌聲輸入進行音樂資料庫搜尋的可行性。

特別值得一提的是，落於 50 名後的測試樣本聲音幾乎都出自同一個人的歌聲，倘若我們將該測試者的聲音完全移除，將可得到更佳的辨識率。由此可知個人的演唱功力將會直接地影響到『多模式音樂檢索』系統的辨識率。

4.2 敲擊比對實驗結果

在這個測試實驗中，我們請來 9 位使用者(7 位男士，2 位女士)敲擊了 269 段測試聲音，每段測試聲音為 15 秒鐘，取樣頻率為 11025 Hz，解析度為 8 bits。

圖十三為 269 段測試歌聲辨識率圓餅圖，在系統的回傳資料中，正確歌曲落於第一名的比率約為 15%，落於前十名的比率約為 51%，落於前一百名的比率則為 80%。實驗數據看起來似乎不高，然而測試的資料庫包含了將近 11744 首歌曲，也就是前一百名是占了所有資料庫的 0.85%的量，足以證實以敲擊節奏輸入進行音樂資料庫搜尋的可行性。



圖十三 敲擊辨識率圓餅圖

5. 結論

在這篇報告中，我們介紹了一套『多模式音樂檢索』的音樂搜尋引擎，並經由實做完成一套連續聲音輸入的網路音樂搜尋引擎。針對包含了 11744 首中、英、台語的音樂資料庫，在我們 2000 首的測試樣本聲音中，前十名正確辨識率約為 87%。為了簡化整個比對程序，目前我們的音樂資料庫以 MIDI 檔案為主，比對哼唱也必須由歌曲的開頭為原則，不過經由測試結果發現，即使由歌曲的中央開始比對依舊能維持一定的辨識率，只不過大約會拉長辨識時間為原本的六倍。

目前系統的資料庫已經擴建至二萬多筆資料，配合叢集運算，系統回應時間亦有相當不錯的成果。相較於以前的第 1.1 節所提的相關系統，我們的系統具備了以下的優點：

1. 本系統可以直接從麥克風將使用者聲音入比對，不需要額外手動輸入歌曲音符、節奏等資訊。

2．使用者能直接以連續歌聲進行比對的音樂搜尋系統，而前人的系統在哼唱時都必須加入額外、不自然的停頓，造成使用的不方便。

3．無論使用者哼唱的節奏是快是慢、音調是高是低，本系統大部分都能進行正確的辨識。

4．使用者不需要任何的額外訓練，都可以在第一時間下直接進行哼唱、敲擊、語音的輸入與比對。

5．系統後端有強大的叢集/網格運算功能，可以在資料庫擴建增大時，依然能在有效時間內得到回應。

我們相信，在多媒體時代的來臨與多媒體資料庫的日漸豐富，配合數位典藏的資料庫，多模式的多媒體搜尋方式也會日益重要。

6. REFERENCES

[1] A. Ghias, J. Logan, D. Chamberlain, B. C. Smith, “Query by humming-musical information retrieval in an audio database”, ACM Multimedia ’95 San Francisco, 1995.

[2] Roger J. McNab, Lloyd A. Smith, Jan H. Witten, “Towards the Digital Music Library: Tune Retrieval from Acoustic Input” ACM, 1996.

[3] Roger J. McNab, Lloyd A. Smith, Jan H. Witten, “Signal Processing for Melody Transcription” *Proceedings of the 19th Australasian Computer Science Conference*, 1996.

[4] Roger J. McNab, Lloyd A. Smith, “Melody transcription for interactive applications” *Department of Computer Science University of Waikato*, New Zealand.

[5] Gold, B. and Rabiner, L. "Parallel processing techniques forestimating pitch periods of speech in the time domain," J. Acoust. Soc. Am. 46 (2), pp 442-448, 1969.

[6] Tomonari Snada and Yoichi Muraoka, “A WWW-based Melody Retrieval System - An Indexing Method for A Large Database” ICMC2000, 2000.

[7] Kosugi, N., Nishihara, Y., Sakata, T., Yamamuro, M., and Kushima, K., "A practical query-by-humming system for a large music database," In Proc. ACM Multimedia 2000, November 2000.

[8] B. Chen and and J.-S. Roger Jang, "Query by Singing", 11th IPPR Conference on Computer Vision, Graphics, and Image Processing, PP. 529-536, Taiwan, Aug 1998.

[9] I-Yang Lee, J.-S. Roger Jang, and Wen-Hao Hsu, "Content-based Music Retrieval from Acoustic Input", 12th IPPR Conference on Computer Vision, Graphics, and Image Processing, PP. 325-330, Taiwan, August 1999.

[10] J.-S. Roger Jang, Jiang-Chun Chen, Ming-Yang Kao, "MIRACLE: A Music Information Retrieval System with Clustered Computing Engines", 2nd Annual International Symposium on Music Information Retrieval 2001, Indiana University, Bloomington, Indiana, USA, October 2001.

[11] J.-S. Roger Jang and Ming-Yang Gao, "A Query-by-Singing System based on Dynamic Programming", International Workshop on Intelligent Systems Resolutions (the 8th Bellman Continuum), PP. 85-89, Hsinchu, Taiwan, Dec 2000.

[12] J.-S. Roger Jang, Hong-Ru Lee, "Hierarchical Filtering Method for Content-based Music Retrieval via Acoustic Input", The 9th ACM Multimedia Conference (Oral paper, acceptance rate 16%), PP. 401-410, Ottawa, Ontario, Canada, September 2001.

[13] J.-S. Roger Jang, Hong-Ru Lee, Ming-Yang Kao, "Content-based Music Retrieval Using Linear Scaling and Branch-and-bound Tree Search", IEEE International Conference on Multimedia and Expo, Waseda University, Tokyo, Japan, August 2001.

[14] 許嘉忻、李宏儒、王瓊雯、張智星, "由歌曲波形抽取主旋律以進行音樂檢索", Proceedings of the Seventh Conference on Artificial Intelligence and Applications (第七屆人工智慧與應用研討會), Tai-Chung, Taiwan, Nov 2002.

[15] J. Brown and B. Zhang, “Musical frequency tracking using the methods of conventional and ‘narrowed’ autocorrelation”. *Journal of the Acoustical Society of America*, Volume 89, Number 5, pages 2346-2354, 1991.