音色認知與合成　NSC 90-2213-E-002-092

執行進度報告　　90/08/01~91/07/31

計畫主持人 劉長遠　國立台灣大學　資訊工程學系

本研究依計畫書所訂進度執行，分成三個部分同時進行.依進度已完成的部分如後. 第一個部分是歌聲分析部分，第二個部分是樂譜分析，第三個部分是樂器聲音分析，此三部份具爲音色認知與合成所需. 先述第一部份歌聲分析，針對歌聲中所蘊藏的音色內涵，建立一個聲音平台，在此平台上，可以計算且展示出各個模式參數.與聲音內涵，此模式爲主持人提出，用以合成與剝離音色特徵，此平台構造如下.

建構一個視覺化的歌聲分析系統，系統的基本功能中，從最初始的藉由輸入一段聲音訊號，得到能量頻譜圖，進而顯示更符合人類聽覺系統的 melscale 向量圖，同時也可以將這些圖，以灰階、反灰階及彩色的方式顯示，很輕易地藉由眼睛的觀察就可以得到對聲音初步認識，看出聲音訊號的整體大致特徵。更進一步地，在系統中，提供 sonograph 功能，可以用來檢視同一段聲音合成組與對照組的差異，輕易地分辨各種模式的差異性。

系統中以 SOM(Self-Organizing Map)類神經網路模型爲基礎，建立了聲音的在頻域(frequency domain)訊號特徵抽取功能和時域(time domain)訊號的雜訊分離及特徵模擬。時域功能中，直接利用 MAP 原理(Maximum a *posteriori* criteria)於類神經網路，藉由公式

$$\mathbf{c} = \arg\max_{\mathbf{i}\in N}\{p(\mathbf{Wi} \mid \mathbf{x_s})\}$$

尋找最佳的神經元 **c**，神經元據此調整其對訊號的轉換函數 (transfer function)，並依此函數提供兩種功能，將聲音檔剝離其原始聲音特徵和雜訊的功能，並能由此產生一組新的濾波器庫(filter bank)，可以運用於未知的聲音訊號。並能直接由系統播放 聽到處理完成的各種聲音訊號。

頻域系統中，除了能夠顯示所有輸入訊號的前置處理結果和利用一維 SOM 神經元陣列，進行特徵抽取與排列，還可以將特徵模存檔，再利用所有存檔過的特徵模資料庫，進行對未知訊號的特徵比對，同時利用「音序拓樸分析」進行對聲音時序扭曲(time warping)的修正，達到良好的音色特徵辨識的效果，而在特徵比對的子系統中，直接顯示未知訊號的所有特徵模變化情形和庫存特徵模相似度的圖像資訊和數據，直接利用眼睛觀察，就能得到聲音資料在各時間和特徵模交互作用的狀態。

第二個部分是樂譜分析 此部份主要在設計自動機能夠產生或逼近已知樂譜我們製作了一個數學系統能夠產生節奏其原理與製作如下

音樂的三大要素是旋律(melody)，節奏(rhythm)與和聲(harmony)。其中，節奏乃

是決定音樂性格最基礎且最重要的因素。所謂節奏，是指在一個單位時間中，進行各種長短音符的組合，因而產生各種樂曲的感覺與特色。由於一個單位時間，如一小節，或是二分音符，我們通常可將它細分成二或三個子單位(sub unit)，之後又可將這些子單位在細分成二或三個更小的單位，來形成各種不同的歷時序列，因此，節奏可以想成是階層架構的描述。

我們提出將林德門結構應用在音樂節奏的處理與展現 而林德門結構是將音樂節奏的階層特徵編碼後再 用林德門結構特有的圖形表現方式，爲不同的節奏，畫出獨一無二的”節奏樹’。本計畫將林德門結構用在解析音樂上爲世界上首先提出

因此，將一首樂譜的節奏編譯成林德門結構，而此結構就好比是這首樂譜的 Bar Code，包含了每小節節奏的微觀架構，並且可以將整首樂譜的宏觀結構用類似樹的分枝般地畫出來，同時展現整首曲子巨觀與微觀的特徵。

根據 Longuet-Higgins 的理論，我們只考慮標準的音符單位。即每個音符只能被分割成二或三個子單位。通常大部分典型的樂曲都符合這個要求。而有些特別的音符，如連音符（五連音，七連音等），由於其最小單位無法用上述方法分割，因此我們將這些音獨立出來，放入一個節奏資料庫中(oracle)，在此資料庫裡存放各式各樣特殊的節奏，(譬如裝飾音中的漣音，琶音，震音，迴音等)，參照此例外在資料庫的 index.，在結構中做上記號，並用特殊的符號或顏色在樹的分枝上表現，那麼便能處理樂譜中這些精彩的例外，將它以不破壞樹的整體結構下表現出來，而又能抽出樂譜中所有奧妙的神韻部分。

利用結構，我們可以將流傳千古的經典之作，如巴哈，蕭邦，舒伯特，李斯特等作曲家的名曲，來 compile 出其獨一無二的標記階 層符號串，只汲取節奏的部分，畫成各有特色的節奏樹，運用林德門結構的數學結構將整首曲子沒入此數學結構中 由此數學推論知曉作曲家們作曲時的脈絡，將曲子的整體一覽無遺。除此之外，還能將這些美妙的節奏，抽出樂句組合，用電腦合成打擊樂器來演奏，爲大師的傑作呈現另一番風貌。我們甚至能先產生一棵美麗的樹，再轉換成節奏，並結合類神經網路的威力，做到自動作曲的功能。且應用在自動從音樂聲 中辨識樂譜 **節奏架構流程如下**

1) 讀入 MIDI 檔
2) 分析每小節音符結構，處理不在 metrical unit 上的音符，跨音符拍子(cross-bar note)以及切分音(syncopation)，拆成合理的連音(tied note)，並找出有無特殊的例外音符（如：五連音，琶音等），將整個音群獨立處理，之後在階 層符號串上記錄在資料庫中對應的的 index
3) 將每小節音符結構轉換成林德門結構的階 層符號串形式，從最小的音符兩個或三個 merge(視當時的拍子)，直到 merge 成一個 note,此時應該剛好是一個小節的長度
4) 得到每小節的階 層符號串，之後便可根據需要進行小節的 merge，形成 8 小節的”小樂句”，或是 16 小節的”大樂句”，parse 合併後的 B 階 層符號串畫出

樂譜的”節奏樹”。

第三個部分是樂器聲音分析 此部份以製作小提琴聲音爲主 計算方法如下

cluster.c 是以一堆資料: 小提琴弓對弦所造成的壓力(x 的第一維; input),拉弓的速度(x 的第二維; input), 以及所對應的小提琴聲音的頻率(y; output), 並且使用 EM algorithm 來訊練一個 cluster weighted model. 當 prediction error $<C_{yy}|\ x>$ 變化很小的時後, 我們便停止 train(在 cluster.c 裡 iteration number = 100). 我們再以原來的壓力和速度的資料(即 input-set x)代入此 model 產生預測的 output 頻率, 即<y| x>並且寫入檔案”freqtest”中. 之後我們只要將產生出的”freqtest”餵進程式 freq2wav.c 中,便能得到*.raw 檔案格式, 再利用 sox 把*.raw 轉成*.wav 就可得到小提琴的聲音檔了. 由於我們所控制的自由度(free degree)只有弓的壓力以及速度, 其它的自由度(如音色的變化)便被預設在讀入的頻率檔裡, 所以所產生的聲音檔裡僅能反應出同一音色的大小聲, 以及緊湊度(節拍).

所謂 cluster weighted model 即是將描述 input-output space 的機率密度 p(x, y) 擴展至 clusters(p(x, y)= $\sum_{m=1}^{M} p(y,x,c_m)$)再使用 EM algorithm 得到 input-output 空間在每一個 cluster 的機率分佈情況之後, 便可得到<y | x>

$$<y\ |\ x>=\frac{\sum_{m=1}^{M} f(x,\ \ _m)\,p(x\ |\ c_m)\,p(c_m)}{\sum_{m=1}^{M} p(x\ |\ c_m)\,p(c_m)}$$

我們可以將每一個 cluster 視爲一個 local model 因此我們只要找出每一個 model 的 參數 p(y| x, c_m), p(x| c_m), 以及 β_m 就可計算<y | x>, 其中 p(x| c_m)=N(μ_m, C_m), p(y| x, c_m)=N(f(x, β_m), $C_{y,\ m}$), N 表高斯分佈, f(x, β_m) 表每一個 cluster 的 output prediction. 如何得到這些參數呢, 在此我們使用 EM algorithm.

EM algorithm 分成兩個 step: expectation-step 和 maximum-step.

E-step: 計算每一個 cluster 在每一個 input-output-space 的機率

$$p(c_m\,|\,y,x)=p(y,x\,|\,c_m)/\,p(y,x)=p(y,x\,|\,c_m)\,p(c_m)/\sum_{l=1}^{M}p(y,x\,|\,c_l)\,p(c_l)$$

M-step: 利用統計學的 maximize likelihood function 的方法將 p(x, y)對 β_m 作偏微分, 得到 $_m = B_m^{-1}\bullet A_m$, 其中$[B_m]_{i,j}$=<$f_i(x,\beta_m)f_j(x,\ \beta_m)$>$_m$ and $[A_m]_{i,j}$=<$y_i f_j(x,\ \beta_m)$>$_m$.

再重新計算 cluster parameters: $p(c_m)=\frac{1}{N}\sum_{n=1}^{N}p(c_m\,|\,y_n,x_n)$

$$p(x\,|\,c_m)=\frac{1}{(2®)^{d/2}\left|c_m\right|^{1/2}}e^{-(x-«_m)c_m^{-1}(x-«_m)/2}\ ;\ p(y\,|\,x,c_m)=\frac{1}{(2®)^{d/2}\left|c_{ym}\right|^{1/2}}e^{-(f(x,\ \ )-y)c_{ym}^{-1}(f(x,\ \ )-y)/2}$$

我們得到每一個 cluster 的 weight, β_m 之後就能計算<y| x>