

第十一屆旺宏科學獎

成果報告書

參賽編號：SA11-279

研究題目：曲風魔術師——以 MIDI 實作多元
曲風與自動改編的研究

姓名：詹易衡

關鍵字：MIDI、自動改編曲風、資料庫統計

壹、 研究動機

「曲風」顧名思義，歌曲的風格。是指音樂在整體表現上呈現出具有代表性的獨特樣貌。現今許多歌唱比賽的參賽者多會挑戰「舊歌新唱」，藉由改編原曲的曲風，帶來令人耳目一新的感受。而我們平常聽到或想起一首歌時，也會不禁好奇，如果將其以不同曲風呈現，會有什麼樣的感覺？基於這樣的理由，我們想嘗試能否藉由電腦程式自動賦予音樂不同的曲風。

貳、 研究目的

- 一、 蒐集不同曲風的音樂，分析曲風特性
- 二、 設計改編樂曲的自動化流程
- 三、 利用建立好的曲風模型及現有的音樂理論，檢驗曲風轉換的符合程度
- 四、 讓本研究的程式能藉由輸入的一段音樂，產生不同曲風的樂曲

參、 研究設備及器材

- 一、 64 位元筆記型電腦
 - 處理器：Intel(R) i5-2410 @ 2.3GHz
 - 記憶體：4.00GB
 - 作業系統：Windows 7
- 二、 麥克風
- 三、 耳機
- 四、 相關程式：
 - (一)、 編譯程式：Visual C++ 2012、Dev-C++、Matlab R2012a
 - (二)、 作曲程式：Overture、Noteworthy Composer 2
 - (三)、 音樂播放器：Windows Media Player
 - (四)、 文字存取程式：記事本
 - (五)、 二元檔讀取程式：Binary Viewer

肆、 研究過程

一、 緒論

將選定的三種曲風各二十首 MIDI 曲，依小節切割成音樂片段建立成音樂資料庫，提供輸入歌曲改變曲風，將輸入的 MIDI 歌曲切割成音樂片段後，與曲風資料庫中的片段作搜尋比對，依照和弦適當與否、旋律相似度及音符數目等比對標準，進行音符代換，重組成一首歌輸出。目前實驗了兩種代換法，分別是：

- (一) 資料庫代換法：利用旋律相似度比對，保留資料庫主旋律。
- (二) 音高保留法：利用音符個數比對，保留輸入主旋律。

另外統計三種曲風的音樂特性，進行分析。

二、文獻探討

(一)、 MIDI 簡介

MIDI 是本研究所使用的音訊格式，以下將針對其格式作簡介。

樂器數位介面（Musical Instrument Digital Interface，簡稱 **MIDI**）是一個工業標準的電子通訊協定，為電子樂器等演奏裝置（如合成器）定義各種音符或彈奏碼，容許電子樂器、電腦或其它的演奏配備彼此連接、調節和同步，得即時交換演奏資料。在電腦上 MIDI 資訊可被儲存成.mid 格式的檔案，其「記錄發聲資訊」的特性在處理上比 MP3、WAV 等「記錄波形」格式的檔案來的容易。常見的 MIDI 格式有三種：General MIDI、GS、XG。其中 General MIDI 是廣泛在各種電子裝置都支援的最基本 MIDI 格式。MIDI 發聲器材是靠 MIDI Message 控制，MIDI Message 中包含許多 MIDI Notes（MIDI 音符），一個 MIDI Note 的屬性主要有：

- **Channel**：所屬的頻道。General MIDI 中有 16 個頻道，一類樂器一個頻道，一個頻道 8 個音色，共 128 種音色。
- **Pitch**：音高。中央 C 之值為 60，每增（減）半音其值加（減）一。
- **Velocity**：強度。其值範圍從 0 到 127。

另外 MIDI Controller（控制項）能改變 MIDI Note 的發音性質。常用的 Controller 有：Controller 1（顫音效果）、Controller 64（延音效果）……等。

(二)、 Midi 格式解析

Header Chunk

Chunk ID	Chunk size	Format type	tracknum	timedivision
MThd	00 00 00 06	0~2	1~65535	Ex:480

此為每首 MIDI 音樂的基本標頭，占了 6 bytes 的大小。有三種 Format type，0 代表單音軌，1 為多音軌，2 為多音軌且節奏獨立。tracknum 代表音軌個數。timedivision 為所佔時間單位數。

Track：

Chunk	Chunk size	Track event data
MTrk	Variable length	See Events

此為每個音軌個別的資訊。

Events：所有音符的事件，包括

Delta-time	Event	Channel	Note	velocity
Ex:81 00	Ex:0x9(Note on)	0~15	0~127	0~127

Delta-time:間隔時間。有固定格式，如範例中的 81 00(Hex)，轉為 2 進制，為 10000001 00000000(共 2Bytes)，每個 Bytes 的第一位代表後面是否還有 Byte，若為 1 則代表後面還有 Byte，若為 0 則代表這是最後一個 Byte，故實際時間為 0000001 0000000(Bin)=128(Dec) (Beats)

Event 與 **Channel** 合為 1Byte(代表在該聲道發生的事件)例如 event 9 為 **note_on** 後面會各有 2Byte，分別代表 **Note**、**velocity**。

在各個 **Track** 結束時 會用 4 個 **Byte (00 FF 2F 00)**代表結束

依照 **MIDI** 定義的格式，撰寫程式進行 **MIDI** 音樂的輸入和輸出。

(三)、 基本樂理簡介

一首歌曲包含旋律、節奏及和弦，並含有音名、調號、拍子等元素。

1. 音樂三要素：節奏、曲調、和聲。

(1) **節奏**：抑揚頓挫，由音之長短、強弱、快慢，組合而成。

(2) **曲調**：指音的連續進行，由音的高低組合而成。曲調沒有和聲，仍然可以構成音樂，但必須有節奏。

(3) **和聲**：兩個以上高低不同的音，同時或先後連續出現所產生的混合音。

2. 調號 (Key Signature)

意指音階具一定的高度時，第一音（主音）的位置，固定在某一音名上。例如，C大調就是以 C (**Do**)音當作主音的大音階調子。

3. 大小調(Major, Minor)：根據其音階的組成不同，調號大致分為大調和小調。

(四)、 曲風介紹

現在被歸類的曲風種類繁多，但卻僅只有抽象的定義，因此本研究對於曲風的歸類來自各 **MIDI** 網站對曲風做的分類。而目前有流行鋼琴樂(**slow soul**)、鄉村樂(**country**)、龐克搖滾(**punk**)，每種曲風都有其節奏、旋律及和弦的特性，造成聆聽時不同的感受。以下將針對此三種曲風作簡介。

1. 流行鋼琴樂

為最基礎的四拍節奏，藉由清晰的和弦達到簡單卻不失豐富的伴奏效果，許多流行樂的鋼琴改編曲皆採用此種模式。因其主旋律與和弦界限分明、容易偵測的特性，此曲風也是本研究的基本曲風，所有資料統計、程式處理流程，都以之為預設資料庫，進行初步測試。



圖(一) 流行鋼琴樂範例。其主旋律與和弦分別固定在第一、第二音軌，且和弦相當容易偵測。

2. 鄉村樂

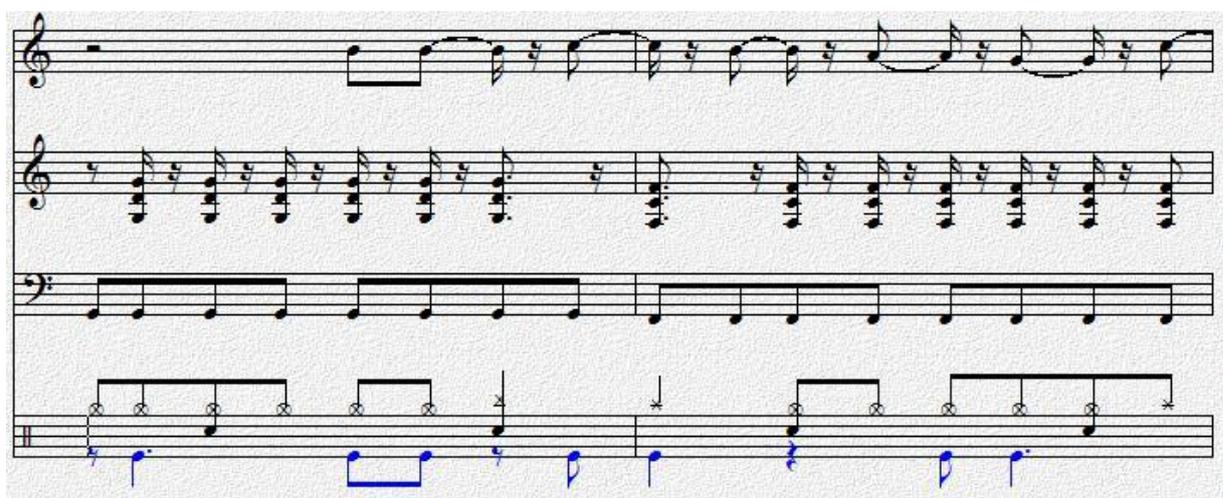
1920 年代發源於美國南部的鄉村音樂，目前已發展出許多不同的分支，各有其特色。為求資料的一致性，本研究蒐集的鄉村樂，是以最傳統的形式為主。主要樂器有班卓琴、小提琴以及用來做為主要伴奏的吉他和貝斯。其主旋律音符密集、和弦簡單且時常循環出現。



圖(二) 鄉村樂範例。第一音軌為主旋律，第二、三音軌為伴奏用的吉他、貝斯。

3. 龐克搖滾

龐克搖滾為搖滾樂的一個分支，於 1974 到 1976 年間，在美國、英國及澳洲等國家萌芽。歌曲短暫、節奏快速且強烈、旋律簡單好記，歌詞時常處理到政治、反社會等議題。



圖(三) 龐克搖滾樂範例。第一音軌為主旋律，第二、三音軌為吉他、貝斯，第四音軌為鼓組。

(五)、 最長共同子序列 (Longest Common Subsequence)

此為本研究比對主旋律相似度時用到的概念。所謂的「序列」指的是一串相同型態的資料，例如一串數字「1, 4, 3, 10, 8」；而依照由左到右的順序，取幾個數字出來，如「1, 3, 8」，即稱為「子序列」。「最長共同子序列」即是兩段序列相同的子序列中，長度最長的那一個。找出兩個序列

的 LCS，通常採用「動態規畫(Dynamic Programming)」這個較有效率的演算法。

(六)、 WAV (Waveform audio format)

為一種未經過壓縮的音訊格式，記錄著振幅大小與時間的關係。

1. 取樣頻率(Sample ate)

每秒鐘紀錄的資料點數，點數愈高，品質愈好。常用的取樣頻率有：
8000hz(電話音質)、16000hz(一般語音辨識採用)、44100hz(CD 音質)

2. 取樣解析度(Bit Resolution)

每個聲音資料點所用的位元數，常用的數值有：8bit、16bit。

(七)、 音高追蹤

1. 音高追蹤基本流程

- (1) 將整段音訊訊號(大多為.wav)切成音框(Frames)，鄰近音框可重疊
Frame size: 每個音框的大小，必須至少能夠包含兩個以上的基本週期，才能抓取語音的特性。而人聲頻率範圍大約界在 50~1000 hz 之間，音框大小必須大於 $\text{sample rate}/50$ 個點

Frame rate: 每秒鐘取樣的音框數量

- (2) 算出每個音框對應的音高
- (3) 排除不穩定音高
- (4) 對整段音高進行平滑化

2. 幾個常見算出音框對應音高的方法

(1) 時域：

- A. ACF: Auto correlation function
- B. AMDF: Average magnitude difference function

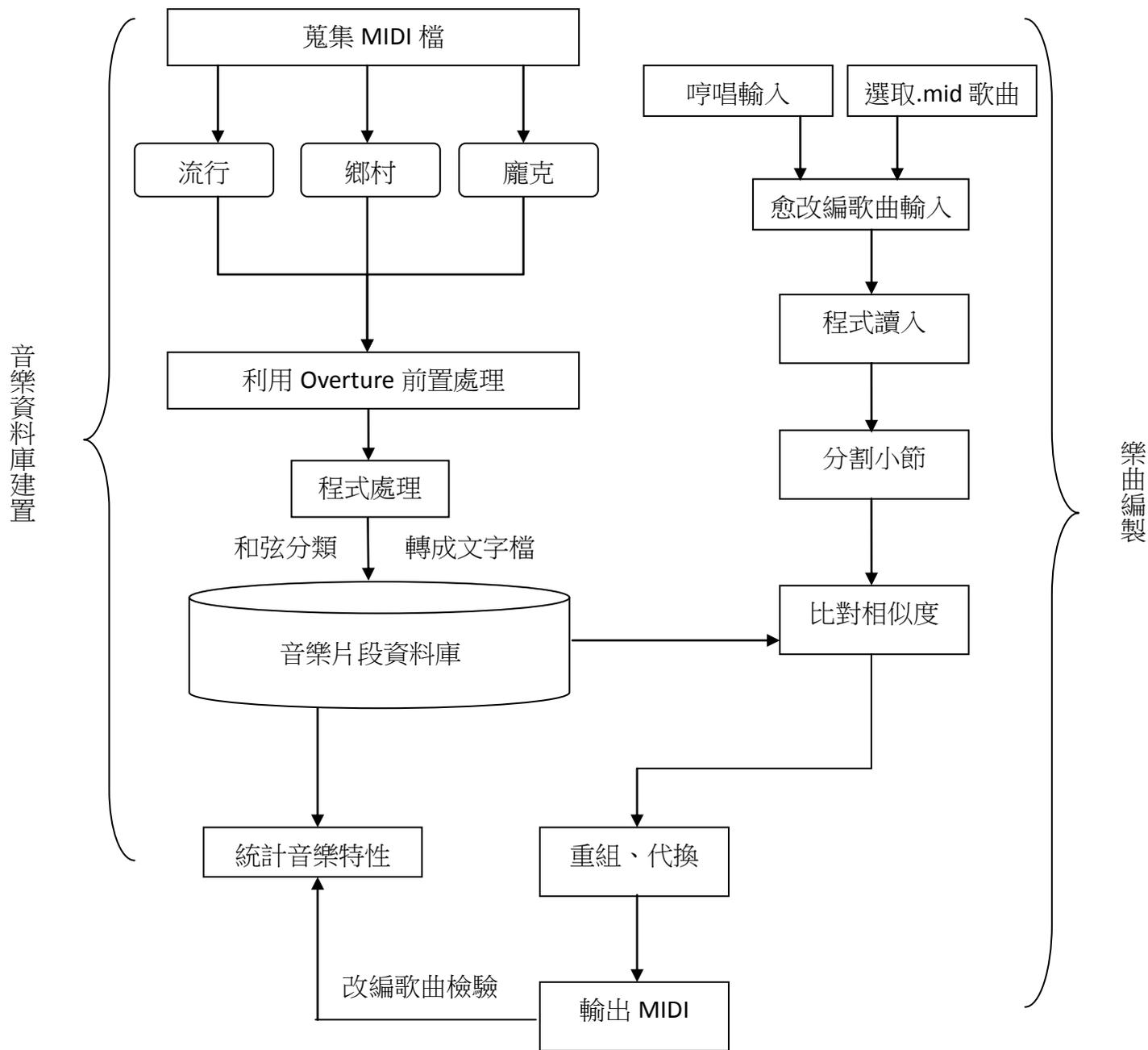
(2) 頻域

- A. Cepstrum method
- B. Harmonic product spectrum method

本研究使用的方法為 AMDF，詳細過程會在研究過程中解釋。

三、研究過程

(一)、研究流程



圖(四)研究流程圖

(二)、資料庫建立流程



圖(五)資料庫建立流程圖

1. 下載資料庫所需檔案

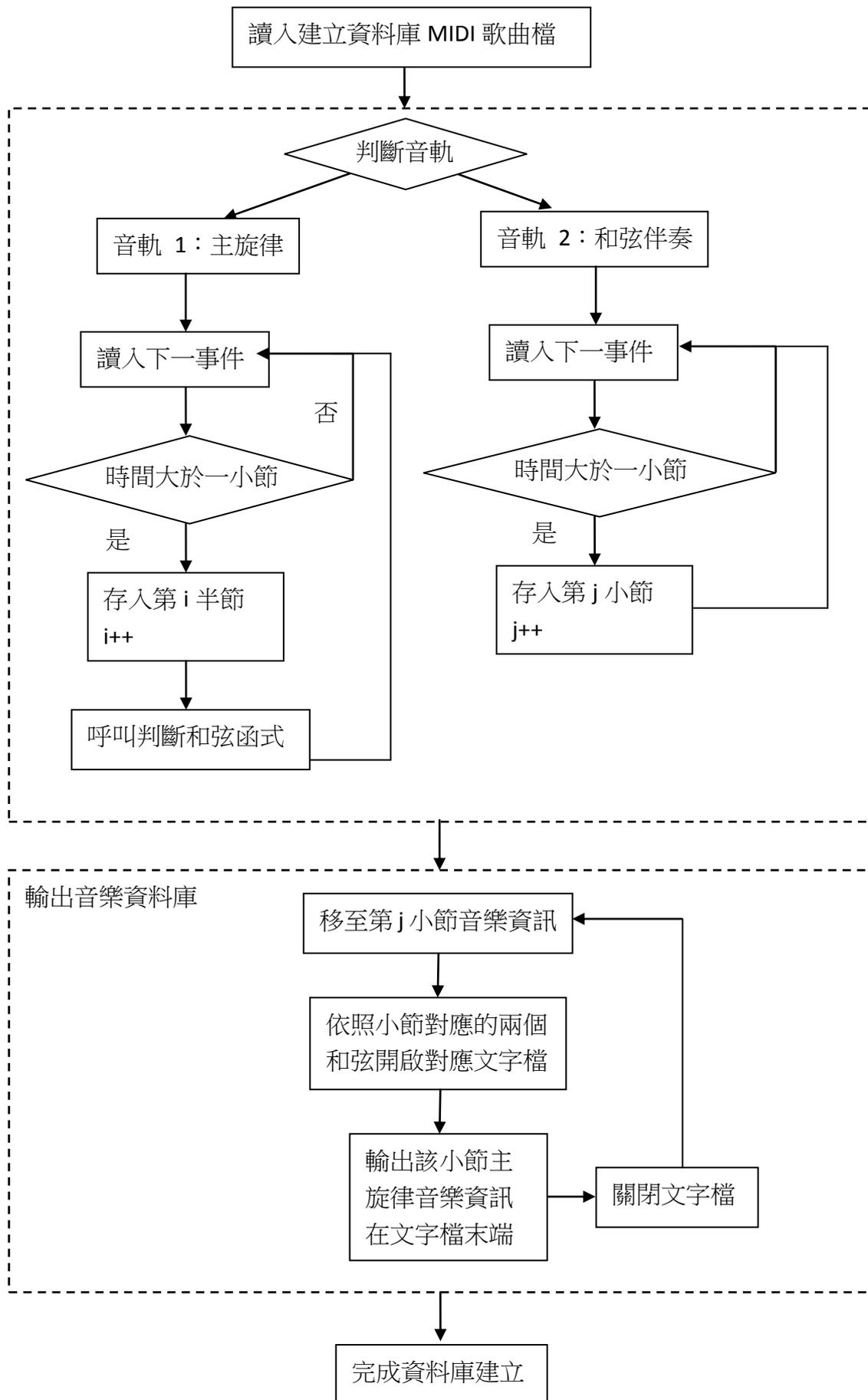
首先下載已經製成琴譜的.ove 檔，處理後再轉成 midi 格式；或到提供免費 midi 音樂的網站下載 midi 檔，利用 Overture 程式轉成琴譜處理後，再轉回 midi 檔。本研究目前針對 4/4 拍的音樂，下載時亦須注意拍號。

2. 前置處理後轉成 MIDI

一首歌可用不同調子（如 C 大調、F 大調等）表示，但只有音高上的不同，聽起來大致的感覺是一樣的。但統計時，如果不考慮調性，原本屬於同一個性質的音便會被當作不同的音看待，所以必須先將所有歌曲移成同調。利用 Overture 作曲程式，先將所有琴譜的調性移至 C 大調或 A 小調以統一調號，再把缺少主旋律或缺少伴奏的小節刪除以便於事後程式處理。完成後琴譜匯出成 MIDI 檔。

3. MIDI 音訊處理

為了將各小節的音樂資訊建立成資料庫，利用 Visual C++ 撰寫出的程式讀入 MIDI 檔，去除疊音，保留同時刻最高的音符後，將取得的音訊資料轉換成數字。



(圖六)音樂片段資料庫統計程式流程圖

4. 曲風特性統計

為了以數據化的方式呈現出各曲風的差異性，以及便於事後作檢驗的用途。統計的項目如下表：

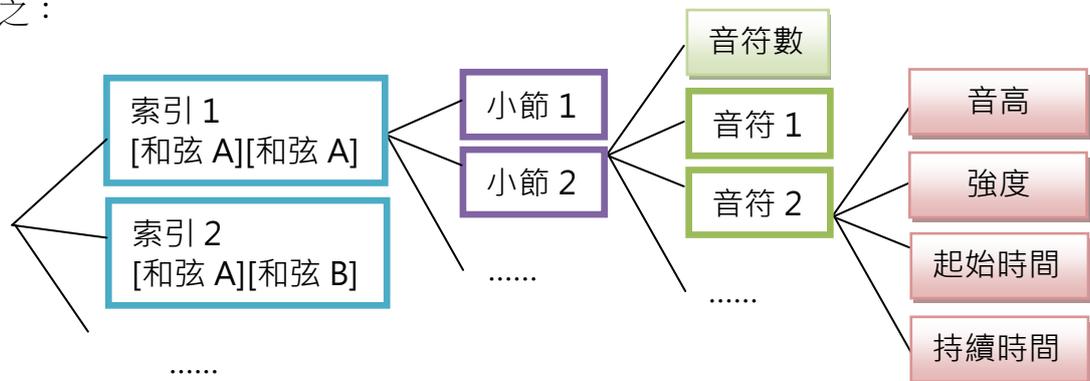
統計項目	求法	用途
每分鐘音符數	「每小節平均音符數」乘以「每分鐘節拍數」除以四	分析各曲風的旋律密集程度
連續音高差距	「所有兩個相鄰音的音高差值總和」除以「差值個數」	分析音和音之間的起伏程度
音高標準差	「所有音減去平均音高」的平方和再開根號	分析歌曲的音域集中程度
小節音符數目	統計各曲風各音符數目/小節的片段數目和分布	觀察各曲風每小節的音符數目分布
音階分布	求出每首歌音階分布的比例，加總後再求出該曲風的分布比例	觀察各曲風常出現的音或不常出現的音
和弦分布	求出每首歌和弦分布的比例，加總後再求出該曲風的和弦分布比例	觀察各曲風常出現的和弦或不常出現的和弦
和弦轉移機率	計算所列的七個和弦下一小節轉移到七個和弦分別的機率，得出一七階狀態	觀察各曲風是否有常見的和弦走法

表(一)曲風統計特性

統計完成後，下一個步驟，便是將資料庫轉存成後續的編曲程式容易讀取的形式。

5. 依和弦分類

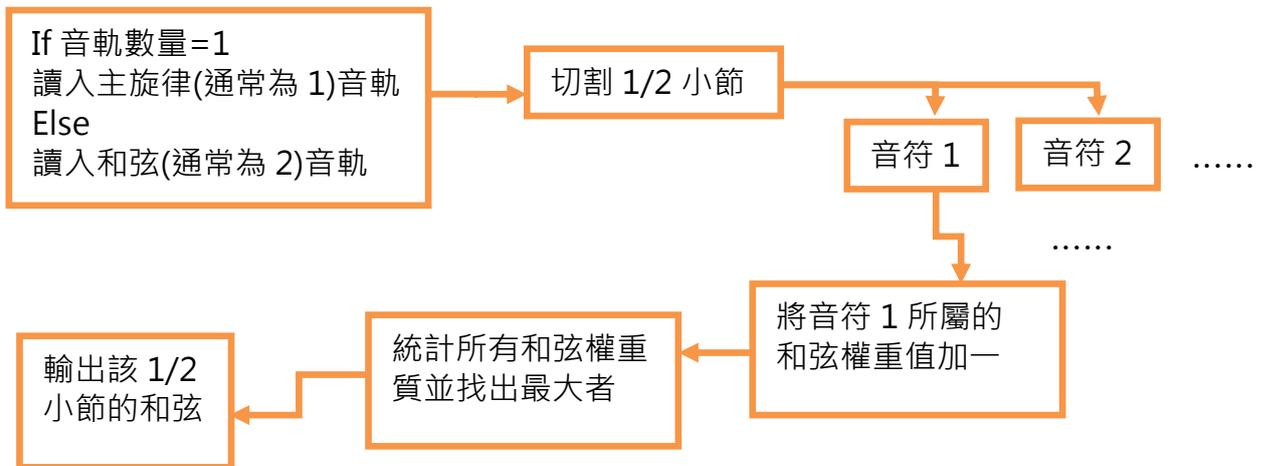
在兩種比對方法中，為了使資料比對的效率提升，建立樂句資料庫時依照**和弦**建立檔名索引，由於下載琴譜時已把所有歌都轉成**C**大調，此處只要存隸屬於**C**大調的所有常見和弦即可；索引中包含隸屬於該索引的小節；記錄小節的資訊有該小節的**音符數目**和**音符資訊**；音符資訊包含每個音符的**音高**、**起始時間**、**持續時間**及**力度**。總共五層的結構，以下圖簡化之：



圖(七)資料庫架構圖

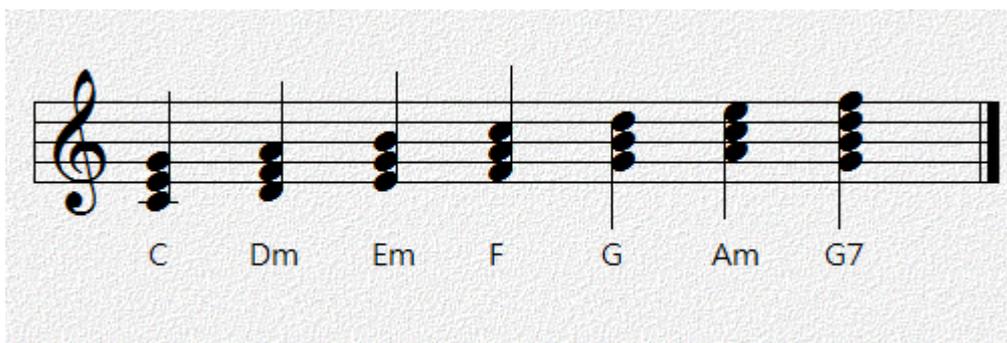
將每首歌的伴奏（即和弦）部分固定在第二音軌，在偵測和弦時，只需處理第二音軌即可。由於本研究蒐集的資料庫每小節大多出現兩個和弦，因此將每小

節對分成兩段，每段分別以 C 大調的和弦表做比對，讀入的音若為和弦的組成音之一，則該和弦的權重加一，依此類推，最後權重值最大者即為該段之和弦；如果有兩個以上的和弦權重相等，則取較先出現的和弦。



圖(八)和弦判斷流程圖

手動列出之 C 大調常見和弦表如下圖：



圖(九)C 大調常見和弦

範例：



圖(十)和弦範例

統計結果：前兩拍 A 有 2 個、E 有 1 個，判斷結果為 ACE (Am)和弦；後兩拍 E 有 1 個、B 有 1 個、G 有 1 個，判斷結果為 EGB(Em)和弦。

資料庫實際的分類方式如下圖，每個文字檔名稱代表小節中兩個和弦轉換成數字的代號，0 為 C，1 為 D，以此類推；文字檔中包含該小節的資訊。

名稱	修改日期	類型	大小
0-0	2012/5/2 下午 08...	文字文件	141 KB
0-1	2012/5/2 下午 08...	文字文件	13 KB
0-2	2012/5/2 下午 08...	文字文件	6 KB
0-3	2012/5/2 下午 08...	文字文件	17 KB
0-4	2012/5/2 下午 08...	文字文件	18 KB
0-5	2012/5/2 下午 08...	文字文件	1 KB
0-6	2012/5/2 下午 08...	文字文件	5 KB

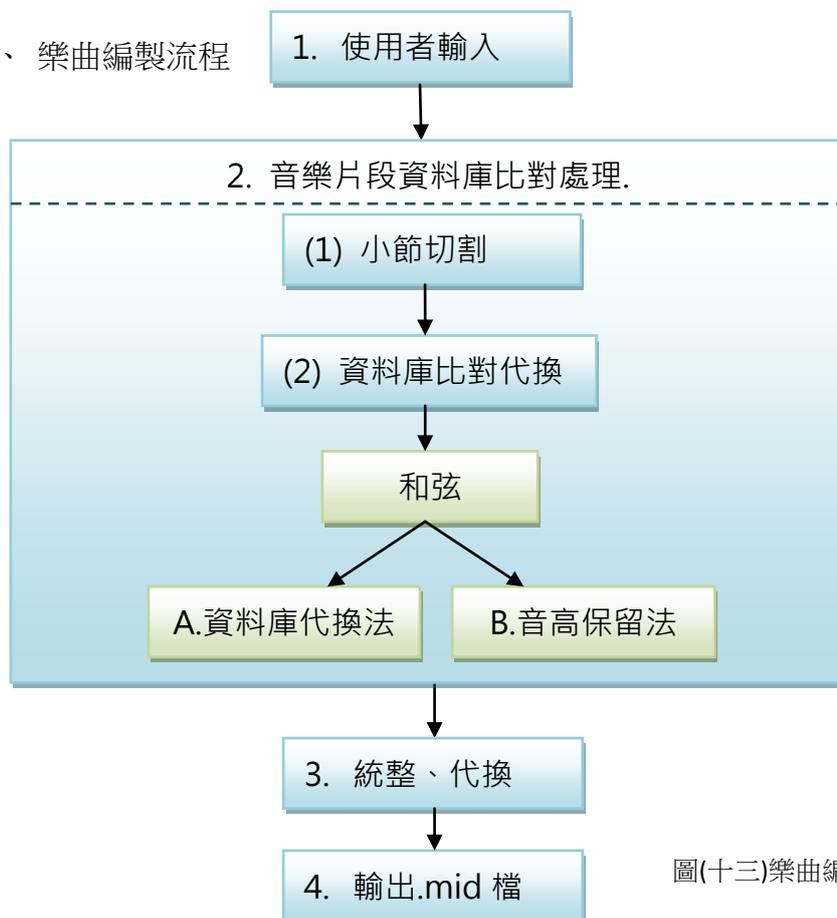
圖(十一)和弦分類文字

檔



圖(十二)小節資訊記錄

(三)、樂曲編製流程



圖(十三)樂曲編製流程圖

1. 使用者輸入：

將一段使用者提供包含主旋律、和弦的 MIDI 格式音樂用程式讀入，使用者可選擇曲風、快慢和調性。

2. 資料庫比對代換

(1) 小節切割

將輸入的音樂，分別以一小節為單位切割後，再分別和資料庫中相同長度的片段一一比對。

(2) 資料庫比對

A. 資料庫代換法

從相對應和弦分類中挑選主旋律相似度最高的小節。相似度判斷法：令 L = 兩段主旋律的最長共同子序列長度， K = 較長那段的長度，則取其相似值為 L/K 。輸出時保留資料庫主旋律。

B. 音高保留法

從相對應和弦分類中挑選主旋律音符個數相同的小節，將輸入的主旋律音高代換進所選中的小節中。因此輸出時原本輸入的主旋律音高會被保留，但節奏改變。

為了避免相同的旋律因為差了八度而被判定為不同的音，開始比對前，先將資料中每小節的音高限制在同一範圍內，作法為取該小節中最低的音，除以 12 後的商數 Q (Q 即為該音隸屬的八度)，再將小節內每個音的音高減去 $Q*12$ ，如此一來即可將最低音音高都限定在 $0\sim11$ 的八度範圍，比對時只需將輸入旋律之音高平移至相同範圍即可。

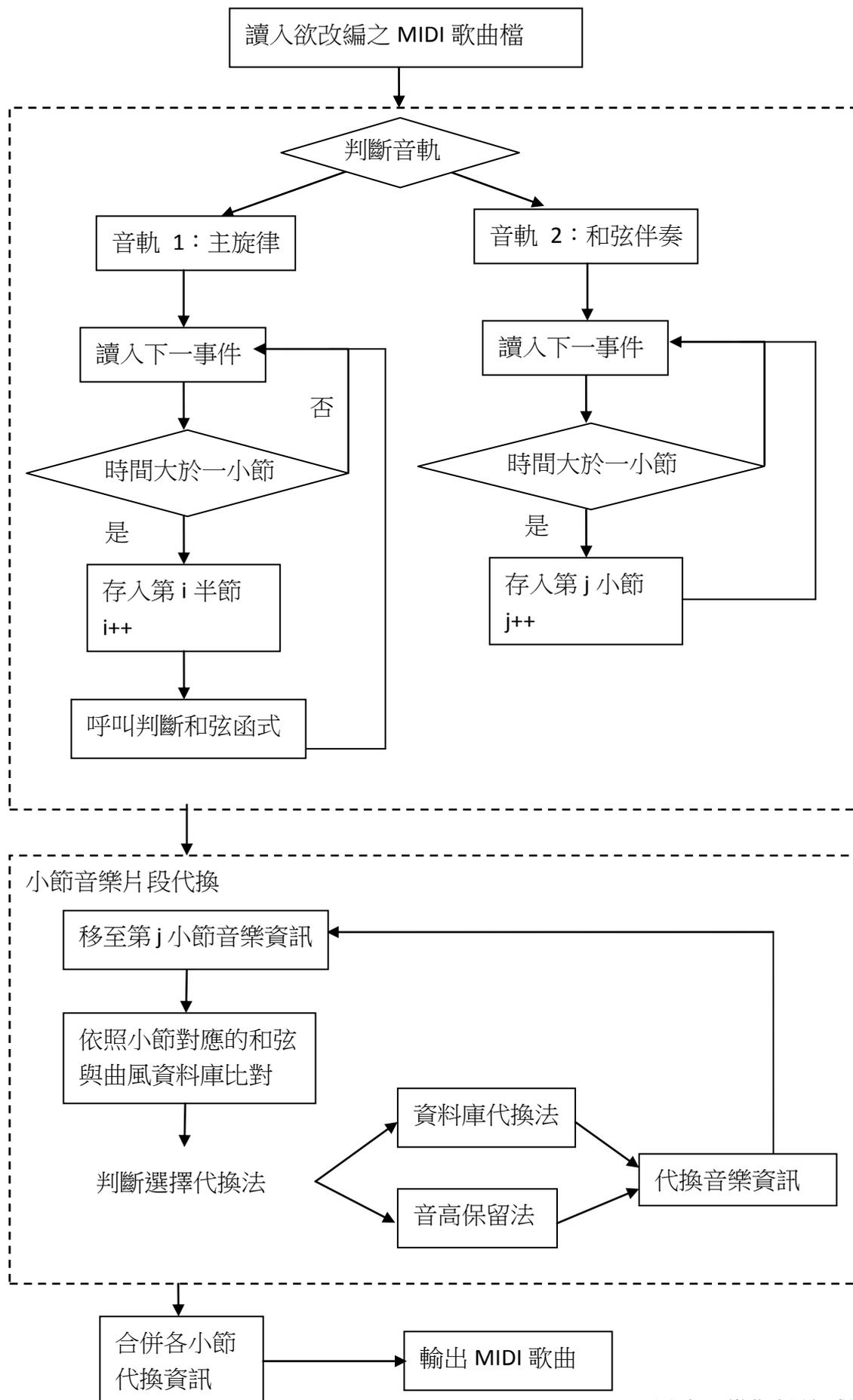
在各音樂片段的儲存中，必須判斷是否相同而加以分類，但若在代換旋律時也必須逐步比對，是非常花時間的，因此需要快速的找到音樂片段在資料庫中的位置。我們先依照各音樂片段所屬的和弦作為索引，從大量資料中過濾出符合條件的音樂片段，再以兩種代換法作代換。

3. 統整、代換

比對完成後，統整資料庫中挑選出來的小節。主旋律處理完成後，將和弦音軌，以及其他伴奏音軌合併成一文字檔。

4. 輸出 MIDI

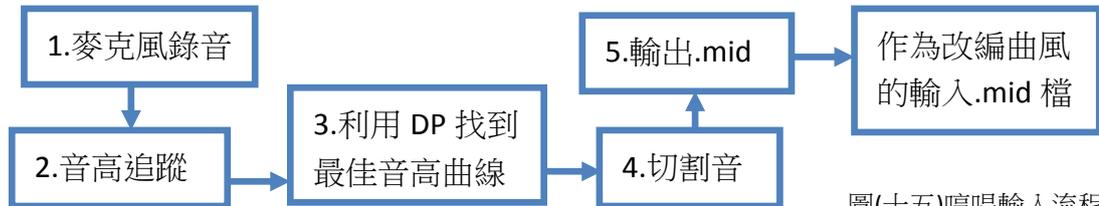
將文字資訊轉換成 MIDI 格式輸出。



圖(十四)樂曲改編程式流程圖

(四)、使用哼唱輸入

以.mid 檔案作為改編曲風的輸入音樂檔對於一般大眾使用者並不是那樣的方便，因為並不是每個人皆會用打譜軟體編寫歌曲，而網路上大多多的 MIDI 歌曲都含有許多音軌，編曲的方式也並非統一，因此隨意選取一首輸入可能造成程式判斷不正確而無法達到想要的效果，或者發生無法辨識的錯誤。使用哼唱的方式輸入曲子，將能方便使用者做改編曲風的工作。



圖(十五)哼唱輸入流程圖

1. 麥克風錄音

利用筆記型電腦內建的麥克風或者購買額外的麥克風皆可以做.wav 檔案的聲音錄製。錄音的來源可以是人聲歌唱或者樂器演奏。將錄到的音波資訊寫成.wav 檔，再用 matlab 匯出的 dll 函式對剛輸出的 wav 檔作音高追蹤。

2. 音高追蹤

(1) 切割音框

在本研究中音框取樣的參數有：音框大小 640 以及音框率 100 代表會每個音框會有重疊的部分

(2) 算出各音框對應之音高

AMDF(average magnitude difference function)

$$\text{amdf}(\tau) = \sum_{i=0}^{n-1-\tau} \text{abs}(s(i) - s(i + \tau))$$

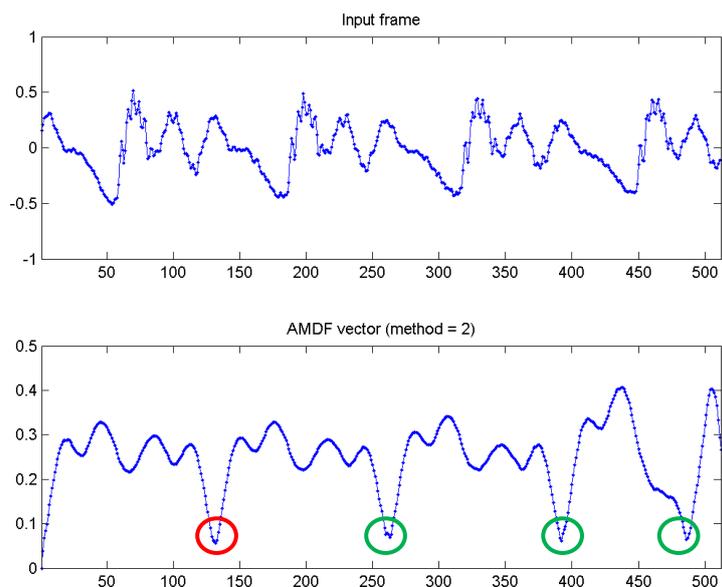
AMDF 的做法：取一個音框 A 與同樣資料的音框 A'，每次皆將 A' 向右平移一單位後將兩音框函數相減後取絕對值，因此每次都會減少頭和尾兩次運算。算出所有的 AMDF 值後便能得到一個圖形，圖形中一個週期皆會出現許多區域低點(local minimum)(如下圖綠色圈處)，該音框代表的頻率則需找到圖形中第一個區域低點的索引值(如下圖紅色圈處)，得到的頻率為 $\text{samplerate}/(\text{index of the second local min} - 0)$

以 AMDF 作為音高追蹤的優點有：

時域的音高追蹤較容易，可以省去從時域轉到頻域再做運算的時間

AMDF 只涉及加減法，運算數值較 ACF 小也較快

以單音輸入的.wav 檔來說能夠有很高的精準度。



圖(十六)波型與 AMDF 向量對照圖

(3) 利用 DP 找到最佳的整段音高

用 AMDF 找 pitch 常有個問題，便是誤判第一個區域低點而將頻率判斷成兩倍或一半，也就是差一個上下八度音，因此要在 AMDF 矩陣中找到一條路徑為 amdf 函數值和最小並且音高之間差值要小，因此定義下列公式

$$\text{cost}(\mathbf{p}, \theta, m) = \sum_{i=1}^n \text{amdf}_i(p_i) + \theta \times \sum_{i=1}^{n-1} |p_i - p_{i+1}|^m$$

θ 為調整整段音高平滑度的參數， θ 愈高代表愈平滑， θ 若為零便代表不考慮平滑度直接採用 amdf 函數的最小值。利用 DP 找出 cost 函數中最小者便為最佳解。

(4) 旋律追蹤

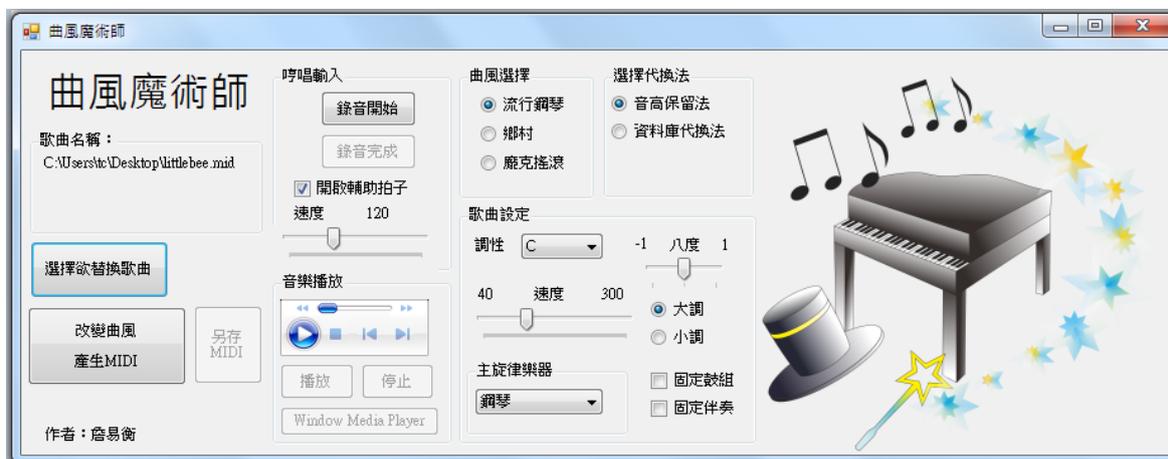
得到的音高向量還是時間對應的連續音高值，必須音高轉換成音符的音高、起始時間和結束時間才能對應到 MIDI 的格式作輸出。

(5) 輸出 MIDI

將旋律追蹤的結果寫成一個單音軌的 .mid 檔，作為改編曲風的輸入歌曲，直接接上原本歌曲改編流程。

伍、 結論

- 一、統計三種曲風：流行鋼琴、鄉村、龐克搖滾共 60 首歌的音樂資訊，並分析不同曲風的音樂特性和數據之間的相關性。
- 二、實做出將輸入的 midi 音樂依照資料庫代換各小節以改變曲風的程式，程式介面如下圖：



圖(十七)程式介面圖

使用方法：

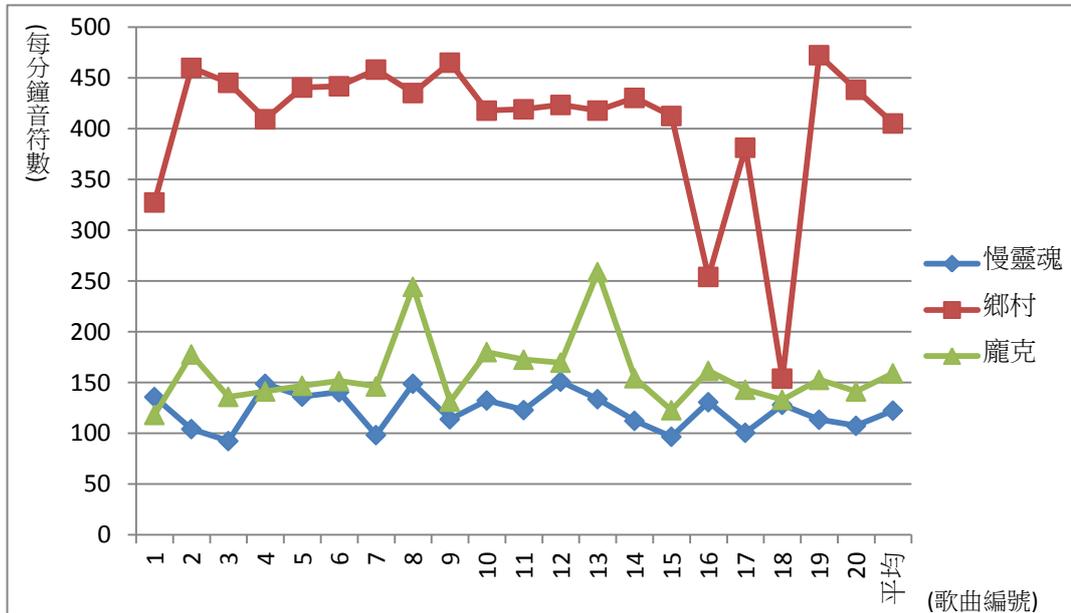
- (一)、 從資料夾中選取欲替換的 midi 音樂檔或者直接拖曳至視窗內
- (二)、 點選「播放」鍵可試聽原本歌曲
- (三)、 選取曲風
- (四)、 調整調性
- (五)、 調整音高八度
- (六)、 調整速度
- (七)、 選擇代換法
- (八)、 點選「改變曲風，產生 midi」鍵
- (九)、 開始播放
- (十)、 可點選「另存」鍵將產生的歌曲存至使用者自訂的位置
- (十一)、 點選「Windows Media Player」鍵，以 Windows Media Player 開啟 midi 檔

陸、 討論及應用

一、 曲風統計結果探討

三種曲風各挑 20 首進行統計。流行鋼琴、鄉村、龐克分別有 1353、1423、839 小節。

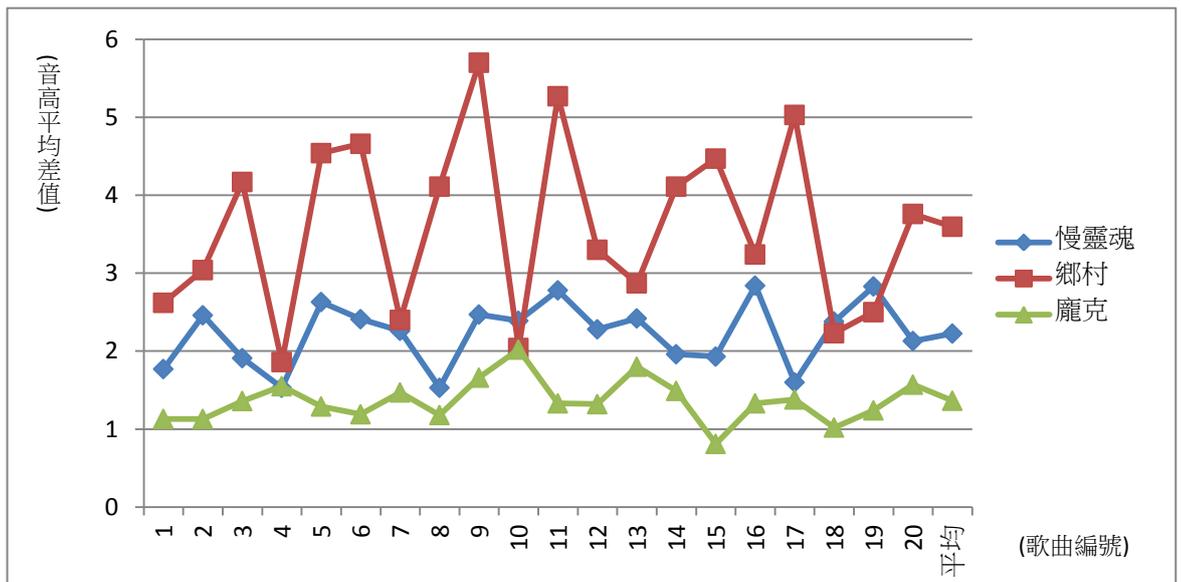
(一)、 每分鐘音符數



圖(十八)單位時間音符數

由於本研究所蒐集的鄉村樂大都節奏輕快，音符顆粒密集；而龐克搖滾節奏感強，有一定的速度；流行鋼琴則偏抒情，多為慢歌。因此，鄉村的旋律最密集，龐克次之，流行鋼琴最稀疏。

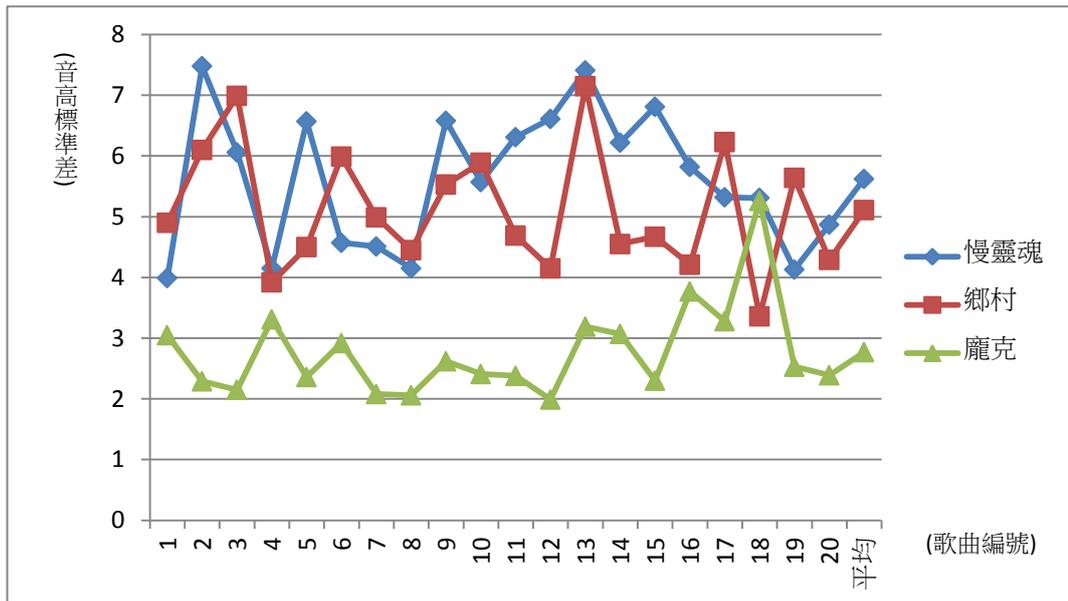
(二)、 連續音高差距



圖(十九)連續音高差距

大體上，鄉村樂的連續音高差距較大，流行鋼琴、龐克則較小。推測原因：鄉村樂相較其餘兩個曲風較少出現連續的重複音。

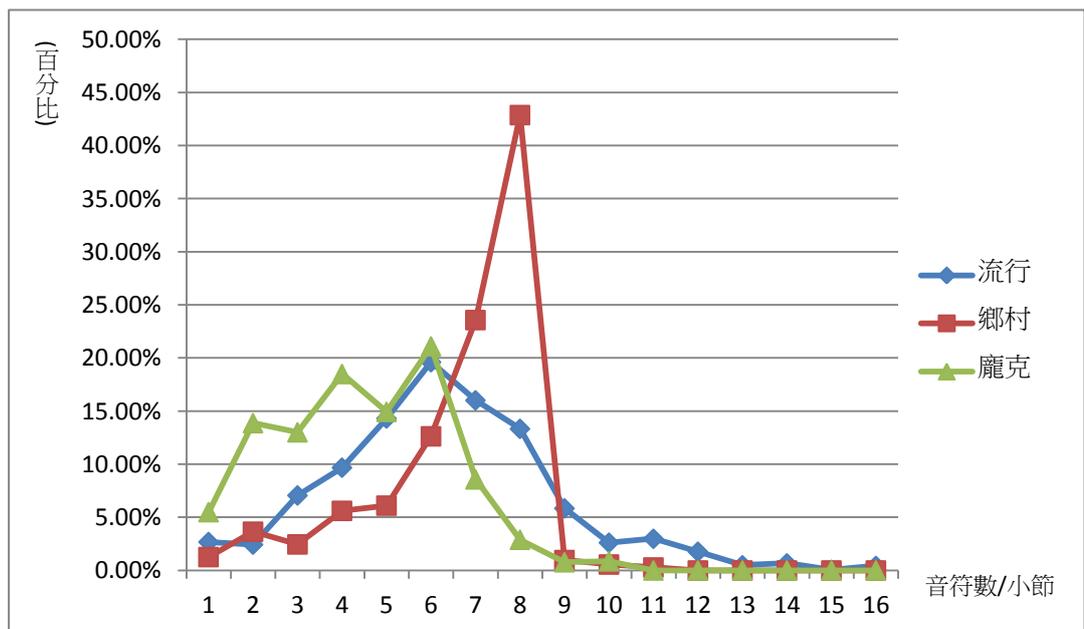
(三)、音高標準差



圖(二十)音高標準差

音階分布流行鋼琴、鄉村的音域範圍較廣，龐克則較窄。推測原因：流行鋼琴樂多由流行歌改編，而流行歌多在展現旋律的優美及歌手的歌喉，節奏明快的鄉村樂旋律常在兩個八度間游走，音域自然較廣；旋律簡單的龐克，只在表現強烈的節奏感及歌詞的創新，音域亦較窄。

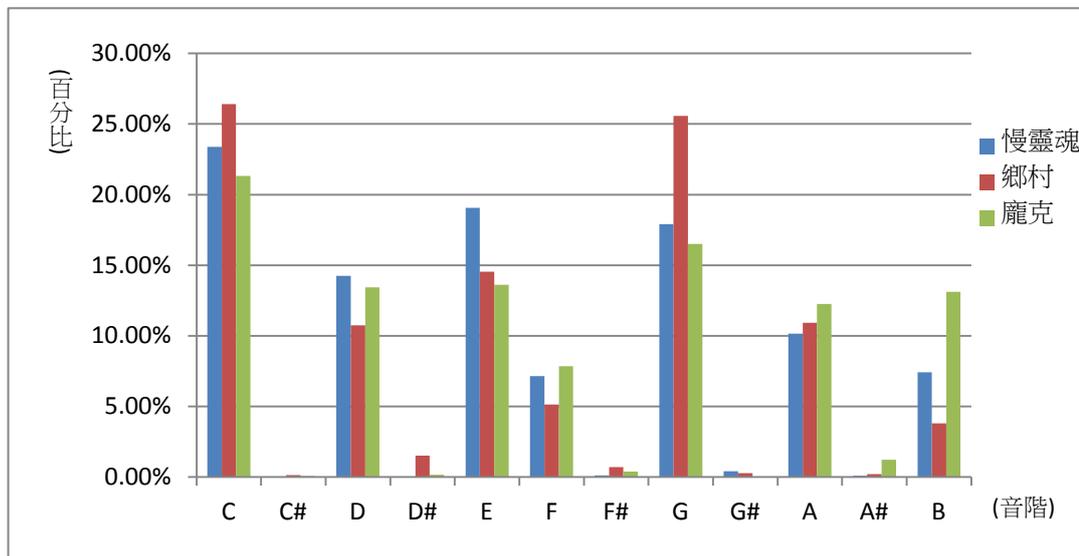
(四)、小節音符數



圖(二十一)小節音符數

2~8 個音符是三種曲風最常出現的小節。其中流行音符各數種類較為廣泛；龐克音符數少的小節較多；鄉村則音符數多的小節較多，8 個音符的小節尤其突出。針對音高保留法，可作為使用者輸入音樂時的參考。

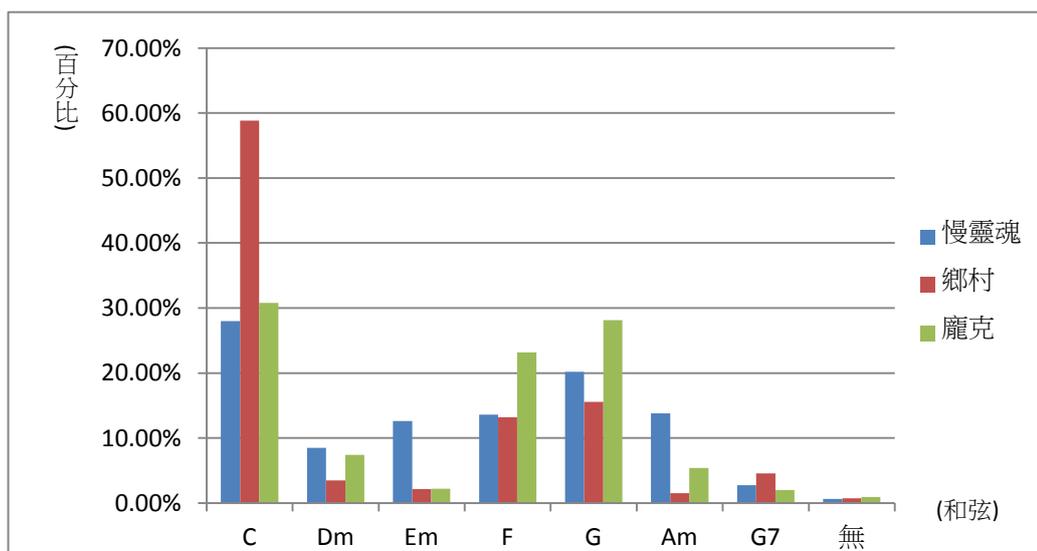
(五)、音階分布



圖(二十二)音階分布

由於在統計前已把所有歌曲轉成 C 大調，其音階的組成音 (C, D, E, F, G, A, B) 自然會最多，其中 C 大調的主和弦組成音 C、E、G 較多。值得注意的是，鄉村樂的 G 明顯高出另外兩種曲風，F、B 兩音出現比例卻較低，很有可能是鄉村樂特有的音階形式。

(六)、和弦分布



圖(二十三)和弦分布

由於所有歌曲都已轉成 C 大調，其主和弦 C 和弦，在三種曲風的比例都是位居最高，其中鄉村更是明顯。另外屬於 C 大調正和弦的 F、G 亦較多，屬於副和弦的 Dm、Em、Am 則出現較少。至於七和弦 G7 本是不常用的和弦。

(七)、和弦狀態轉移矩陣

流行鋼琴	C	Dm	Em	F	G	Am	G7	Null
C	46.04%	4.74%	7.01%	24.72%	37.45%	14.29%	41.33%	57.89%
Dm	4.80%	16.38%	16.46%	5.97%	3.93%	16.17%	2.67%	0.00%
Em	8.17%	6.90%	21.95%	6.82%	8.80%	22.91%	30.67%	0.00%
F	13.10%	16.38%	16.77%	15.34%	7.87%	15.36%	0.00%	5.26%
G	20.23%	44.83%	4.57%	25.57%	22.28%	11.05%	13.33%	5.26%
Am	5.97%	7.76%	32.01%	10.23%	17.42%	19.41%	0.00%	5.26%
G7	1.30%	2.59%	0.61%	11.08%	1.12%	0.54%	12.00%	5.26%
Null	0.39%	0.43%	0.61%	0.28%	1.12%	0.27%	0.00%	21.05%

鄉村	C	Dm	Em	F	G	Am	G7	Null
C	75.57%	24.29%	2.22%	26.12%	37.76%	0.00%	46.35%	84.21%
Dm	0.63%	22.86%	11.11%	1.79%	4.69%	25.00%	0.00%	0.00%
Em	2.90%	0.00%	36.67%	0.00%	0.52%	35.71%	0.00%	0.00%
F	7.87%	0.00%	50.00%	60.94%	1.04%	0.00%	0.00%	0.00%
G	9.01%	27.14%	0.00%	7.14%	41.93%	39.29%	10.42%	0.00%
Am	0.19%	25.71%	0.00%	0.00%	1.82%	0.00%	0.00%	0.00%
G7	2.83%	0.00%	0.00%	4.02%	11.98%	0.00%	43.23%	0.00%
Null	1.01%	0.00%	0.00%	0.00%	0.26%	0.00%	0.00%	15.79%
龐克	C	Dm	Em	F	G	Am	G7	Null
C	53.74%	8.65%	8.51%	28.82%	20.17%	7.78%	24.14%	13.33%
Dm	6.33%	49.04%	0.00%	2.96%	1.95%	0.00%	0.00%	0.00%
Em	3.45%	0.00%	31.91%	1.23%	1.95%	0.00%	0.00%	0.00%
F	13.24%	30.77%	44.68%	45.81%	16.49%	15.56%	17.24%	26.67%
G	16.89%	9.62%	0.00%	17.24%	56.18%	16.67%	48.28%	26.67%
Am	4.41%	0.00%	14.89%	0.49%	0.87%	60.00%	0.00%	0.00%
G7	1.34%	0.00%	0.00%	3.20%	1.52%	0.00%	10.34%	0.00%
Null	0.58%	1.92%	0.00%	0.25%	0.87%	0.00%	0.00%	33.33%

矩陣元素代表代表列和弦接到行和弦的機率。每行總和皆為 100.00%

二、轉換結果探討

	資料庫代換法	音高代換法
優點	<p>1. 旋律、節奏都會被改變，如果銜接順暢，將更能符合該曲風特性。</p> <p>2. 歌曲整體樣貌被翻轉，有時能給人意想不到的驚喜感。</p>	<p>1. 保留輸入歌曲的旋律走法，降低了連貫性不足的風險。</p> <p>2. 僅節奏改變，很容易聽出歌曲原貌。</p>
缺點	<p>1. 輸入的歌曲常常找不到相似度高的片段，輸出結果面目全非。</p> <p>2. 各音樂片段間大多互無關連，輸出結果連貫性明顯不足。</p>	<p>1. 除節奏、樂器外，無法彰顯曲風主旋律音高方面的特性。</p> <p>2. 候選片段會被音符數目限制住。</p>
共通點	<p>優：樂器、節奏皆能達到指定曲風的特質。</p> <p>缺：無考慮到曲風和弦走法的特性。</p>	
特殊狀況與改善方法	<p>音符高低的可能組成有很多種，提高了找到相似片段的難度。能在比對時利用平移的方式，集中音樂片段的音高八度範圍，使音名相同的音樂片段也能被視為相同。</p>	<p>若找不到符合該小節和弦及音符數目的片段，先容許比對音符數目差距為 2~3 個音符的小節，若還是找不到，則改成與其最相近的和弦，依其後選和弦的順序一一嘗試。</p>

三、哼唱輸入遇到的問題與解決方法

- (一)、音高校正：大多數的人並無法將音很準的固定唱在對的音高上，甚至在一個音之內都可能有浮動。但相鄰近的音程差大致上是比較準確的。因此取得判定出兩兩音符的差值後，將第一個音移到標準整數音高上做為基準點，在把差值四捨五入後依序算出其他的音高。
- (二)、音拍校正：音高之間間斷大多並不會剛好落在精準的拍子上，因此找出所有的切斷點並找尋與所有標準斷拍處差值最小者並以之替代。

四、應用價值

(一)、 作為遊戲或影片之配樂

可依遊戲或影片場景不同而做各種的曲風改編，簡單豐富變化多。

(二)、 音樂教學的輔助教材

幫助學生認識不同曲風的音樂特性及歌曲欣賞。

(三)、 提供使用者娛樂

讓使用者自行輸入歌曲改編曲風，娛樂兼欣賞音樂，改編歌曲的新風貌令人有驚奇感。而使用者若是自己編奏一段簡單的 MIDI 歌曲，或是直接使用方便的哼唱歌曲輸入，經由本程式的改編能大幅提高原本歌曲的豐富度並附有曲風的特色，是一簡單讓使用者作曲的方法。

五、未來展望

(一)、 更好的代換法

嘗試以不同長度音樂片段為單位進行比對代換，能否找出相似度與連貫性間的平衡點。另外，如果在重組時，加入一些樂理建構的篩選，實驗是否能改善整體流暢性。

(二)、 擴充資料庫大小

目前各曲風的歌曲數目皆稍嫌不足，多半是因為人工的前置處理繁雜又費時。期望未來能研發出自動前置處理系統，對於各種曲風的 midi 檔都能作結構化的裁剪，如此一來將能大大提升擴建資料庫的效率。

(三)、 增加曲風種類

先前另外考慮過的曲風如爵士、搖滾樂，都因其形式多變、太多分支、歌曲之間的差異性大，導致程式處理不易而被捨去。最後決議以慢靈魂、鄉村、龐克三種歌曲間差異不大，調性、主旋律、和弦也容易判定的曲風進行實驗。然而，目前僅對曲風作粗分，未處理到「曲風中還有曲風」的問題，期望未來能將曲風的分類做得更加完善、多樣。

(四)、 更完善的檢驗法

目前的方式是直接從曲風資料庫中取出音樂片段來代換，其流暢性卻無從判定。一首歌好不好聽，人耳最能分辨，但為了講求客觀，應嘗試找到將其量化的方法，或者藉由大量的問卷調查以提高可信度。

(五)、 正確判斷樂句和和弦

目前是以時間單位切割音樂片段，並不能以樂句做切割，而和弦則時兩拍作為一單位做判斷，不能保證一個小節只有兩個和弦，可能導致和弦的判斷錯誤。若能精準判斷樂句和和弦作為切割與代換，將能有連貫性更高的樂曲產生。

柒、 參考資料及其他

- 一、流行鋼琴網：<http://www.popiano.org/>
- 二、Free MIDI Files: <http://ifnimidi.com/>
- 三、MIDI File Format <http://www.sonicspot.com/guide/midifiles.html>
- 四、WAV File Format <http://www.sonicspot.com/guide/wavefiles.html>
- 五、Wikipedia - General MIDI: http://en.wikipedia.org/wiki/General_MIDI
- 六、柯佩芝(2009)以旋律及和弦特徵分類曲風及和弦伴奏自動配置。大同大學資訊經營研究所碩士論文
- 七、鄭雯妮(2008)以統計方法與音樂理論為基礎之和弦辨識系統。國立清華大學資訊工程學系碩士論文
- 八、顏輝智、呂仁園(2006)基於旋律追蹤及節拍追蹤的歌聲音符聽寫系統
International Workshop on Computer Music and Audio Technology
- 九、Audio Signal Processing and Recognition 張智星
<http://neural.cs.nthu.edu.tw/jang/books/audiosignalprocessing/index.asp>