|  |  |  |
| --- | --- | --- |
|  | **以即時音量偵測實作節奏訓練** |  |
| **學生姓名：李東岳 指導老師：吳坤熹 教授** | | |

1. 簡介

在學習吉他的過程中，常常遇到抓不到節奏的問題，讓我想自己實作一個訓練程式。

此程式會在畫面上顯示樂譜、偵測使用者的輸入並判斷是否打在拍點上。除了預設的樂譜外，使用者還能匯入自製樂譜。

1. 系統技術
2. 即時音量偵測 (Real-time volume detection)

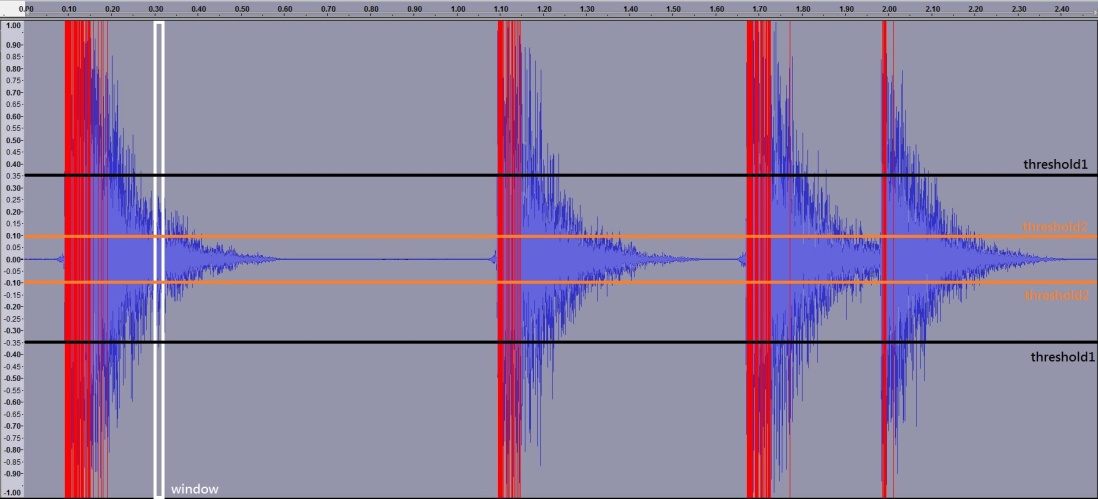
聲音經由麥克風接收後，經過PCM(Pulse-code modulation)將類比訊號數位化，而得到一串由數字組成的一維陣列。在陣列中取一個時間長度為t (t<30ms)的window(見圖一白框)，並將window中樣本S以下列算式計算，即可獲得該時間點的平均音量。

1. 起音點偵測 (Onset detection)

當一個聲音輸入時，在起始位置的能量會最強，之後慢慢衰退。以下面這段節奏為例：

音符.jpg

圖一為使用者按照這段節奏拍手所得到的訊號，其中紅色的部分為起音點。當音量超過threshold1(圖一黑線)時，判定為有效輸入的開始；當音量衰退到threshold2(圖一橘線)之下時，判定為有效輸入結束。一次循環代表一個有效的輸入。



(圖一)訊號分析

1. 實作內容

程式會開啟兩個thread，一個控制聲音輸入與運算，另一個控制畫面與音符指標。

當一個聲音輸入時，程式會判斷輸入是否有效。若為有效的輸入，程式會計算從按下開始到本次輸入的時間，與當前指標指向的音符時間做比較後，將結果輸出到螢幕上。(見圖二)

在控制畫面thread中，程式以八分音符作為一個單位，根據BPM(Beats per minute)計算每個單位的時間長度，並作為timer的長度。

按下開始後，程式會計算當前指標所指向的音符長度與八分音符長度比值，並放入counter。每當timer到了的時候，程式會把scan bar移動一格，並將counter減一。當counter歸零時，程式會將指標指向下個音符。

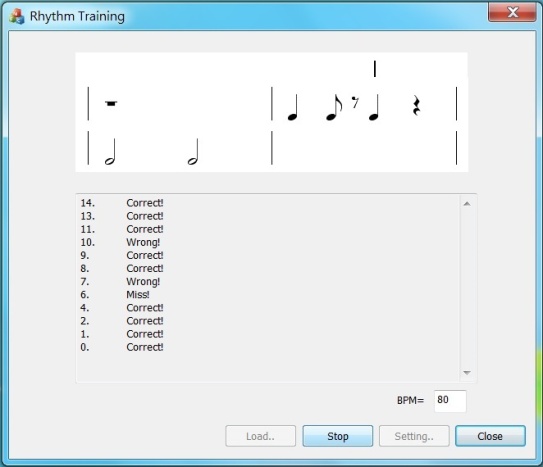
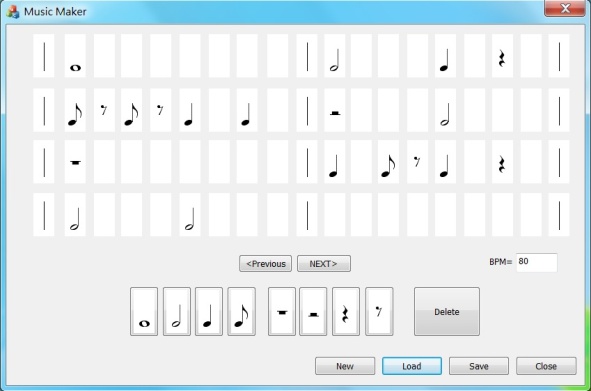
(圖二)聲音訊號處理流程

1. 實作成果

使用者可以使用樂譜編輯程式(見圖三)製作或修改樂譜，透過畫面下方的按鈕新增或刪除音符，編輯完成時可以點擊Save按鈕存檔。

開啟節奏訓練程式(見圖四)，使用者可以將剛編輯好的樂譜載入，設定BPM後按下Start開始訓練。使用者必須根據畫面上方的樂譜與掃描棒打出節奏，程式會將結果即時輸出到畫面中央。

若使用者發現麥克風收音太敏感或收不到，可以點選Setting設定音量門檻(Volume threshold)。



(圖三)製作樂譜 (圖四)節奏訓練

1. 問題討論與未來展望

目前程式是以音量作為有效輸入的判定依據，但環境噪音的音量超過一定值時，就會造成假警報(False alarm)。未來嘗試加入即時音高偵測(Real-time pitch detection)，分析環境噪音的頻率並過濾掉。

1. 參考資料
   1. 程式碼：
      1. Low level Digital Audio API：

<http://home.roadrunner.com/~jgglatt/tech/lowaud.htm>

* + 1. Reading From Buffer In WaveIn(C++)：

<http://www.dreamincode.net/forums/topic/292766-reading-from-buffer-in-wavein-c/>

* 1. 文章：
     1. Onset(Audio) – Wiki：

<http://en.wikipedia.org/wiki/Onset_(audio)>