

# 箏樂滑音之自動識別

沈瑞祥

中華大學資訊工程學系

Email:m09602003@chu.edu.tw

劉志俊

中華大學資訊工程學系

Email: ccliu@chu.edu.tw

中國民族樂器彈奏技法的特色是對於「韻」的重視，在常見的民族樂器中，如古箏、古琴、揚琴、笛，尤其是對古箏與古琴類的樂器而言，特別追求「以韻補聲」的境界。但自古以來，有關韻的演奏分析與傳承都只能夠以師徒間親自彈奏教授的方式進行；在本研究中，我們希望可以藉由對古箏滑音技法進行自動偵測，能將古箏演奏樂曲轉換為樂譜，並標示出其中的滑音及其特性。滑音識別技術是對傳統樂器內涵分析的利器，此外，相較於傳統的師徒面授方式，此技術更有利於後人自我學習時所用。

**關鍵詞**—滑音、記譜、起音點偵測、音高偵測、滑音識別

## 一、研究背景

隨著多媒體的快速發展以及多媒體壓縮技術的進步，目前有大量的多媒體資料在網際網路上快速的傳播與交換，因此對在多媒體資料的分類與查詢技術，顯得日益重要。所以，有關多媒體資料內涵式分析之相關研究，愈來愈受學術界的重視。

在各種多媒體資料中，音樂是最重要的多媒體資料之一。早期的音樂內涵分析，主要是由音樂理論學者與心理學家，以人工的方式對音樂作品做分析與探討。但近些年來在音樂內涵分析方面的研究，逐漸轉為透過擷取音樂低階特徵，對主要的音樂特性如旋律、節奏、調性、速度等，在偵測、比對、分析、分類、索引等方面以自動技術進行探討。但這些研究與探討，幾乎是以西方音樂與樂器為主軸，而以鋼琴、鼓等主題較為常見。而對於我國傳統上極具特色的民族樂器與音樂，如古箏、二胡、揚琴、笛、琵琶、簫等樂

器之內涵式研究分析，目前仍是以定性描述式的探討為主。

古箏樂曲中，通常右手演奏技法為表聲技法，而左手演奏技法為表韻技法，如典型的八大表聲技法：托、擊、抹、挑、勾、剔、打、摘[1]；典型的表韻技法有滑音、顫音、揉音等左手技法中，有相當多的基本型是以滑音作為基礎演變而來。因此，滑音在表韻技法上扮演著相當重要的角色。箏樂中，如何使用右手表聲技法與左手表韻技法使一首箏曲的意境呈現，是最主要的探討課題之一。右手技法部份基本上只是音的產生，而左手技法部份才是箏樂的精華所在——韻的表現。

一個滑音的彈奏，可能會因假指甲材質差異、彈奏者的心情、彈奏時觸弦角度、彈奏時力度大小、彈奏時觸弦深度之間不同的組合而產生不同情境。正因為滑音變化如此之大，也因此相當的難以自動偵測與分析。所以我們初步研究目標，便以箏樂之滑音技法自動識別技術來進行探討。

黃好吟[9]在箏曲彈奏技法與記譜研究中，對於各種技法有相當明確的定義。書中也將許多箏曲，用手寫的方式一一記錄下來，對於樂曲的傳承與教學方面，相當珍貴。對於滑音，依滑音的個數、與範圍來定義，主要可區分成上滑音、下滑音、連線滑音、上迴音、下迴音等五種。

張儷瓊在記譜研究中[1]，將所有記譜派別與記譜差異度做了整合分析與比較，也對於各派別技法使用技巧加以深入說明。對於古箏樂曲彈奏品質分析上，葉娟初[3]與翟源[4]提出了彈

奏技法之使用對於箏樂整首樂曲品質的差異分析。

焦力對於滑音定性描述的研究中[6]，對於滑音做了滑音種類以及滑音彈奏間差異感分析。文中對滑音功能亦有明確的說明，此外對如何練習滑音也有相當深入之討論。在其他滑音延伸的技法中，雷華也對於夾彈、提彈技法的基本彈奏方式提出分析研究[5]。

林雅琇針對潮州箏曲，運用不同素材來詮釋樂曲的差異性與特性，展現出潮州箏樂與其他派別箏樂的不同[7] [8]。另外文中也以潮州名曲《寒鴉戲水》做為例子，對於不同版本差異進行深入比較，探討各門派在詮釋《寒鴉戲水》的差異。

音樂識別可應用在多重奏樂曲演奏中的樂器識別、利用音色特徵做音樂上的識別、以及利用基頻偵測技術做音樂混合樂器的識別。在中國民族樂器古箏的彈奏識別上，也有人利用顫音特徵，做樂曲相似度比較分析[2]。

一首音樂是由一連串樂音(tones)或音符(notes)所組成。而在一首音樂中，樂音與音符的數量是相當龐大的難以用人工方式進行切割，所以音樂分析的首要技術便是能將一首數位音樂以樂音為單位進行自動分段(segmentation)，以利後續的其他音樂特性分析與研究。因此起音點偵測(onset detection)便是音樂內涵分析首先必須具備的技術。

對於數位音樂的起音點偵測技術，最早是由Goto[16] [17] [18] [19] 提出一種架構在實體音樂訊號的即時起音點辨識方法。此方法藉由建立節奏偵測模型，利用節奏在樂理上的特性，可以有效分析出音樂的節奏資訊。

Klapuri[10]將音樂訊號經由梳形濾波器(comb filters)轉換，並將節奏在樂理上的特性帶入觀測機率模型，並且利用HMM來預測節奏出現位置的週期，最後帶入相位量測，可以找出起

音點出現的地方。

Dixon[12] [13] [14] 將三種現有的起音點偵測特徵值：頻譜通量(spectral flux)，相位偏移(phase deviation)，複數領域(complex domain)等偵測方法加以改良與整合，再加上調整判定參數的設定，可以達成極好的起音點偵測效果。經由實驗結果顯示，利用頻譜通量的改良方法，不僅計算方式簡單，正確率也是最高。

音高偵測(pitch detection)是語音分析研究中，重要的核心技術之一。常見音高偵測技術大致上可以分時域自相關分析法(autocorrelational analysis)、頻域的傅立葉轉換分析以及頻譜係數法分析等。根據的主要音高偵測實驗比較結果顯示，各種不同的分析法，都各有優缺點，如頻域上的方法，常常對泛音容易產生誤判，時域的方法，對低頻訊號有解析度上的問題，由這些實驗結果可得知，採用何種音高偵測方法，必須視待分析的音訊特性而定。

Tolonen[28] 提出一個音高分析模型，計算信號頻道之自相關與高頻通道，並且求出自相關函數。文中改進簡易自相關係數(SACF)成擴充型簡易自相關係數(ESACF)，並將其應用於多重基頻偵測與週期分析。

Kitahara[22] 在多重樂器演奏識別樂器研究中，利用基頻偵測與樂譜比對的技術進行音色特徵值擷取，再藉由PCA與LDA相關技術，來進行多重樂器識別。

本文主要研究目標在於針對箏樂，發展能自動識別其樂曲中滑音技法的技術。目前大多數在樂音識別方面的偵測研究，都是在探討對起音點、拍子、節奏、力度、音高、小節等音樂特性之識別技術，而我們希望不止只是單單對於音樂樂音識別，而是對音樂彈奏技法中的滑音進行識別。

滑音偵測技術有相當多的應用，說明如下：

- 箏樂自動繪譜之軟體開發：我們可以將技法偵測後的結果，自動產生該樂曲之樂譜與技

法標示。

- 古箏樂曲相似度識別與分類：可以應用於古箏樂曲與樂曲之間的相似度識別，或依其作者、地方文化等做曲風分類。
- 箏樂學習評分：應用在箏樂教學學習方面。例如在老師示範彈奏後，可以自動比對學生的練習是否和老師所示範彈奏之方式相似。
- 古箏樂器識別：在西方樂器識別方面的相關研究相當多，所以我們也可以將類似的技術應用於古箏樂器識別。
- 交響化國樂樂器配置分析：最近古箏漸漸受到西洋音樂影響，走向交響化。而我們所研究的技法偵測，可以應用在交響化的國樂演奏中，對中國民族樂器作配置分析。
- 箏曲即時彈奏記譜：在箏曲彈奏中，可以即時的記錄彈奏方式，將其自動製譜，利於未來箏曲數位典藏與研究之用。

本論文的章節結構說明如下：在第二節中，我們將介紹有關箏樂滑音識別整體架構。第三節說明箏樂起音點偵測技術。第四節將說明箏樂基頻偵測技術。第五節針對箏樂滑音判別技術說明，並且分別說明滑音兩大特徵：滑音變化區段時間間隔與滑音變化區段音程差異。第六節則對我們提出之箏樂滑音判別方法進行實驗及結果分析，最後總結本論文並提出未來研究方向。

## 二、箏樂滑音識別整體架構

### (一) 箏樂滑音識別的主要流程

滑音識別主要流程中，包括了古箏箏曲輸入、起音點偵測、基頻偵測與滑音識別等步驟，如圖 1 所示。各步驟涵義說明如下：

- 箏曲輸入：在箏曲方面，我們所採用的箏樂樂曲由古箏老師與台灣藝術大學學生所親自彈奏之樂曲，以及經典箏曲 CD 錄音。
- 起音點偵測：一首古箏箏曲是由一連串的樂

音組成。所以我們先利用起音點偵測技術將一首古箏箏樂切分為一連串的樂音，便於進行後續的分析。

- 基頻偵測：由於滑音技法會對音高產生變化，所以我們必須進行基頻偵測求得每個樂音的音高變化。
- 滑音判斷：在古箏滑音技法中，會形成各樂音之間音高與時值變化，我們可利用各樂音音高與時值變化，來進行滑音判斷。

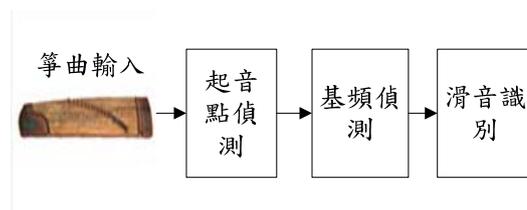


圖 1 滑音識別主要流程圖

### (二) 箏樂滑音的涵意及模型

箏樂滑音根據[9] 可大致分為上滑音、下滑音、連線滑音、上迴音、下迴音等五種。各種滑音的涵義定義如下：

- **上滑音**：右手撥絃後，左手隨即按絃使音升高二度或三度稱之。如圖 2 譜例所示，為一個由 4 音滑至 5 音之上滑音。



圖 2 上滑音範例

- **下滑音**：左手先按絃，使音升高二度或三度，待右手撥絃後，左手隨即放鬆，使音高復原稱之。如圖 3 譜例所示，為一個由 6 音滑至 5 音之下滑音。



圖 3 下滑音範例

- **連線滑音**：凡用加箭頭的弧線把兩個或兩個以上的音符連接起來，表示從第一個音符依次滑奏到最後一個音符，這種滑音稱為“連

線滑音”。如圖 4 譜例所示，為一個由 5 音滑至 6 音再由 6 音滑至 5 音，最後由 5 音滑到 6 音之連線滑音。

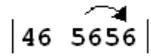


圖 4 連線滑音範例

- **上迴音**：右手撥絃後，左手即將音按高大二度或小二度或小三度，再迅速放回本音，形成音符的迴轉效果，這種滑音，稱“上迴音”。如圖 5 譜例所示，為一個由 6 音滑至 5 音再滑回 3 音之上迴音。

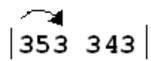


圖 5 上迴音範例

- **下迴音**：左手先按譜高音低一絃，使與譜音同音高，右手撥絃後，左手稍放隨即再按成譜音，使形成一回轉音，這種滑音稱為“下迴音”。如圖 6 譜例所示，為一個由 3 音滑至 1 音再滑回 3 音之下迴音。

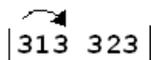


圖 6 下迴音範例

本文對滑音的識別研究，初步以上滑音與下滑音為主。在一個完整的上滑音與下滑音彈奏過程中，可依時間劃分為起滑區段(A)、滑音變化區段(B)與終滑區段(C)等三個區段。如圖 7 所示，各區段涵義說明如下。

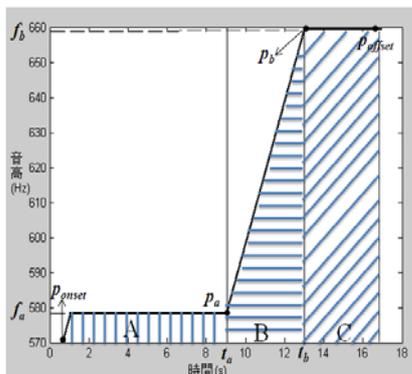


圖 7 滑音各區段涵意架構

- **起滑點區段(A)**：箏樂訊號中，一個樂音的開始彈奏之處稱之為起音點( $p_{onset}$ )；起音點之頻率值，即稱之起音點頻率( $f_a$ )。在此區段中，頻率大致上維持一定值。
- **滑音變化區段(B)**：為滑音技法偵測最重要的區段，包括起滑點( $p_a$ )、終滑點( $p_b$ )、頻率變化量( $\Delta f$ )、時值變化量( $\Delta t$ ) 等滑音主要參數，其中頻率變化量與時值變化量是識別滑音的重要特徵。滑音技法中，開始樂音與終止樂音之音程，即頻率變化量 $\Delta f$ ，大約相差大二度或是小三度。因此，起滑點與終滑點之音程，會產生約 200 或 300 音分的頻率變化量。整個滑音變化區段時值 $\Delta t$ 約佔滑音總時值範圍之四分之一，而此區段大致位於 2/4 至 3/4 滑音總時值之間。
- **終滑點頻率區段(C)**：這個區段中，包括了終滑點( $p_b$ )與終音點( $p_{offset}$ )。當頻率滑動至要求音準之時稱為終音點；滑音在終滑點之後會有一段頻率穩態的區塊，在這個區塊中的頻率值，即稱為終滑點頻率值( $f_b$ )。最後是整個樂音的結束點，我們稱之終音點( $p_{offset}$ )。

### 三、箏樂起音點偵測

本文所使用之起音點偵測方法結合 Dixon[12] [13] 及 Goto[18] [19] 兩種起音點偵測法。方法說明如下：

步驟一：前置處理

首先計算輸入箏樂訊號之波封(envelope)，將左右聲道取平均，再進行半波整流，將負波形方面轉換成正波形，並且在每間隔 10ms，以 100ms 為音框大小，求出振幅平均值後進行平滑化處理。

步驟二：古箏音域分頻

將古箏大致按照每根絃的音域範圍，使用一組帶通濾波器進行分頻。

步驟三：起音點特徵值計算

先前研究中，多數的研究都只單單以時域方面為主軸或是以頻域方面為主軸進行研究，而 Dixon 的方法則是將時域方面與頻域方面特徵結合，計算出兩類八種特徵值。在時域方面，包括振幅差、尖峰振幅差、振幅遞增值、尖峰振幅遞增值等；在頻域方面包括音量、尖峰音量、音量斜率、尖峰音量斜率等。八種起音點偵測特徵值的計算方法如下：

(A)時域特徵值

振幅差特徵值計算方式如下：

$$F_1[j] = \sum_{i=hjr}^{(hj+w)r} |S[i+1] - S[i]| \quad (1)$$

其中 $F_1[j]$ 代表第 $j$ 個音框之振幅差、 $S[i]$ 代表在 $i$ 時間時的訊號振幅、 $r$ 代表取樣頻率、 $w=0.1$ 秒代表視窗大小(window size)、 $h=0.01$ 秒代表前移值(hop size)。

尖峰振幅差特徵值計算方式如下：

$$F_2[j] = Peaks(F_1, j, 5, Threshold_1) \quad (2)$$

其中 $F_2[j]$ 代表在正負5個音框內的最高振幅差、 $Peaks$ 代表在指定範圍內的最大波峰值、 $Threshold_1$ 代表振幅變化選取的門檻值。

振幅遞增特徵值計算方式如下：

$$F_3[j] = 0.2 \sum_{k=0}^3 kF_1[j+k] - 0.3 \sum_{k=0}^3 F_1[j+k] \quad (3)$$

其中 $F_3[j]$ 代表振幅持續增加的程度。

尖峰振幅遞增特徵值計算方式如下：

$$F_4[j] = Peaks(F_3, j, 5, Threshold_2) \quad (4)$$

其中 $F_4[j]$ 代表在正負5個音框內的最高振幅遞增值。

(B)頻域特徵值：

音量特徵值計算方式如下：

$$F_5[j] = \sum_{k=0}^{255} Peaks(FFT(S', hjr', 512)[k], j, 5, 0) * FFT(S', hjr', 512)[k] \quad (5)$$

其中 $S'$ 代表 $S$ 中的訊號變化量

尖峰音量特徵值計算方式如下：

$$F_6[j] = Peaks(F_5, j, 5, 0) \quad (6)$$

音量斜率特徵值計算方式如下：

$$F_7[j] = \sum_{k=0}^{255} Peaks(FreqSlope[k], j, 5) * FreqSlope(S', hjr', 512)[k] \quad (7)$$

尖峰音量斜率特徵值計算方式如下：

$$F_8[j] = Peaks(F_7, j, 5, 0) \quad (8)$$

步驟四：起音點判斷

Dixon 利用這八種特徵進行起音點分類。一般研究都是使用特徵值進行起音點比對偵測，但 Dixon 的方法則是利用分類器進行起音點分類與比較，藉由基因演算法來偵測出起音點。本文則改用 kNN 分類器，使用八種特徵值預先訓練起音點判別程式來進行起音點偵測。

#### 四、箏樂基頻偵測

##### (一) Licklider 內耳模型

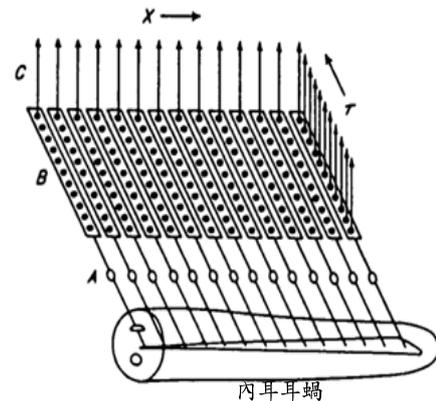


圖 8 Licklider 內耳模型

在[23]一文中，Licklider 提到在許多心理學感知科學的研究結果顯示，人耳聽覺對音高的感受呈現出雙重特性：音高的感受同時包含頻率(frequency)與週期(period)兩方面的性質。以內耳

的解剖構造而言，耳蝸(cochlea)的不同部位會對不同的頻率而起反應，而聽覺神經系統(nervous system)的功能類似對訊號進行自相關分析。據此，Licklider 提出利用一個二維模型來表示內耳對音高感受的雙重特性。如圖 8 所示，A 表示耳蝸不同部位  $x$  對頻率的過濾作用；B 為神經元對訊號產生延遲  $\tau$  時間的作用；C 為最後產生一組訊號自相關運算的結果。

在早期的電腦計算能力相當不足，而 Licklider 所提出的內耳模型，所需求的運算量相當龐大，所以提出內耳模型時，並沒有受到重視，現今，隨著電腦運算能力的不斷提昇，漸漸使得 Licklider 內耳模型的龐大運算變的可行。Lyon 強調聲音訊號的處理必需人耳聽覺模型為依歸 [25]，所以他根據 Licklider 的內耳模型提出一套內耳聽覺的計算模型(computational model)。此模型的優點在於對聲音的分析提供了更多的細節資訊，也對雜訊擁有更好的容忍度。

### (二)Slaney 音高偵測方法

Slaney 根據 Licklider 的內耳聽覺計算模型提出兩種實做方式[25]，並在 MATLAB 環境下開發了一套音訊處理函式庫 Auditory Toolbox[26]。以此模型為基礎，配合窄化自相關函數(narrowed autocorrelation function)來考慮泛音的影響，Slaney 提出一套音高偵測方法，可以在多重雜訊干擾下偵測聲音的音高[28]。由於古箏演奏具備餘音繚繞的特性，音高的偵測必須考量在大量殘響條件下進行，我們認為 Slaney 的音高偵測方法相當適合用來進行古箏的音高分析。

Slaney 的音高偵測方法主要架構如圖 9 所示，對於古箏的音樂訊號先經過一組濾波器分析出不同頻帶，來模擬耳蝸基底膜(basilar membrane)的作用。然後通過半波整流濾波器(half wave rectifier, HWR)模擬內毛細胞(inner hair cells)的功能，最後通過自動增益控制(automatic gain

control, AGC)模擬神經元的調節作用。

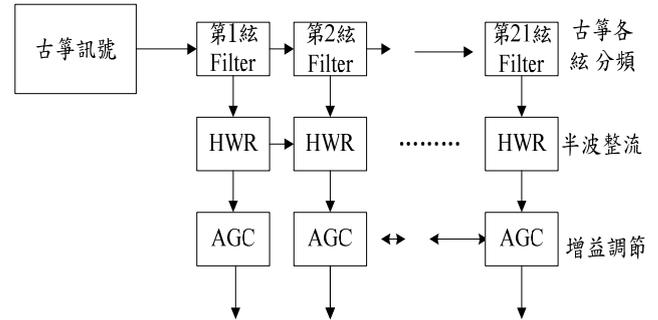


圖 9 使用 Slaney 的方法偵測古箏音高方統架構圖

### (三)古箏基頻偵測範例

圖 10 為一實際古箏波形範例，使用 Slaney 的內耳模型計算程式得到的內耳分析圖(cochleagram)如圖 11 所示，代表各個時間各個聽覺神經元的訊號。將內耳分析圖進行自相關分析得到圖 12 之自相關分析圖，代表古箏分析樣本在各個時間點對聽覺神經元的輸出反應。最後用 Slaney 的音高偵測方法得到的音高如圖 13 所示。

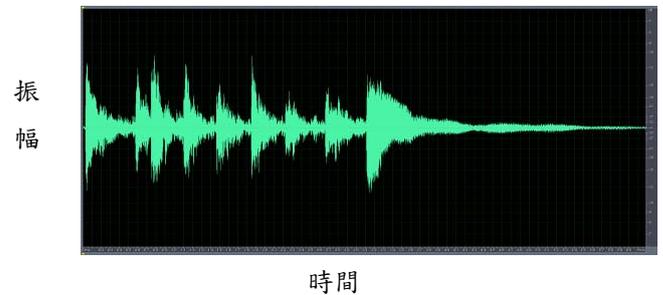


圖 10 古箏左手技法波形圖

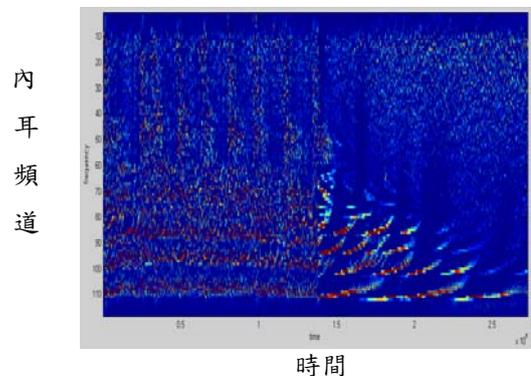
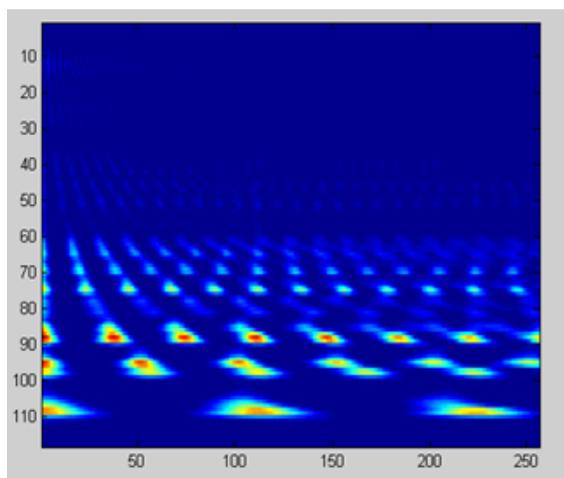


圖 11 古箏樣本內耳分析圖

內耳頻道



自相關延遲時間(音框)

圖 12 古箏樣本聽覺神經元自相關分析

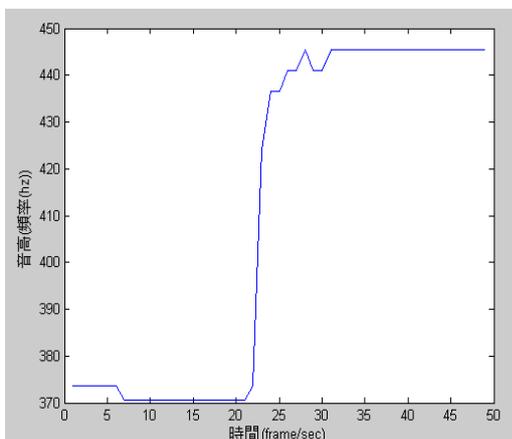


圖 13 使用 Slaney 的音高偵測方法的古箏上滑音樣本之音高分析結果

### 五、滑音識別

#### (一) 滑音識別模型

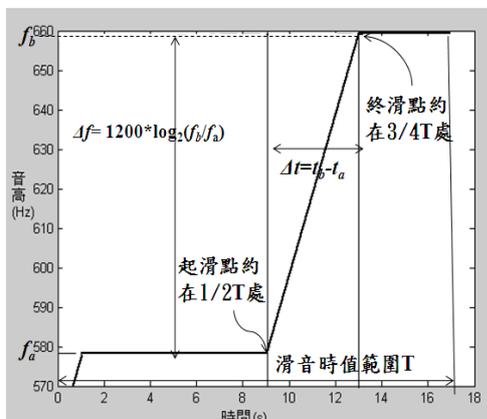


圖 14 滑音判別模型

由於古箏技法甚多，本研究主要以自動偵測箏樂技法中，上滑音與下滑音為研究目標。基本滑音判別架構如圖 14 所示。如前所述滑音變化之音程為大二度或小三度；而起滑點大約在滑音總時值  $T$  之  $1/2$  處，終滑點則約在總時值  $T$  之  $3/4$  處。

#### (二) 滑音識別方法

步驟一：偵測起音點(onset)與終音點(offset)

首先，先進行起音點與終音點偵測，找出各樂音的起始點  $p_{onset}$  與終音點  $p_{offset}$ ，如圖 15 所示。

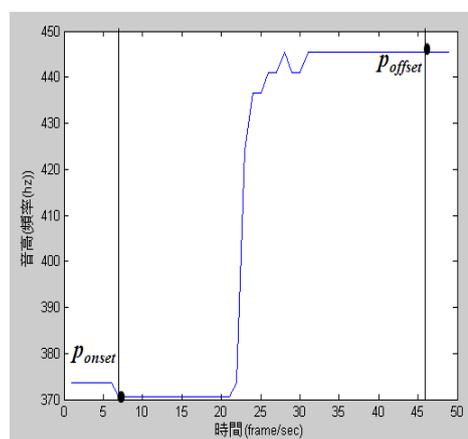


圖 15 起音點與終音點示意圖

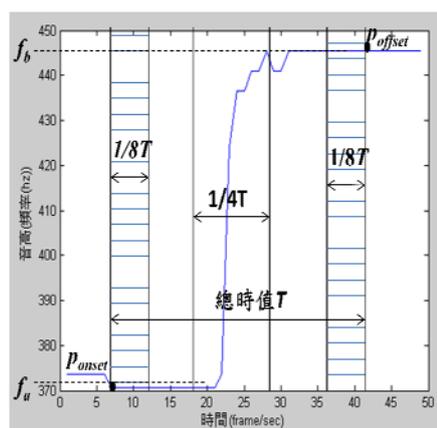


圖 16  $f_a$  與  $f_b$  計算來源示意圖

步驟二：求前  $f_a$ 、 $f_b$ (前  $1/8T$  與後  $1/8T$ ) 頻率值

一個完整滑音彈奏中，滑音變化區段約佔整個訊號的  $1/4$ (約位於整個滑音技法的  $2/4 \sim 3/4$  區

塊間),並且在這 1/4 的區段內會產生頻率變化。因此,我們可計算滑音總時值的前 1/8 穩定頻率區段之平均頻率值作為起滑點頻率  $f_a$ ;滑音總時值的後 1/8 穩定頻率區段之平均頻率值作為終滑點頻率  $f_b$ ,如圖 16 所示。

步驟三:  $f_a$ 、 $f_b$  頻率穩定判斷

由於  $f_a$ 、 $f_b$  對於滑音點偵測影響相當大,所以必須要判斷  $f_a$ 、 $f_b$  是否為穩定值。如頻率變化太大,即判斷為非滑音,並將其去除。

步驟四: 追蹤  $f_a$ 、 $f_b$  與偵測  $p_a$ 、 $p_b$

在進行辨識滑音之前,必須追蹤起滑點頻率  $f_a$ 、終滑點頻率  $f_b$ ,以作為計算  $\Delta f$  所用。接著偵測起滑點  $p_a$  與終滑點  $p_b$  發生之時間點  $t_a$  與  $t_b$ ,以作為計算  $\Delta t$  所用。

步驟五: 求滑音變化區段之時間間隔

取出  $t_b$  與  $t_a$  進行相減,可算出滑音變化區段中的時間長度  $\Delta t$ 。

$$\Delta t = t_b - t_a \quad (9)$$

其中  $\Delta t$  為滑音變化區段時間間隔,如圖 17 所示。

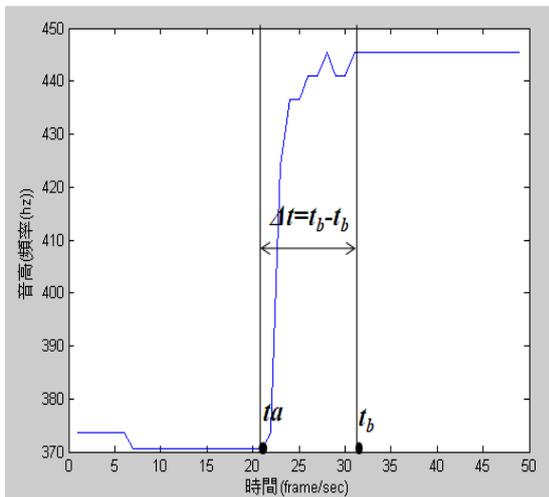


圖 17 滑音變化區段時間長度示意圖

步驟六: 求滑音變化區段之音程

利用  $f_b$ 、 $f_a$  進行音程計算,可求得滑音變化區段音程差異

$$\Delta f = 1200 * \log_2(f_b/f_a) \quad (10)$$

其中  $\Delta f$  為滑音變化區段之音程,如圖 18 所示。

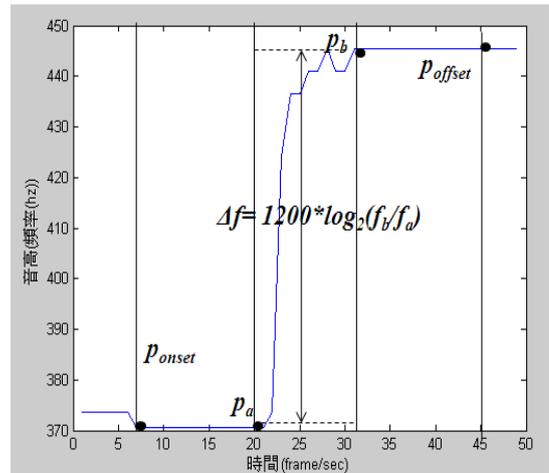


圖 18 滑音音程  $\Delta f$  示意圖

步驟七: 檢驗滑音之音程與時間間隔是否合於滑音限制

$\Delta f$  與  $\Delta t$  是辨識滑音主要特徵參數,我們必需檢查  $\Delta t$  的長度是否滿足時間間隔限制條件:

$$1/8 \leq \Delta t/T \leq 1/2 \quad (11)$$

而  $\Delta f$  的音程限制條件為:

$$|\Delta f - 200| \leq F_{\text{error}} \quad (12)$$

$$\text{或 } |\Delta f - 300| \leq F_{\text{error}} \quad (13)$$

其中  $\Delta f - 200$  與  $\Delta f - 300$  為計算大二度與小三度兩種滑音可能音程之音分差;  $F_{\text{error}}$  為可允許之音分誤差範圍,系統預設值為 20 音分。

(三) 滑音識別演算法

滑音識別之正式演算法詳細敘述於 Algorithm 1。

## 六、實驗

(一) 實驗環境設定

電腦設備: Intel® Core™2 Quad CPU Q6600 @2.40GHz, 2.40GHz, 1.99GB RAM。

實驗樣本: 古箏 24 弦之滑音彈奏樣本、箏曲彈奏集「滑音練習一」、箏樂名曲「出水蓮」等。

(二) 實驗一: 箏樂起音點偵測實驗

實驗樣本：



圖 19 古箏曲「出水蓮」前兩樂句

實驗方法：

步驟一：在相同環境變數下，讓 10 位不同的受測者聽取相同的待測樂句。

步驟二：讓受測者以人工標記樂句中各個樂音的起音點位置。

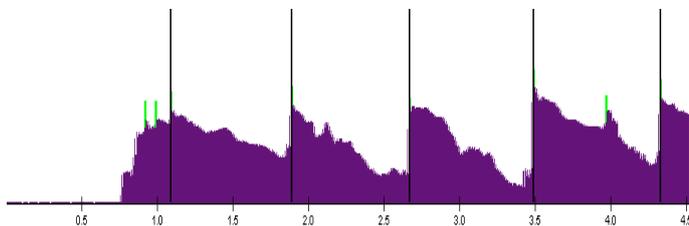
步驟三：步驟一與步驟二每人重覆做五次，而每次誤差值不可高過於 0.01 秒。

步驟四：將取出來的五個人工標示值取平均，作為此受測者的起音點人工辨識結果。

步驟五：再對人工標記起音點與 BeatRoot 程式偵測起音點結果做比較，分析誤差值變化。

實驗結果：

BeatRoot 斷音



人工斷音



圖 20 人工與 BeatRoot 起音點識別(前五樂音)

由表 1 可以看出人工斷音標準差較大的只有樂音 3，此例主要是因為其中有一位人工斷音人員，對於此樂音，所斷出來的時間，與其他人工斷音人員所斷出來的結果不相同(漏斷)。其餘樂音人工斷音標準差在 0.05 秒以下，可以根據

此統計值來設定程式斷音正確參考標準。而在樂音 6 與樂音 9 時，主要是受到古箏彈奏中的餘音干擾較強，使得 BeatRoot 所產生之斷音點誤差值較大。

表 1 出水蓮前十個音以人工起音點識別與 BeatRoot 起音點識別誤差值

樂音	人工斷音點(秒)	人工斷音標準差(秒)	BeatRoot 斷音點(秒)	BeatRoot 與人工之誤差(秒)
樂音 1	1.04	0.021	1.08	0.04
樂音 2	2.04	0.048	1.95	0.09
樂音 3	2.65	0.253	2.68	0.03
樂音 4	3.42	0.021	3.48	0.06
樂音 5	4.17	0.035	4.29	0.12
樂音 6	6.03	0.023	5.79	0.24
樂音 7	7.08	0.019	6.89	0.19
樂音 8	8.06	0.026	7.88	0.18
樂音 9	8.25	0.028	8.5	0.25
樂音 10	9.14	0.015	9.19	0.05
平均	—	0.048	—	0.16

(三) 實驗二：箏樂基頻偵測與滑音判別實驗

實驗樣本：

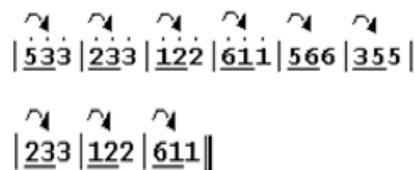


圖 21 箏曲彈奏集「滑音練習一」範例

實驗方法：

由於在箏樂滑音技法中，滑音變化區段約佔整個樂音時值部份的 1/4，然而這 1/4 的範圍，大致落在整個樂音起音點至終音點時值的 2/4 時值點與 3/4 時值點之間，而  $t_a$ 、 $t_b$  即是 2/4 時值與 3/4 時值之時間點。 $f_a$ 、 $f_b$  分別介於起音點至 2/4 時值點與 3/4 時值點至終音點， $f_a$ 、 $f_b$  我們就必需由這範圍中取得，再利用音程計算，計算出  $\Delta f$ 。

實驗結果：

表 2 起滑點頻率  $f_a$  與起滑點頻率  $f_b$  實驗結果

樂音	起滑點頻率 $f_a$ (Hz)	終滑點頻率 $f_b$ (Hz)	$\Delta f$ (音分)	理論值 (音分)	誤差 (音分)
<u>53</u>	584.7	658.2	206	200	6
<u>23</u>	490.5	584.7	305	300	5
<u>12</u>	438.4	490.5	193	200	7
<u>61</u>	348.98	438.4	399	300	99
<u>56</u>	327.3	348.98	111	200	89
<u>35</u>	294.3	327.3	193	200	7
<u>23</u>	245.0	294.3	317	300	17
<u>12</u>	218.1	245.0	202	200	2
<u>61</u>	474.7	218.14	-1347	300	-1053

從表 2 的實驗結果中，可以看出大部份都在誤差值的允可範圍內。而第三與第四個樂音，由於訊號之頻率較不穩定，使結果出現誤差較大；而在第九個樂音中，可以發現這個樂音誤差值極大，原因是來自基頻偵測中，由於對太低頻的樂音頻率解析度不足無法正確偵測基頻所造成。

表 3 各滑音  $p_{onset}$ 、 $p_{offset}$ 、 $p_a$ 、 $p_b$  發生時間點

樂音	$p_{onset}$ (音框)	$p_{offset}$ (音框)	$p_a$ (音框)	$p_b$ (音框)	$\frac{\Delta t}{T}$
<u>53</u>	3	53	28	41	0.26
<u>23</u>	53	159	106	131	0.23
<u>12</u>	212	266	239	253	0.25
<u>61</u>	266	363	316	321	0.05
<u>56</u>	363	464	413	438	0.24
<u>35</u>	464	518	491	504	0.24
<u>23</u>	583	632	608	620	0.24
<u>12</u>	632	738	682	697	0.14
<u>61</u>	738	793	770	782	0.22

由表 2 與表 3 之結果，我們進一步檢查  $\Delta f$  與  $\Delta t$  是否滿足滑音限制條件。如前所述，在  $\Delta f$  限制條件部份，我們將挑選音分差小於 20 音分的結果；在  $\Delta t$  限制條件的部份，將挑選  $1/8 \leq \Delta t/T \leq$

$1/2$  範圍之樂音。實驗結果，有六個滑音符合滑音限制條件被正確偵測出，但也有三個受到不同因素，而無法偵測出，分別是第 4、5、9 樂音。在第 4 與第 5 樂音中，所出現的誤差來源，來自訊號頻率的不穩定；而第 9 樂音是由於自相關分析受到汎音及前一音殘響干擾，出現基頻偵測錯誤的問題。而針對出水蓮前二樂句進行滑音偵測的結果，在這兩樂句中，共有七個滑音，偵測結果可正確抓取出五個滑音點，但無法偵測出的另外兩個滑音點，主要原因也是因為偵測起音點錯誤所造成，所以未來需對起音點偵測方法繼續進行調整，以提高滑音識別整體準確率。

## 七、結論

在偵測古箏滑音技法方面，最大的問題就在於整個箏樂中，上一個樂音之餘音，會伴隨在下一個樂音之中，形成餘音繚繞現象。由於古箏餘音的持續時間相當長，使得下一個樂音的起音點常常難以判斷，若改用多重基頻偵測技術應可改善此一問題。

未來的研究方面，我們大致上有二個目標：一、使用大量經典箏樂曲目實驗。箏曲裡，技法的運用會因不同派別也有所不同，所以我們希望可以針不同派別且經典之箏曲也進行人工與自動滑音識別實驗。二、其他表韻技法偵測：古箏技法相當多，即使同一首箏曲也有各種不同流派的彈奏方法，我們必需對更多的技法進行偵測方式之探討，朝向完整自動的內涵分析目標前進。

## 八、誌謝

本論文研究為國科會補助之研究成果，計畫編號 NSC 98-2221-E-216 -030。

## 九、參考文獻

- [1] 張儷瓊，“現代古箏演奏技法符號之探討”，*藝術學報*，第 69 期，pp. 39-52，2001.
- [2] 張儷瓊，“客家箏曲《出水蓮》乙凡音之按顫

- 手法模式分析一個時距定量的測音方法與附加頻率落點分析之應用研究”，*藝術學報*，第 80 期，pp. 175-194，2007.
- [3] 葉娟初，“論台灣現代箏樂作品演奏技法與記譜之傳統與變遷”，*國立台灣師範大學民族音樂研究所藝術學域領研究生論文發表會*，pp.2-1 - 2-16，2004.
- [4] 翟源，“試析箏的彈奏技法與聲音品質的關係”，*西安音樂學院碩士學位論文*，2006.
- [5] 雷華，“說說古箏基本彈奏法-夾彈、提彈”，*音樂天地*，第三期，pp. 51-52，2007.
- [6] 焦力，“論古箏的滑音”，*四川音院學系民樂系論文*，pp. 59-64，2007.
- [7] 林雅琇，“潮州箏樂素材在當代箏樂創作中之運用”，*演奏與詮釋研究生學術研討會論文集*，第二屆，2008.
- [8] 林雅琇，“潮州箏曲《寒鴉戲水》之版本研究”，*演奏與詮釋研究生學術研討會論文集*，第二屆，2008.
- [9] 林東河、黃好吟，“*箏曲彈奏集(第一冊)*”，*學藝出版社*，1990.
- [10] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, “A Tutorial on Onset Detection in Music Signals,” *IEEE Trans. on Speech and Audio Processing*, 2005.
- [11] C. Duxbury, J. P. Bello, M. Davies and M. Sandler, “Complex Domain Onset Detection For Musical Signals,” *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK, September, 2003.
- [12] S. Dixon, “Onset Detection Revisited,” *SimonProc. of the 9th Int. Conference on Digital Audio Effects (DAFx'06)*, Canada, September 18-20, 2006.
- [13] S. Dixon, “Simple Spectrum-Based Onset Detection,” *Music Information Retrieval Evaluation eXchange - MIREX*, 2006.
- [14] S. Dixon, “Simple Spectrum-Based Onset Detection,” *Music Information Retrieval Evaluation eXchange - MIREX*, 2006.
- [15] S. Dixon, “Learning to Detect Onsets of Acoustic Piano Tones,” *Journal of New Music Research*, Vol.30, No.1, pp 39-58, 2001.
- [16] M. Goto and Y. Muraoka, “Beat Tracking based on Multiple-agent Architecture A Real-time Beat Tracking System for Audio Signals,” *Proceedings of the Second International Conference on Multiagent Systems*, 1996.
- [17] M. Goto, “A Chorus-section Detection Method For Musical Audio Signals,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.V-437-440, April, 2003.
- [18] M. Goto and Y. Muraoka, “A Beat Tracking System for Acoustic Signals of Music,” *Proceedings of the second ACM international conference on Multimedia*, pp.365-372, 1993.
- [19] M. Goto, “A Predominant-F0 Estimation Method for Polyphonic Musical Audio Signals,” *Proceedings of the 18th International Congress on Acoustics*, pp.II-1085-1088, 2004.
- [20] M. Goto, “A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals,” *Speech Communication*, Vol.43, Issue 4, 43 pp. 311–329, 2004.
- [21] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps,” *EURASIP Journal on Advances in Signal Processing*, 2007.
- [22] A. P. Klapuri, A. J. Eronen, and J. T. Astola, “Analysis of the Meter of Acoustic Musical Signals,” *IEEE Transactions on Audio, Speech, and Language*, Vol. 14, No. 1, January 2006.
- [23] J.C.R. Licklider, “A Duplex Theory of Pitch Perception,” *Experientia*, 7, pp. 128-133, 1951.
- [24] R. F. Lyon, “A Computational Model of Filtering, Detection, and Compression in the Cochlea,” in *Proceedings of IEEE ICASSP*, pp. 1282-1285, 1982.
- [25] M. Slaney, “Lyon's Cochlear Model,” *Techn. Rep. # 13*, Apple Computer Inc., 1988.
- [26] M. Slaney, “Auditory Toolbox,” Version 2, *Technical Report No: 1998-010*, Internal Research Corporation, 1988.
- [27] M. Slaney and R.F. Lyon, “A Perceptual Pitch Detector,” in *Proceedings of IEEE ICASSP*, pp. 357-360, 1990.
- [28] T. Tolonen, “A Computationally Efficient Multipitch Analysis Model,” *IEEE Transactions on Speech And Audio Processing*, Vol. 8, No.6, November 2000.

**Algorithm 1 : Detect\_Sliding\_Tone( $S(t)$ )**

```
/*Input: 古箏箏曲原始波形訊號  $S(t)$  */
/*Output:  $S(t)$  中所有滑音之集合  $P$  , 其中每一滑音包含起音點
時間( $t_{onset}$ )、終音點時間點( $t_{offset}$ )、起滑點時間點( $t_a$ )
、終滑點時間點( $t_b$ )、起滑點頻率( $f_a$ )、終滑點頻率( $f_b$ ) */
0: begin
1:  $P_{onset} = \text{Onset\_Detect}(S(t))$ ; //  $P_{onset}$  為起音點之集合
2:  $P_{offset} = \text{Offset\_Detect}(S(t))$ ; //  $P_{offset}$  為終音點之集合
3:  $P = \emptyset$ ; // 起始  $S(t)$  中所有滑音之集合  $P$ 
4: for each tone ( $p_{onset}, p_{offset}$ ) in  $P_{onset}$  and  $P_{offset}$ 
5: // 其中  $p_{onset} \in P_{onset}$ ,  $p_{offset} \in P_{offset}$ 
6: begin
7:  $f_a = \text{AvgFreq}(S(t), p_{onset}, 1/8)$ ; // 計算滑音前 1/8 時值平均頻率
8:  $f_b = \text{AvgFreq}(S(t), p_{offset}, 1/8)$ ; // 計算滑音後 1/8 時值平均頻率
9:  $p_a = \text{Freq\_Trace}(p_{onset}, f_a)$ ; // 由起音點向後依照  $f_a$  頻率追蹤
10: // 起滑點  $p_a$ 
11:  $p_b = \text{Freq\_Trace}(p_{offset}, f_b)$ ; // 由終音點向前依照  $f_b$  頻率追蹤
12: // 起滑點  $p_b$ 
13:  $t_a = p_{onset} - p_a$  // 起滑點位置之時間
14:  $t_b = p_{offset} - p_b$  // 終滑點位置之時間
15:  $\Delta f = 1200 * \log_2(f_b/f_a)$  // 滑音音程計算
16:  $\Delta t = t_b - t_a$  // 滑音時值計算
17: if Constraint_Check( $\Delta f$ ,  $\Delta t$ ) = Up
18: // 根據音程  $\Delta f$  與時值  $\Delta t$  檢查是
19: // 否滿足上滑音條件限制
20: add( $Up, t_{onset}, t_{offset}, t_a, t_b, f_a, f_b$ ) to  $P$ 
21: // 增加一個上滑音資訊至  $P$  集合
22: else if Constraint( $\Delta f$ ,  $\Delta t$ ) = Down
23: // 根據音程  $\Delta f$  與時值  $\Delta t$  檢查是
24: // 否滿足上滑音條件限制
25: add( $Down, t_{onset}, t_{offset}, t_a, t_b, f_a, f_b$ ) to  $P$ 
26: // 增加一個下滑音資訊至  $P$  集合
27: end
28: return  $P$  //  $S(t)$  中所有滑音之集合  $P$ 
29: end
```