敲鍵式音樂檢索系統-觀音

A Query by Tapping Music Retrieval System-KUAN YIN

蕭樹人
國立交通大學電資專班數圖組
show.eic92g@nctu.edu.tw

柯皓仁
國立交通大學圖書館
claven@lib.nctu.edu.tw

摘要

在查詢音樂形式資料時，大部分的系統需透過與該音樂相關之文字詮釋資料進行關鍵字檢索，如歌曲名稱、作曲家、演奏者等，但是在使用者只知道音樂片段旋律的情況下，傳統檢索方式將無法直接檢索這種非文字型態的音樂資料。敲鍵式音樂檢索系統為一種內容式音樂檢索系統（Content-Based Music Retrieval, CBMR），它提供了一個全新的查詢方式，系統預先擷取每一首歌曲的關鍵旋律，並將其旋律中相鄰音符間隔之時間差存放在音樂資料庫中，藉由使用者在查詢時持續敲擊鍵盤產生的時間間隔長短不一之節奏，透過適當之近似比對技術，能得到節奏相近的音樂原件。本篇論文將展示如何建構一個敲鍵式音樂查詢系統，系統以"觀音"為名，旨在希望將人們腦海中的音樂聲音，透過很直覺的方式將其轉換成節奏並敲擊鍵盤輸入查詢，系統依相似程度排序，回應音樂之曲名、作曲者、聲音等查詢結果，讓眼睛能直接觀看到這段心裡的聲音。

在查詢音樂資料時，敲鍵式系統紀錄每一次按鍵到下一次按鍵的時間差表示音樂旋律中每個音符長短不一之節奏，不同於以輸入音高變化為主的內容式音樂查詢系統，它更適合用於一般沒有經過嚴格音感訓練的愛樂者，即使沒有準確辨別音高的能力，也能藉由每一段音樂都擁有的另一項音樂特徵即"節奏"，利用敲鍵的律動而檢索出對應之音樂資料。

關鍵字

內容式音樂查詢、敲鍵式音樂查詢、近似比對、觀音、節奏。

1. 簡介

搜尋引擎是一般人在有資訊檢索需求時第一個想到的工具，藉由使用者提供關鍵字，搜尋引擎會檢索與這些關鍵字相關的文件。不可否認地，搜尋引擎在進行關鍵字檢索的表現上，已經有令人滿意的效率及結果。然而目前多媒體數位物件技術日趨成熟，人們資訊需求不再侷限於文字型態的資料，愈來愈多的圖片、聲音、影片等多媒體文件將會成為搜尋對象，但到目前為止，多媒體文件的檢索技術仍未臻成熟。本篇論文主旨在研究如何設計一套音樂檢索系統，透過非文字型態的方式直接檢索音樂內容，讓使用者能夠很容易地找到其所想要的音樂資料。

直接利用音樂內容進行檢索的音樂檢索系統又可稱為內容式音樂檢索系統，音樂內容本身包含了大量的資訊，最常被挑出來描述音樂特徵的有旋律（Melody）和節奏（Rhythm），擷取這些音樂特徵存放在資料庫中，使用者在查詢時輸入部分的音樂特徵屬性，系統將比對資料庫中所有特徵屬性值，並回應給使用者近似的音樂資料，整個查詢過程中使用者不需輸入文字型態的詮釋資料，取而代之的是輸入音樂內容的特徵屬性。

本論文架構如下：第二節敘述內容式音樂檢索系統的相關研究工作；第三節闡述觀音系統的架構及採用的演算法；第四節說明為評估觀音系統所做的實驗，以及實驗結果的討論；第五節總結本論文並提出未來研究方向。

2. 相關研究工作

內容式音樂檢索系統在國內外均有相關研究在進行，在這些研究工作中，大致可將音樂內容檢索系統分為三個問題領域進行探討：

(1) 音樂屬性的擷取：一個音樂媒體資料是由多種不同的音樂屬性所組成，目前常見用在各種研究中的音樂屬性包含：

- 旋律（Melody）：即擷取音樂之音高屬性，爲了避免調性的差異，也有些研究是採音程（Interval）表示音樂的旋律[5][10][11][13]。
- 旋律輪廓（Melody Contour）：將歌曲之旋律依音高升、降、持續的變化情形分別以 U、D、R 三個字母表示[3][4]。
- 主旋律（Perceived Melody）：若歌曲由多個聲部所構成，則在擷取旋律屬性時，要先分析如何在多個聲部中取出單一之主旋律[7]。
- 節奏（Rhythm）：歌曲中相鄰音符的時間差[1][2]。

(2) 使用者查詢介面的設計：這方面的研究目地在提供使用者利用直覺的方式輸入查詢音樂資料，常見的查詢介面包含：

- 彈奏（Query by playing）：使用者直接透過 MIDI 樂器彈奏一段查詢旋律[12]。
- 輸入（Query by typing）：要求使用者直接在查詢介面上輸入音高簡譜字串或旋律輪廓[4][5]。
- 哼唱（Query by humming）：使用者直接透過麥克風哼唱出查詢片段[3][4][13]。
- 敲擊（Query by tapping）：在 MIDI 電子鼓或麥克風等輸入設備上敲擊出歌曲之節奏[1][2]。

(3) 近似比對（Similarity Matching）演算法：受限於人耳聽力敏感度高低、回憶音樂片段的能力、及查詢音樂時的表達能力，再加上音樂原件也有可能因為不同詮釋方式而呈現出不同的音樂特徵屬性，所以針對這種資料的特性，近似比對技術被視為基本檢索需求，常見的近似比對演算法包含：編輯距離（Edit Distance）[2][3][11] 、EMD(Earth Mover's Distance) 距離 [9] 、直接量測（Direct measure）[1] 、n-note[5][10]等。

3. 敲鍵式音樂查詢系統-觀音

3.1 系統架構

圖 1 是參考 Birmingham [14]所述的一個內容式音樂檢索系統架構，幾乎所有現行之系統都是按照這種架構在發展，觀音系統也可以用圖 1 所包含的模組一一說明，這些模組主要可分三類：

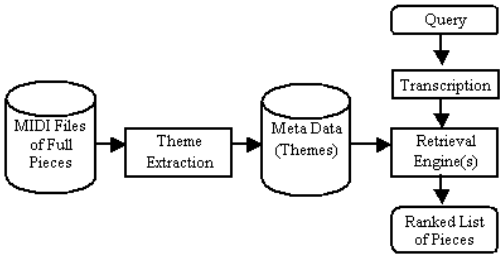

圖 1：系統架構圖

(1) 儲存模組類（圓柱狀）：

- MIDI files of Full pieces：尚未處理過的原始音樂元件，本論文中所處理的音樂資料限於 MIDI 之音樂檔案格式。

- Meta data（Themes）：每一個 MIDI 的音樂資料都須經過適當的索引程序，擷取出該音樂的主要特徵並視為索引值。本論文首先萃取用旋律表示的關鍵旋律片段，再將關鍵旋律片段轉換成節奏型態，亦即將原來用相對音高描述的旋律，轉換成用音符開始時間差描述的節奏，而這些轉換後的資料節奏即為一首音樂資料的索引。

(2) 程序模組類（四角形）：

- Theme extraction：關鍵旋律擷取之程序可分手動與自動兩種，手動方式即是先將歌曲聆聽數次，再靠聽覺感受去定位關鍵旋律片段之位置；而自動模式適用於結構形式較單純之歌曲，觀音系統利用 Suffix tree[8]分析音樂旋律輪廓屬性，以擷取關鍵旋律片段。
- Transcription：將使用者輸入之查詢節奏轉換成和 Themes 資料庫相同的節奏時間單位值，系統使用之時間計量單位為 1/1000 秒。
- Retrieval Engine：利用合適之近似比對演算法，比較輸入節奏向量與資料庫音樂片段節奏向量的差異程度。觀音系統先利用 n-gram 法將向量切割為數個顆粒（Grams），為了解決使用者可能對節奏進行之速度因主觀認知不同，導致查詢節奏相對於資料節奏全部增加或減少一個固定比例，觀音系統提出節奏商標準差（Rhythm quotient standard different）法來解決此一問題，將資料節奏和查詢節奏兩者相對應的 grams 一一相除，統計相除後的結果並計算其標準差，標準差值越小，其相似度愈高，最後用 DICE[17]函數計算兩個向量之相似度。

(3) 輸入輸出模組類（四圓角形）：

- Query：使用者只要在輸入方塊中，以印象中的節奏速度在鍵盤上敲擊，此時不用費力地彈奏出每個音符之音高，系統會自動紀錄相鄰二次按鍵被按下時的時間差，並將這些時間差值視為查詢節奏。
- Ranked list of pieces：將符合最小相似度要求之歌曲全部輸出，將查詢結果依相似度高低順序排列。由於使用者輸入的查詢節奏有可能較資料庫中的資料節奏長或短，故在比較前要取兩段節奏中較短的節奏為主，比較也只限於由第一個音開始。

3.2 屬性擷取並建立索引片段

自動索引的程序可以分為下面四個步驟進行：

(1) 擷取符合 General MIDI 規範[16]檔案的事件發生時間順序：將 MIDI 檔透過 MIDI file DisAssembler[15]程式，可以把 MIDI 格式之音樂資料，轉換為程式較易剖析之純文字格式，在該文字檔中會依時間發生順序紀錄每一個 MIDI 事件之開始與結束。

(2) 比較每個 Note On 事件之音高變化，並組合為旋律輪廓字串：在剖析 MIDI 文字檔時只會擷取每個 Note On 時音高值變化的情形，並將音高升、降、持續的變化情形分別以 U、D、R 三個字母表示。

(3) 利用詞尾樹（Suffix tree）演算法，分析歌曲旋律輪廓之重複片段：根據音樂心理學的研究認為作曲家大量運用重複片段（Frequent Patterns），普遍為創作音樂的特徵之一[6]，而這些重複片段一般均為整首曲子之精華所在，也是聽眾最容易記得的部分。利用詞尾樹（Suffix Tree）演算法[8]，可以得到字串中的重複片段，即出現次數大於 1 次，且必須包含 2 個字元以上之子字串，若該子字串也出現於包含它且較長之重複字串，則該子字串不算重複片段。

(4) 在決定出以旋律輪廓表示的關鍵旋律片段後，還需要參考該片段位於 MIDI 樂譜內的小節（Measure）

位置，依該節拍速度（Tempo）值，將音高資料轉換成以 1/1000 秒為單位的節奏數字資料，並存入 Themes 資料庫中，這些節奏數字資料表示前一個音符開始的時間，到下一個音符開始時間之時間差。

3.3 敲擊式查詢介面

現行敲擊式（Query by tapping）音樂檢索系統皆需透過麥克風[2]或 MIDI 電子鼓[1]等設備才能讓使用者輸入查詢節奏，檢索系統也多以單機執行的方式設計。爲了改善現行使用者操作介面設計不便之缺點，觀音系統設計之初即希望能讓使用者不透過其他輔助設備，直接就使用電腦鍵盤輸入節奏，而在查詢介面的設計上也採 Web based 類似 Google 搜尋引擎般的簡潔介面。如圖 2 所示為觀音目前設計之使用者介面，在查詢節奏欄位裡的數字串，是系統紀錄使用者每次按下鍵盤空白鍵的時間差值，所使用的單位和 Themes 資料庫內存放的數值單位一樣，均為 1/1000 秒，圖 2 中的查詢節奏為研究者依照小星星這段童謠在研究者心中的節奏速度所敲出來的(501, 481, 510, 511, 531, 481, 1041 ,501, 480, 501, 501, 521, 480)。

圖 2：觀音系統敲擊式查詢介面

3.4 近似比對技術

受限於不同演奏者詮釋音樂的不同方式，所以位於音樂資料庫中的音樂片段，並不一定和查詢者想像的旋律(或節奏)完全相同，再加上查詢者本身對音樂片段之記憶能力，及對音樂的表達能力，非常容易造成音高不準、多個音、少個音等現象，因此近似比對演算法設計的優劣，將決定內容式音樂檢索系統是否能正確檢索出查詢者期望之音樂片段。

在觀音系統中使用 n-gram 搭配節奏商標準差方法，來解決上面所提到的問題。先將查詢向量中的元素一個個地與對應之資料向量元素相除，可得到一組節奏商值，再計算這些節奏商值的標準差，得到的標準差可用來衡量這些商值資料和平均數之間的差異量，當兩個節奏向量間的節奏商標準差值越小，則這兩段節奏越近似。

舉例而言，使用 4-gram 的做法，將節奏向量中相鄰 4 個元素互相結合，分別切割查詢節奏向量與資料節奏向量，產生兩組 4 個數字的集合，再依序計算節奏商標準差，並設定一個顆粒離散界限值 δ，當標準差低於此門檻值才視為相同。最後再透過 DICE 係數計算兩個向量的相似程度，假設 P 與 T 分別代表查詢節奏與資料節奏的 4-gram 集合，則相似度可表示成：

$$sim(P,T) = \frac{2*|P \cap T|}{|P|+|T|}$$

觀音系統中使用的近似比對演算法採下面三個步驟：

(1) 首先比較查詢節奏 P 與資料節奏 T 之長度，假設較短之節奏長度為 minL，則 P 和 T 依 4-gram 的方式由頭至尾可切割成(minL-3)個顆粒，每個顆粒中有 4 個元素[5]。

minL = min(Length(P)，Length(T))

(2) 由頭至尾依序比較 T 與 P 每兩個顆粒之相似度，每兩個顆粒之相似度利用節奏商標準差方法決定。

- 先計算 T 和 P 對應 gram 之 4 個元素值的節奏商值 q1、q2、q3、q4：

  q1=P(1)/T(1)

  q2=P(2)/T(2)

  q3=P(3)/T(3)

  q4=P(4)/T(4)

- 再利用標準差公式，計算這 4 個值的變異離散程度。

  //計算 q1、q2 、q3 、q4 之標準差值，

  //標準差值越小則 T 和 P 越相似。

  sinStd=Stddev(q1,q2,q3,q4)

  //當節奏商越大其標準差允許在越大範圍間變化

  StdFactor=(q1+q2+q3+q4)/4

  //設定顆粒離散界限值為 δ ，

  //若標準差介於 δ *stdFactor 之內，

  // 則將此次顆粒比較視為相同。

  if sinStd < (δ *StdFactor)

  match++

- 最後再利用 DICE function 決定整個 P 與 T 的相似度。

  similarity= 2*match/ [(minL-3) + (minL-3)]

4. 實驗與討論

觀音系統提供了一個直覺的方式，讓使用者很輕易地輸入查詢片段，直接查詢音樂內容，雖然系統介面操作非常友善，但是系統之查全率（Recall rate）、查準率（Precision rate）、容錯能力，也是非常重要的議題，必須被評估與討論。

4.1 實驗資料

在進行以下評估實驗時，觀音系統收集到的 MIDI 音樂約有 120 首，挑選的歌曲都是一般人熟悉的古典音樂，歌曲樂派則包括了巴洛克、古典、浪漫、國民、現代等不同時期之曲目，由其中擷取出的關鍵節奏片段約 190 段並存放於 Themes 資料庫中，而每一段之平均長度約為 18.87 個音符。

在本評估方式中，將採童謠小星星為例，利用相同的查詢節奏向量，比較在不同索引模式的表現。P 表示所輸入的查詢節奏，T 表示小星星位於音樂資料庫的主題節奏索引片段，圖 3 表示小星星之 P vs T 圖。

P=（501,481,510,511,531,481,1041,501,480,501,501,521,480）

T=（591,592,635,585,588,638,1225,644,599,629,614,608,624）

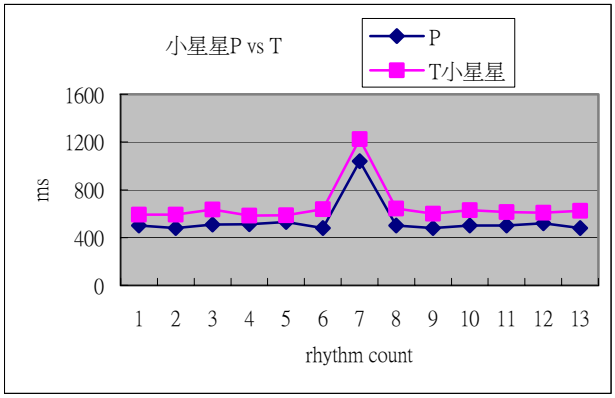

圖 3：小星星 P vs T 節奏曲線圖

4.2 顆粒離散程度評估

直接利用系統預設值（顆粒離散界限值 δ =30%）進行小星星音樂片段檢索，當輸入查詢節奏 P 後，系統回應之音樂片段如圖 4：

圖 4：系統回應(顆粒離散界線 δ=30%)

小星星雖然被找出來了且相似度為 100%，但是因為海頓的驚愕交響曲節奏也極為類似且相似度一樣達 100%，所以小星星被排到第二名，如圖 5 所示當下修顆粒離散界限至 8%左右，可以區分出驚愕與小星星到底誰比較接近查詢節奏 P。

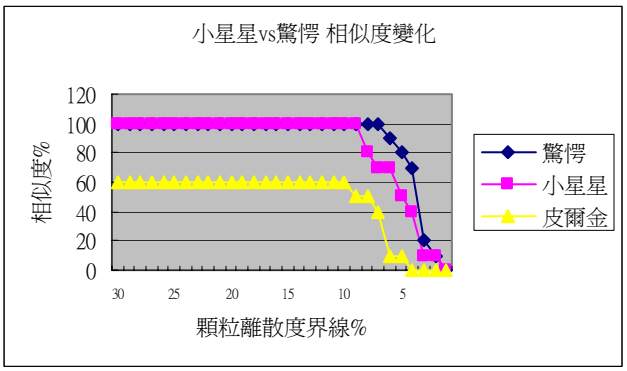

圖 5：小星星 vs 驚愕 相似度變化趨勢

在 30%~9%之間，這兩首歌曲都維持在 100%與 P 查詢節奏相似，但是到了 8%，小星星的相似度開始下降，實驗結果得知查詢片段 P 相似於驚愕更甚於小星星，這種現象會發生在當 Themes 資料庫中不同節奏片段卻擁有極其類似的節奏，故觀音可能會回應出讓使用者"驚愕"的查詢結果，這也能讓愛樂者來感受類似的節奏搭配不同旋律在不同歌曲之詮釋。

4.3 時間誤差

接下來的實驗將探討若輸入查詢節奏其中有部分音符之時間間隔不準的問題，一樣以小星星查詢節奏 P 為例，若 P 中的第七個音與第八個音之時間差由 400 變化至 1800（原始的 P 為 1041），對小星星之近似值影響如圖 6：

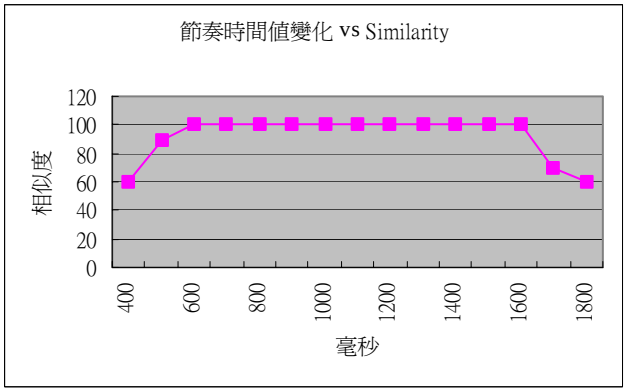

圖 6： P 中時間差變化對整體資料節奏相似度之影響

由圖 6 得知當查詢節奏 P 的第七個音與第八個音在 600~1600 毫秒之範圍區間內變化時，相似度都能維持在 100%，故若只有輸入少數錯誤的時間差音符，且能將錯誤範圍控制在正負 0.5 倍的範圍內，均能得到合理的相似度值。

4.4 多一音與少一音

接下來討論若在查詢節奏 P 多輸入一個音對近似值的影響，因為觀音系統對多輸入一個音的容錯能力須靠 n-gram 演算法來克服，故多一音在 P 裡面的位置將與近似值有很大的關係。若多輸入一個音的時間差為 200，P1~P14 分別表示 200 出現在 P 的位置（P1 表示在一開始時便多輸入一個音、P14 表示在最後才多輸入一個音，其餘類推），圖 7 表示使用 P1~P14 之查詢節奏對相似度的影響：

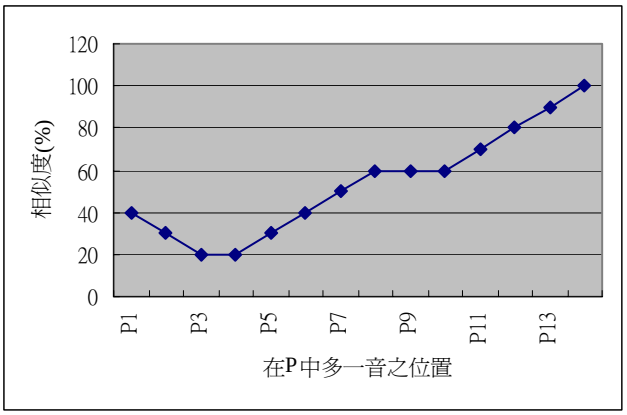

圖 7：在 P 中多一個音對整體資料節奏相似度之影響

繼續討論若在查詢節奏 P 少輸入一個音對近似值的影響，P1~P13 分別表示在 P 中少一個音的位置（P1 表示在一開始時便少輸入一個音、P13 表示在最後才少輸入一個音，其餘類推），圖 8 為使用 P1~P13 之查詢節奏對相似度的影響：

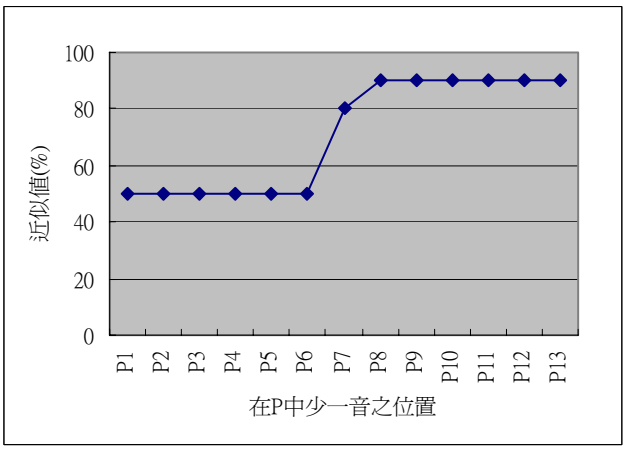

圖 8：在 P 中少一個音對整體資料節奏相似度之影響

由圖 7 與圖 8 可以觀察到，不管是多一個音或少一個音，如果發生之位置在查詢節奏 P 的前面對相似度之影響將大於在 P 的後面。

5. 結論與未來研究方向

綜上所述觀音在內容式音樂檢索系統的三個主要問題領域上，使用的解決方式為：

(1) 擷取之音樂屬性：節奏。

(2) 使用者查詢介面之設計：敲擊式。

(3) 近似比對演算法：節奏商標準差。

當使用者無法由查詢結果發現目標音樂片段時，根據實際測試與參考其他音樂檢索系統的評估報告[2]，可以歸納出幾個原因：

(1) 一般人在輸入查詢節奏時，一開始拍子及節奏都能掌握得不錯，可是越到後面越容易錯亂掉。由於輸入節奏擁有這種特性，恰好能克服 n-gram 在這裡容錯能力不佳的問題，若使用者在輸入查詢節奏時發生多一音或少一音的情形，只要發生的位置位於後段，則觀音系統還是能保持不錯的容錯能力，將查詢者要求的音樂片段標示高度的相似值。

(2) 大部分的 MIDI 檔均不只包括主旋律片段，過多的特殊演奏風格程式將不易分析主旋律，取而代之需透過人工著錄，有人工的部份就易造成人為的疏失。這將是最難克服的一個問題，也就是因為目前由網路上收集之 MIDI 古典音樂結構型式太過複雜，導致自動化工具無法自動分析主旋律並著錄，目前觀音系統中大部分的歌曲皆利用手動分析的方式判斷主旋律，如果能將收集的音樂資料定位在較單純之形式，如手機的鈴聲，相信能有更多的歌曲能透過自動化的方式建立主旋律索引檔。

(3) 即使為同一首歌曲，系統對主旋律的認定不一定和查詢者一致。問題出在當使用者查詢的節奏片段恰巧沒被系統拿來當索引值時，也會導致檢索失敗，此時可能系統需要提醒使用者進行查詢擴展（Query Expansion），用另一段節奏進行檢索。

在這篇論文中展示了一個敲鍵式音樂查詢系統—觀音，並提供直覺的輸入介面讓使用者使用，及發展出適合處理節奏比對之近似演算法；在未來還有一些研究方向可以努力，以加強觀音系統在音樂檢索上的能力：

(1) 提供精確比對功能，當使用者很有把握輸入查詢節奏的正確性時，可藉由調整顆粒離散界限值，精確比對每段節奏類似之音樂片段。

(2) 提供更高的容錯能力，若是使用者對所輸入的查詢節奏沒有把握時，可藉由高容錯能力的演算法，讓相似度不高的音樂片段也能出現在查詢結果中，如同模糊比對之功能。

(3) 設計更聰明的索引策略，增加能自動分析 MIDI 音樂主旋律之比例。

6. REFERENCES

[1] Gunnar Eisenberg, Jan-Mark Batke, Thomas Sikora: *BeatBank- An MPEG-7 compliant Query by Tapping System*: Audio Engineering Society, Convention Paper 6163, 2004.

[2] JS Roger Jang, Hong-Ru Lee, Chia-Hui Yeh: *Query by Tapping: A New Paradigm for Content-based Music Retrieval from Acoustic Input*: The Second IEEE Pacific-Rim Conference on Multimedia, Beijing, China, 2001

[3] 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.

[4] Rodger J. McNab, Lloyd A. Smith, David Bainbridge, Ian H. Witten: *The New Zealand Digital Library MELody inDex*: D-Lib Magazine, May, 1997

[5] Y.H. Tseng: “*Conetent-Based Retrieval for Music Collections*,” In Proc. Of ACM SIGIR”99, Berkley, CA, USA, Pages 176-182, 1999.

[6] V. Bakhmutova, V.D. Gusev, T.N. Titkova: *The search for adaptations in song melodies*: Computer Music Journal, vol. 21, no.1, page58~67, Spring 1997

[7] Alexandra. L. Uitdenbogerd, Justin Zobel: *Manipulation of Music for Melody Matching*: Proc. ACM International Multimedia conference, Bristol, UK, 1998

[8] Dan Gusfield: *Algorithm on String, Trees, and Sequence - Computer Science and Computational Biology*: CAMBRIDGE University Press,1997

[9] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R van Oostrum: *Using transportation distances for measuring melodic similarity*: In Prooceedings of the 4th International Conference on Music Information Retrieval. ISMIR, 2003

[10] Michael Mitzenmacher, Sean Owen : *Estimating Resemblance of MIDI Documents*: , Third International Workshop on Algorithm Engineering and Experimentation , p79–90 , 2001

[11] Alexandra. L. Uitdenbogerd, Justin Zobel: *Matching Techniques for large music databases* : In Proc. of ACM Multimedia, Pages 57-66, 1999

[12] M. Hawley, “The personal Orchestra, “Computing system, Vol. 3, No. 2, PP.289~329, 1990

[13] Arbee L.P Chen, M.Chang, J.Chen, J.L Hsu, C.H. Hsu and Spot Y.S Hua: “ *Query by music Segment: An Efficient Approach for song Retrieval* “: In Proc. Of IEEE Int”l Conf. on Multimedia and Expro, 2000

[14] William Birmingham, Bryan Pardo, Colin Meek, Jonah Shifrin, Dept. of Electrical Engineering and Computer Science, The University of Michugan *: The MusArt Music-Retrieval System D-Lib Magazine : February 2002 Volume 8 Number 2*

[15] Jeff Glatt *: MIDI file Disassembler (http://www.borg.com/~jglatt/progs/software.htm) :* 1998

[16] *http://www.midi.org*

[17] Salton, G: *Automatic text processing. The Transformation, Analysis and Retrieval of Information by Computer*: Reading, MA: Addison Wesley, 1989