基於旋律追蹤及節拍追蹤的歌聲音符聽寫系統

A MUSIC NOTE TRANSCRIPTION SYSTEM FOR SINGING VOICE BASED ON MELODY AND BEAT TRACKING

[1] 顏輝智 [2] 呂仁園

[12]Institute of CSIE, Chang Gung University, Taoyuan 333, Taiwan

[1]hzyien@msp.csie.cgu.edu.tw, [2]rylyu@mail.csie.cgu.edu.tw

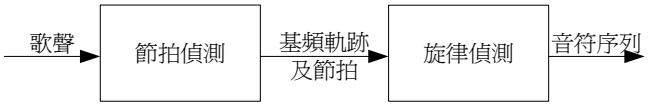
摘要

本文實作了一個音樂轉譯系統，目的要將一段歌聲（Singing Voice）用自動轉寫成音符（Note）序列。系統包含了旋律追蹤（Melody Tracking）以及節奏追蹤（Beat Tracking)兩個子系統。旋律系統沿用 Sequential Adaptive Round Semitones（SARS） 和 Tune Map 兩種方法做旋律追蹤；而節奏追蹤爲新加入的方法，用來偵測歌曲節奏。旋律追蹤使用 SARS 動態調整旋律（Floating Tune Melody）的問題，並以 Tune Map 將音樂文法模型應用到 SARS 的結果上找出正確的音符序列。節奏追蹤以歌唱者提供的輔助節拍增進節奏的辨識效果。SARS、Tune Map 與 Beat Tracking 結合的方法將使系統效能有顯著的提升

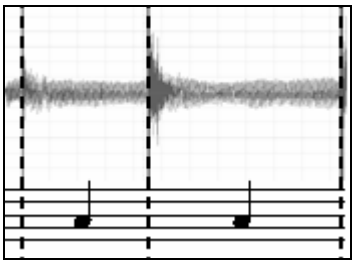
1. 概論

音樂轉譯的定義爲：「聽到一段的音樂聲後，將聽到的部分用音樂符號寫下來的動作」[1]。其實也就是將聲音的信號以音符來表示，音符中則包括了：音高（Pitch）、時長（Duration）、以及其他演奏時所需要的資訊。本系統主要的功能就是以節奏追蹤的方法找出音符與時長，再以基頻和節拍爲輸入，用旋律追蹤的方法找出音長和音高，最後組合而成音符（圖表 1）。

許多樂器可以產生出穩定的基頻信號，所以針對這些樂器所設計的轉譯系統效果都蠻好的。但是以唱歌來說，大多數人都不是專業的歌手，所以很難像樂器那樣發出固定基頻的聲音；歌者沒有辦法穩定的唱出特定旋律，很容易唱出標準音階之外的聲音；歌聲也會隨著歌者的心情上下變動，同一個音不同時候唱出來的音高跟時長會不一樣；演唱的速度會在一定的範圍內變動，表示不是一個定值。種種的不確定因素，使得系統無法預期會收到怎樣的基頻信號，讓歌聲轉譯爲音符變得很不簡單。



圖表 1 歌聲轉譯系統架構圖



圖表 2 聲音的時間長度不一，但是表示同樣長度的音符。虛線爲節拍處

每個音符都帶有時間長度的資訊，如二分音符、四分音符，這表示前者的持續時間是後者的兩倍，搭配演奏速度，就可以知道樂曲有多長的演奏時間。從樂譜中所得到的這種時間資訊，卻很難在歌聲中找到。

因爲唱歌的速度會隨歌唱者的心情改變，使得節奏變得不穩定的，所以我們假設歌者在唱歌時心中會有一個節奏，節奏的速度可能會改變，使拍子的間隔不一樣，但是拍與拍間的音樂元素是一樣表示一拍，

也就是說如果有兩個不同長度的音，但是同樣都在兩個拍子之間，則我們認定這兩個音是同樣一拍。（圖 2）

因此我們假設歌者在唱歌時一併用手打拍子，讓節拍聲一併被錄進系統中，這樣作的好處是：打拍子不會增加太多困擾、這個方式產生的聲音很容易就能偵測出發生的時間、而且節拍聲音只在一瞬間發出，對基頻偵測的結果影響不大。

對於有節拍聲的歌聲，我們攫取其特徵後作分類便可以將各個時間分類出節拍以及分節拍兩類。節拍的資訊可以回過頭來檢查基頻偵測的結果，並調整爲適當的數值，這讓旋律追蹤的來源訊號更加的正確，也提升了辨識效果。

旋律追蹤的目的就是要將基頻數值轉換爲音符代號，但是因爲人聲的不穩定性，所以我們使用了 Sequential Adaptive Round Semitones （SARS）作旋律追蹤，並且以音樂文法調整輸出的音符序列。SARS 在轉換過程中參考相鄰時間點的基頻變化量，動態調整轉換的結果，使不穩定的影響力降低；並且再以樂理知識建立了音樂文法的限制表，接著用 Tune Map 調整 SARS 輸出的音符系列，使其符合音樂文法的限制，進而達成轉換的目的。

歌聲轉譯系統除了可以幫助音樂表演者之外，還可以讓一般沒受過專業訓練的人可以對音樂更有興趣。所以這個系統必須讓人可以自然的歌唱而不需要其他的限制或規則，同時也應該要有辦法轉譯正確的歌聲並且將歌曲修正到歌者想表達的內容。

2. 相關研究

2.1. 節奏追蹤（BEAT TRACKING）

節拍偵測是一個研究已久的領域，這個領域許多人針對不同的資料作假設並作研究解決他。資料大致可以分爲兩類：有標記的資料（如 MIDI）以及純聲音的資料。

在偵測聲音節奏的部分，Simon Dixon發展了一個從數位音樂中偵測節拍的位置的軟體，他將聲波信號中突然增強的信號處偵測出來，接著使用群聚的方式將相近的位置找出來，最後以多個位移時間不同的偵測器去比對有多少個偵測出的節拍位置跟偵測器所走過的路徑是一致的，最多一樣的那一組偵測器所代表的節拍就是要輸出的結果[2]。

M.Goto 假設輸入的歌聲都是爲4/4拍，節奏介於每分鐘61-185個四分音符之間，並偵測出聲音事件發生的時間、和絃改變的時間、以及利用鼓聲的樣板比對出有鼓聲的時間，用這三個參數將聲音的節奏資訊轉換出來[3]。

這些方法都是針對專業歌手的歌聲或唱片音樂的信號作節拍偵測，但卻是對於一般人不穩定的歌聲就沒有特別敘述。

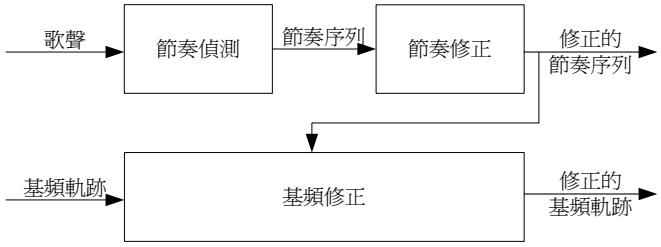
2.2. 旋律追蹤 （MELODY TRACKING）

最簡單的旋律轉譯法就是將音分直接轉換爲音名編號，轉換是將音分值除以 100，小數部分則四捨五入轉爲整數。這個方法稱作 Round MIDI，但這樣轉換對於動態變化的人聲處理不佳，所以之後才有了許多種改良的方法。

McNab et al. 假設每個歌者都有自己的參考音，唱出的音高則是相對於這個參考音的聲音，他們提出了名爲 ”Moving Tuning” 的方法，這方法假設了一個會改變的位移量，它會跟著前一個聲音而改變，將這個位移量加上歌聲中的音高值後再轉換爲音名編號。[4]

Haus et al.提出的方法爲 ”Haus’ Modifued Round MIDI”，他們也是假設歌唱者有自己的參考音做爲參考音高，唱出的音高距離想表達的音高值誤差要最小，所以也假設了一個會變動的位移量，這個位移量加上原有的音高轉換爲音名編號後，調整變動量使得轉換後的結果距離原有的音高要最小。[5]

Round Interval 的方法計算相鄰兩個音高的差值，將這個差值轉爲最接近的音名編號位移量，再加上前一個時間點的音名編號，就成爲目前音高的音名編號。



圖表 3 節奏偵測流程圖

3. 節拍追蹤 （BEAT TRACKING）

我們假設：歌唱者在唱歌的同時以拍手的方式產生正確且清晰的節拍訊號，這個訊號連同歌聲經由特徵擷取後再分類出節拍的時間點，所得時間點經過檢查去除重複偵測的部分，最後再回過頭來將基頻的信號作斷音的處理。

這些步驟，依序是節拍偵測、節拍修正、基頻修正。（圖表 3）

3.1. 節奏偵測（Beat Detection）

這個步驟目的在偵測歌唱者拍打聲音的時間位置。

我們將聲音信號從時域信號轉換爲頻域信號，轉換出的每一個音框，表示特定時間點的聲音資料。

再對個別音框計算與拍手聲有正向關係的特徵參數：

令時間 *t* 且頻率 *f* 的能量強度爲 $x(f,t)$ ，則：

1. 時間 *t* 的頻率能量總和

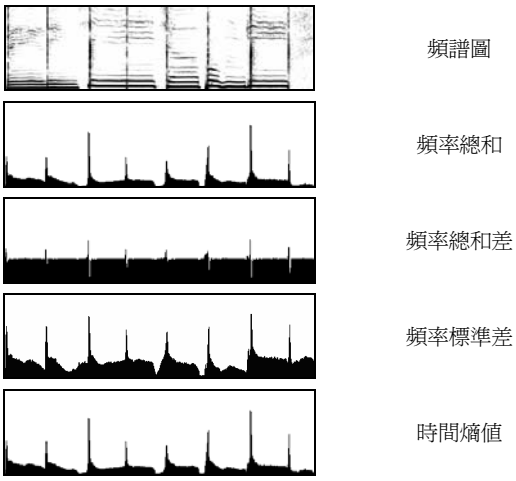
$$\mu_f(t) = \sum_{\forall f} x(f,t)$$

2. 時間 *t* 與時間 *t-1* 的頻率能量總和差

$$\triangle\mu_f(t) = \left| \sum_{\forall f} x(f,t) - \sum_{\forall f} x(f,t-1) \right|$$

3. 每個時間點的頻率能量標準差

$$\sigma_f(t) = \sum_{\forall f} x(f,t) f^2 - \mu_f{}^2$$



圖表 4 頻譜與特徵

4. 令 $Sum = \sum_{\forall t} \sum_{\forall f} x(f,t)$

且 $x'(f,t) = \dfrac{x(f,t)}{Sum}$

則每一個時間點的頻率熵值

$$E_T(t) = \sum_{\forall f} x'(f,t) \log \frac{1}{x'(f,t)}$$

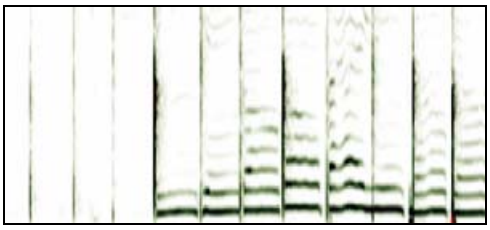
頻譜圖以及計算出的特徵結果如圖表 4。

接下來以 K-Means 分群法將每個時間點依照四項特徵分成節拍以及非節拍兩類，初始值則從所有時間點中選定四項特徵強度值最大的那點爲節拍類，非節拍則選定全部最小的那一點，分類時以阿基理得距離作分類的依據。分類完成後就可以得到節拍以及非節拍兩類。

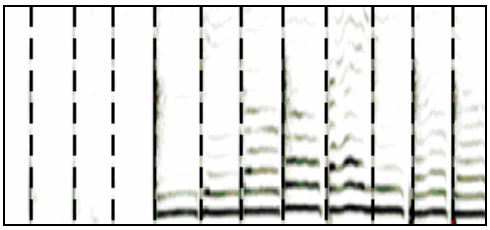
3.2. 節奏修正（Beat Correcting）

偵測出的節拍點可能會發生相鄰的兩個時間點間距離太小的情形，這發生的原因有幾種：

- 氣音及爆炸音計算出的特徵值跟產生節拍的聲音相似。。
- 產生節拍的聲音時間跨越多個音框，可能會分類出連續好幾個節拍時間點。



（A）



（B）

圖表 5 （A） 含拍手聲的歌聲頻譜
（B） 同一個歌聲頻譜，虛線爲節拍處

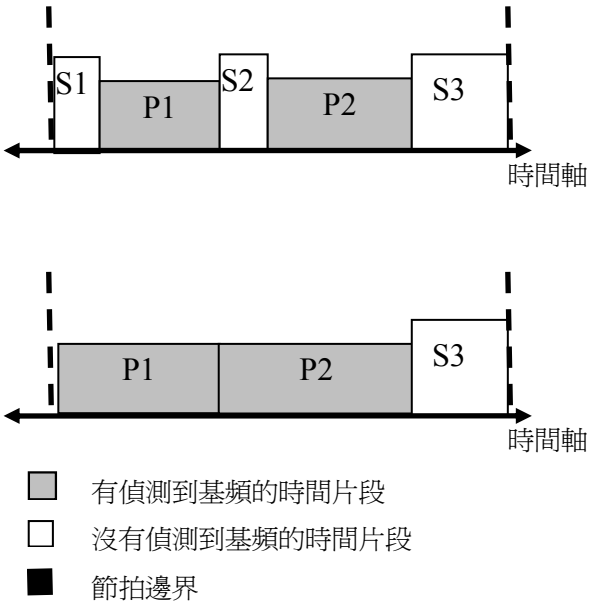
所以在偵測節拍完成後，要對節拍位置距離太接近的偵測點作合併的動作，作法是依照時間順序檢查所有的節拍點，檢查的過程中要記錄前面累積的平均節拍點距離，以這個平均距離作一個基準，如果檢查的節拍點跟前一個節拍點的時間差小於平均節拍距離的一半，則將後面那一點去除。節拍偵測結果如圖表 5。

3.3. Pitch Contour Correct

歌曲的休止處、文字中子音的部分和字與字之間的短暫停在歌聲中會偵測不出基頻，而後兩者因爲跟帶有基頻信號的聲音很接近，所以就會影響到旋律追蹤的結果。這部分則可以利用節拍的時間資訊判斷這些無基頻信號的時間片段，進而決定將這些片段保留、延長、或是與前後的片段合併。

作法是在歌聲做完基頻偵測後，將有基頻和無基頻的相鄰同類點收集成片段，每個片段的持續時間由絕對的時間長度改爲相對於節拍長度的時間，使歌聲變成以片段爲單位的片段序列，而每個片段序列都有一個相對於節拍的持續時間。

在檢查前先決定一個最小輸出單位時間 T，這是旋律偵測結果的最短音，我們設定爲節拍時間的 1/16。小於單位時間的時間片段必須合併或刪除。



圖表 6 片段合併前後，上圖爲合併前，下圖爲合併後

首先對無基頻的片段作檢查，如果片段的時間長度小於 T，且下一個片段沒有跨越到另一個節拍區，則將這個片段與下一個片段合併，若這個片段的長度爲 m，下一個片段的長度爲 n，則合併後片段的基頻值使用內插法的方式，將原先 m 個點的數值擴展爲 n 個點。

接下來在節拍中的片段會只剩下有基頻的片段以及時間較長的無基頻片段，對於有基頻的片段也要作檢查。如果有基頻片段的時間小於 T，則將片段與下一個片段合併，基頻值也是以內插法的方式補上。

節奏偵測的步驟完成後，便可以得到如圖表 5 所示的節奏時間點，而使用節奏時間處理過的基頻資訊將使用於下個步驟。

4. 旋律追蹤 （Melody Tracking）

歌聲被轉換爲基頻序列後，經過旋律追蹤的處理輸出爲音符代號，這個部分經過兩個步驟的運算：“Sequential Adaptive Round Semitones”（SARS）以及“Tune Map”。

這邊我們使用在電腦音樂領域中已經被定義出的音符代號 – 音名編號（MIDI Number），音名編號 N 與基頻數值 f （頻率，單位爲 Hertz）的對應關係

爲：

$$N = 69 + 12 \times \log_2 \frac{f}{440}$$

舉例來說，鋼琴鍵盤的中央 A 的頻率值爲 440 Hz，音名編號則是 69，比他高一個半音的 A#則是 70。

4.1. Sequential Adaptive Round Semitones（SARS）

SARS[6][7] 假設：

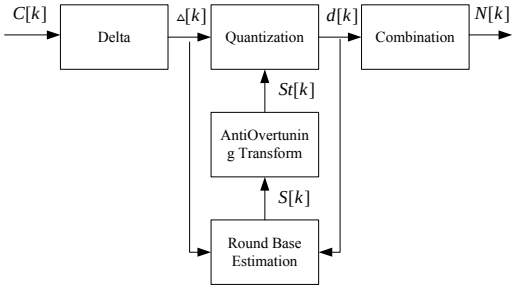
1. 歌唱者只發出跟聲調有關的基頻。
2. 唱出來的歌聲不一定要是 100 音分的整數倍。
3. 旋律的調性會隨著時間而改變，並且跟先前所唱過的音符有關。

舉例來說：當一個歌者在唱歌一段不斷升高的連續高音時，對於能力比較不足的歌者來說，唱出來的音其實與歌譜指定的音不是完全相同的，但是聽的人還是可以知道他唱的是什麼。SARS 的設計就是爲了解決這樣的情形。

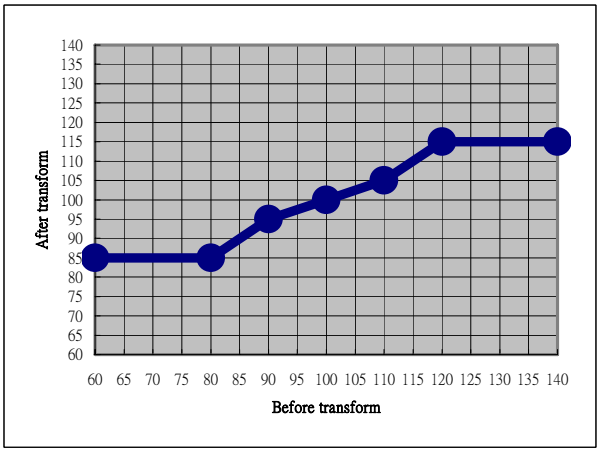
SARS 建立了一個統計模型 ARMA（Autoregressive Moving Average Model），這個模型將前幾個時間點跟目前時間點的基頻差異量、以及 SARS 輸出的結果一併綜合計算，得到一個變異參數輸出。這麼作是爲了模擬人類的聽覺表現，因爲人類的聽覺會受到先前聽到的聲音影響，而時間越久的聲音影響越小，所以這邊選擇所選擇的運算的聲音不會在記憶中停留太久。

圖表 8. 是 SARS 演算法的流程圖，其中 $C[k]$ 是輸入的基頻信號，計算與其前一個時間點的差異而得到 $\Delta[k]$，$d[k]$ 爲 SARS 計算後所得到的音符代號差異量，將 $\Delta[k]$ 與 $d[k]$ 放入 ARMA 產生的參數 $S[k]$，$S[k]$ 經過 Anti-overtuning 轉換，將過大或過小的數值壓抑爲適當的範圍，而成爲 $St[k]$，再將 $St[k]$ 與 $\Delta[k]$ 相加後就得到目前的音名編號位移量，最後將目前的結果與先前的結果合併後就得到輸出的音名編號 $N[k]$。

Anti-overtuning 轉換是爲了避免 ARMA 出現過大或過小的數值。在此將 ARMA 的結果經過線性轉換與



圖表 7 Sequential Adaptive Round Semitones（SARS）



圖表 8 Anti-overtuning Transform for SARS

邊界剪裁的處理，圖表 9 爲數值轉換前後的對照表。

4.2. Musical grammar constraint

4.2.1. Music theory

音樂理論提到：「一組八度音（Octave）被均分爲 12 個音符」。從這 12 個音符中選出特定的幾個音符，組成音符序列則稱爲一組調性（Scale）。每個音符都有一個羅馬字的名稱，這個名稱之爲音名（Degree）。音符系統中某些音符會有特別高的出現機率，這些音符則會被指定爲這首曲子的調號（Key Signature）。西方音樂中，最常出現的調號有 12 種大調及 12 小調。

每一個調性都對應到一組特定的結構：在大調（Major-Scale）裡面，第 “Mi-Fa” 和 ”Ti-Do” 之間是跨越一個半音，其他的部分則跨越兩個半音。

- Major scale: 2, 2, 1, 2, 2, 2, 1.
- Minor scale: 2, 1, 2, 2, 1, 2, 2.

當一段大調或小調的聲音片段被產生時，在這個聲調中的主音會成爲這個段聲音中央調，表示這一段

旋律是以主音爲基礎。

4.2.2. Constraint concept

旋律轉譯要解決的問題就是將基頻序列 $C[k]$ 對應到音名編號 $N[k]$。如果 $N[k]$ 是一個亂數的序列，則結果的搜尋空間有 128 個（對應到 128 個音名編號）。但八度音中是一段從一倍頻率到兩倍頻率的數值範圍，如 440Hz.～880Hz.。假設不考慮不同八度音的相異處，則每一個音符都是八度音內 12 種音符的其中一種。所以搜尋空間會降低爲 12。

雖然聲調的在歌曲中是亂數分佈的，但是一首有調性的歌曲往往會在某些聲調上有群聚的感覺，如 C 大調、E 小調…等。這些聲調結構如文法一般控制了個別音符出現的機率，這樣又使複雜度由 12 降低到 7。

4.2.3. The constraint table

因此我們建立了一個限制資料表，利用調性對於音符的限制減少搜尋空間（表格 1）。

表中的第一個欄位爲聲調位移量，其他欄位爲調性名稱，意思是當樂曲中第一個音符爲 I 時，可能接在後面的音符爲 I+0、I+2、I+4、I+5…I+12，也就是說若 I 爲 Do 時，則後密的音符爲 Re、Mi、Fa…到高音 Do。沒有調性名稱的空白處表示那是樂理規則中不合法的地方；負方向的意義也是一樣，只是要倒著數回去。

因爲小調和大調兩者在經過一個環狀的移調後就具有相同的結構，所以這個表格也可以應用在小調的曲目上。

聲調位移	調性名稱						
-12	I	II	III	IV	V	VI	VII
-11			III				VII
-10	I	II		IV	V	VI	
-9		II	III			VI	VII
-8	I			IV	V		
-7	I	II	III		V	VI	VII
-6				IV			VII
-5	I	II	III	IV	V	VI	
-4			III			VI	VII
-3	I	II		IV	V		
-2		II	III		V	VI	VII
-1	I			IV			
0	I	II	III	IV	V	VI	VII
+1			III				VII
+2	I	II		IV	V	VI	
+3		II	III			VI	VII
+4	I			IV	V		
+5	I	II	III		V	VI	VII
+6				IV			VII
+7	I	II	III	IV	V	VI	
+8			III			VI	VII
+9	I	II		IV	V		
+10		II	III		V	VI	VII
+11	I			IV			
+12	I	II	III	IV	V	VI	VII

表格 1 限制表格

4.3. Tune map

利用限制資料表，我們建立一個 Tune Map 的資料結構，用以檢查音符序列差值是屬於文法結構中的哪一個序列，已確認輸出的結果應該要是哪個音符序列。

之所以要作這樣的檢查，是因爲我們採用差異值的方式來表示音高，所以要知道目前的音是那個聲調，就必須追溯回第一個音，才會知道目前的音爲哪一個音高。但是因爲第一個音也需要參考點的資訊，使得第一個音也無法確認是什麼音調，就無法作爲一個參考點，造成音高無法確認。

儘管起始音的聲調無法確定，但是樂曲中的差異結構卻不會改變，因此可以用來比對目前的音高序列是哪一種聲音結構。

Tune Map 是一個類似二元樹的路徑圖，其中有兩種節點，一種稱作「單一分支（One-branch-node）」另一種則是「雙向分支（Two-Branches-node）」。

每個節點都對應到一個音符，節點內的數值則對應到相對於歌聲第一個音符音高差異值，而節點下方

的集合則對應到這個音符可能的調性名稱。

根節點表示歌聲的第一個音，接著就可以依照新的音增加節點。

如果相鄰的音名編號差值與這個差值四捨五入後的結果差異值 *d* 太大，若 *d* 除以 SARS 所產生的數值 *s* 超過±20%，則增加一個雙分支節點，兩個分支接到這個節點上，分支並接到新的節點上，節點值分別是：目前的音名編號加上（d/s）取無條件進位的數值，以及目前的音名編號加上（d/s）取無條件捨去後的數值。反之則增加單分支節點，新節點的值爲：音名編號加上（d/s）取四捨五入的數值。

節點增加時，要檢查限制表格檢驗新增加的節點是否正確。

在最後當新增加的節點已經不符合任何的聲調結構時，則表示發生變調的情形，我們就找到一個由葉節點到根節點的連接的聲調路徑。追溯這段路徑歌曲的旋律就可以被追蹤出來。

Tune Map 的結果可能不是唯一，表示有多個路徑存在於 Tune Map 中。 這是因爲在歌聲中音符的數量不足造成限制不充分的緣故。因此在 Tune Map 最後建立一個雙分支節點時，我們必須選擇一個最接近原基頻值的那一個節點值，並且將新的節點放到最接近的節點去。最後再由 Tune Map 從數值最接近原音高的節點追溯回根節點產生結果序列。

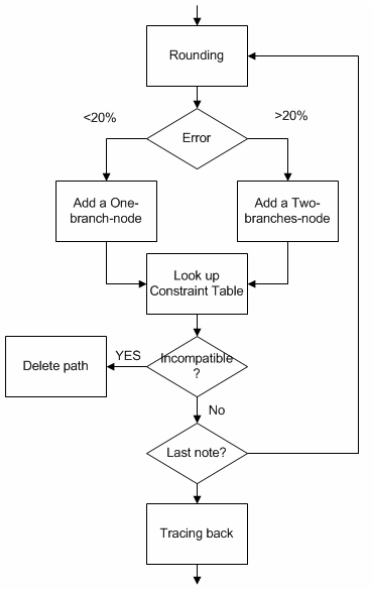
如果每一個節點都是取出數值最接近的那一個分支點，則結果將會跟純粹使用 SARS 的方法一樣。

圖表 10 爲 Tune Map 的演算法流程圖，圖表 11 則是分支找尋的示意圖。

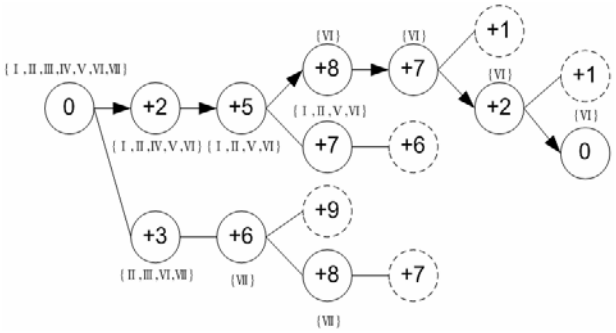
5. 結論與未來工作

本系統加入節奏偵測(Beat Tracking)的功能，讓原先只使用 Sequential Adaptive Round Semitones 與 Tune Map 的音樂轉譯系統，降低了偵測不出基頻的片段對於旋律追蹤時的影響，並且改善了 SARS 與 Tune Map 對於時間的辨識度不足的問題。

在簡易的實驗下，增加了 Beat Tracking 的轉譯系統，改善了原先無法正確運作的某些歌聲的問題。



圖表 9 Tune Map 演算法流程圖



圖表 10 Tune Map 分支找尋示意圖

未來除了需要收集資料並進行實驗外，節奏追蹤的方法也還需要更多的實驗資料，來作修正以及改善，希望未來能達到不需要輔助節拍也能夠正確辨認節奏的目標。

6. 參考文獻

[1] Martin, “A blackboard system for automatic transcription of simple polyphonic music”, MIT Media Laboratory Perceptual Computing Section Technical Report No. 399, 1996

[2] S. Dixon , “A Lightweight Multi-agent Musical Beat Tracking System” , Pacific Rim International Conference on Artificial Intelligence , 2000

[3] M. Goto , “An Audio-based Real-time Beat Tracking System for Music With or Without Drum-soundsVol.30, No.2, pp.159-171, June 2001

[4] R. J. McNab, L.A. Smith and I. Witten , “Signal Processing for Melody Transcription”, Working Paper 95/22, Dept. of Computer Science, University of Waikato, 1995.

[5] G. Haus, and E. Pollastri “An audio front end for query-by-humming systems”, International Symposium on Music Information Retrieval （ISMIR）, 2001.

[6] Chong-kai Wang, Ren-yuan Lyu, and Yuang-chin Chiang ”A singing transcription system using melody tracking algorithm based on Adaptive Round Semitones （ARS） plus music grammar constraints”, Stockholm Music Acoustic Conference （SMAC）, 2003.

[7] Chong-kai Wang, Ren-yuan Lyu, and Yuang-chin Chiang “A Robust Singing Tracker Using Adaptive Round Semitones （ARS）” Proc. of 3rd International Symposium on Image and Signal Processing and Analysis （ISPA03）, Italy, 2003.

[8] Chong-kai Wang, Ren-yuan Lyu, Yuang-chin Chiang, “An Automatic Singing Transcription System with Multilingual Singing Lyric Recognizer and Robust Melody Tracker”, Proc. of 8th European Conference on Speech Communication and Technology (EuroSpeech 2003) , Geneva, Switzerland.

[9] Chong-kai Wang, “An Integrated Singing Recognition System Using A Robust Melody Tracker and A Multilingual Singing Lyric Recognizer” Master thesis, Chang Gung University, 2003.

[10] *Chong-kai Wang, [2]Ren-yuan Lyu, [3]Yuang-chin Chiang ,* “A Floating-tuning melody tracking technique using Sequential Adaptive Round Semitones (SARS) and tune map” , 1ST WOCMAT , 2005