

能表達音樂特徵的人體動畫自動產生機制

Automatic Generation of Character Animations for Expressing Music Features

雷嘉駿
國立政治大學資訊科學系
g9506@cs.nccu.edu.tw

李蔡彥
國立政治大學資訊科學系
li@cs.nccu.edu.tw

摘要

近年來電腦計算能力的進步使得 3D 虛擬環境得到廣泛的應用。本研究希望能在虛擬環境中結合人體動畫和音樂的特色，以人體動畫來詮釋音樂。我們希望能設計一個智慧型的人體動作產生器，賦予虛擬人物表達音樂特徵的能力，讓動作會因為“聽到”不同的音樂而有所不同。基於人類聽覺的短暫性，系統會自動抓取音樂特徵後將音樂切割成多個片段、對每一片段獨立規劃動作並產生動畫。過去動畫與音樂相關的研究中，許多生成的動作都經由修改或重組運動資料庫中的動作。本研究分析音樂和動作之間的關係，使用程序式動畫產生法自動產生多變且適當的詮釋動作。實驗顯示本系統能通用於 LOA1 人體模型和 MIDI 音樂；此外，透過調整系統中的參數，我們能產生不同風格的動畫，以符合不同使用者偏好和不同音樂曲風的特色。

關鍵詞：虛擬環境、人體動畫、音樂特徵。

1. 前言

隨著硬體設備的進步和 3D 技術的增進，往日只能在大型主機上才能執行的虛擬實境，現今在一般桌上型電腦上也能負擔得起，虛擬實境也因此有了廣泛的應用，如科學模擬分析、遠距會議、飛行模擬等方面均有其應用。人們也將許多生活上的行為在虛擬實境中實現，如娛樂視覺效果、環境的導覽系統等。

虛擬實境是由電腦模擬出來的三度空間，使用者能透過虛擬人物 (virtual character) 作為化身 (avator) 進入虛擬世界中活動，與世界中的事物互動。為了加強虛擬實境的真實性，我們希望結合音樂，控制虛擬人物動作的方式詮釋音樂。

音樂的表達，最常見的是為音樂配搭舞步，舞步主要是表達音樂節奏。在過去文獻中 [9][11][12] 使用不同的方法試圖產生出新動作，由於舞步是預先經由學習，有規劃地以舞蹈方式表現音樂，所以以動作資料庫為基礎進行修改以產生新動作是合理的。但是，對沒有專業舞蹈訓練的聽眾而言，在沒有預先學習的情況下，乃會隨著音樂不自覺地擺動著身體，表現出聽到的音樂的特性。這些動作通常是個人的、主觀的和隨意的，這些主觀的音樂表

達除節奏外，旋律常常是吸引聽眾注意並用以表達的音樂內容。因此，根據音樂特性，在沒有動作資料庫下自動化產生使用者偏好的動作以詮釋音樂，是本文將探討的研究議題。

2. 相關研究

旋律是音樂中最容易被記得的部份。Delone [2] 認為任何旋律的必要元素包含音高、節奏和音色，但是音色的改變並不會影響旋律的確認。Delone 的實驗證明，同一段旋律由兩種不同音色的樂器演奏，實驗者能確認出來。

過去有很多研究關於人類是如何建立一個架構來記憶音樂。Dowling [3] 發現旋律的輪廓比旋律更容易記憶；旋律的輪廓可以由音符的音高起伏和節奏長短刻畫。因此兩個輪廓不相似的旋律很容易區別。Dowling 以實驗證明，一段耳熟能詳的旋律和一段輪廓相似但不完全相同的旋律，發現只要相對音程相似的情況下都能確認出來，這個實驗中只測試短期的 (short-term) 旋律。他們嘗試改變旋律的音階來增加確認的困難度，發現音階改變後只要輪廓一樣都能確認出來。此外，人類的即時記憶力是有限的，對於長期的 (long-term) 旋律，演奏下來，多數聽眾只會記得剛剛演奏過的音，更久之前的音往往會馬上忘記。

以人體動畫結合音樂的議題在過去已有不少成果。由動作和音樂之間的關係可分為以下三類：

第一類、根據動作產生音樂：Mishra and Hahn [7] 提出動作和聲音之間應該如真實世界中物體互相碰撞發出聲音般的直接關係，大多數電腦動畫中動作和聲音之間是分開處理的，所以希望在虛擬實境中因應動作即時產生聲音。Nakamura, Kaku, Noma and Yoshida [8] 同樣是使用動作產生背景音樂，兩者不同之處是前者是即時動態地產生音樂，系統強調動作特徵和音樂特徵的對應，而後者是需經人工處理後從資料庫中產生背景音樂。

第二類、音樂產生動作：Oore and Akiyama [9] 使用機器學習 (machine learning) 的方法，從大量的訓練資料中分析、訓練出音樂和動作之間的關係，其後使用者輸入一個新的音樂時，系統會根據過去的訓練結果產生手部動作來對應音樂。此外，Cardle, Barthe, Brooks and Robinson [1] 提出了同時

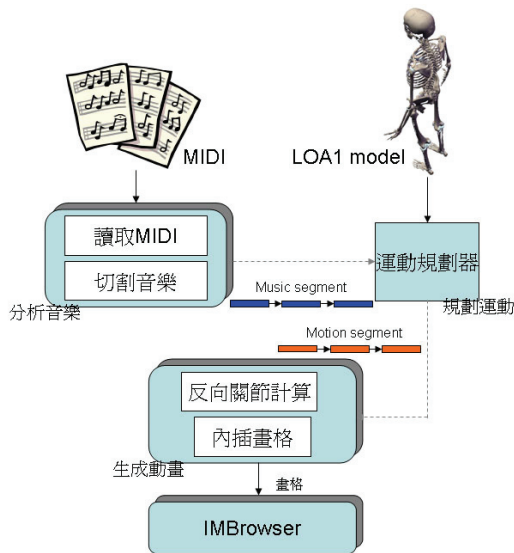


圖 1 系統架構

分析 MIDI 和 audio 兩種不同類型的音樂記錄方法，因為 audio 的好處是能精確地記錄聲音的資料，但是抓取和分析資料比較困難；使用 MIDI 的好處是方便電腦抓取資料，但這種表示法有資訊流失的問題；所以[1]希望綜合兩方優點，系統從 MIDI 和 audio 中擷取和分析音樂後，從資料庫中的動作修改組合出新的動作。

第三類、音樂和動作相互影響：Lee and Lee [5] 延續[1]的研究理論，提出了音樂圖（music graph）的概念。音樂圖相似於製作動畫中使用的動作圖（Motion Graph），圖中的節點記錄的是一段音樂而不是一段動作。經由輸入不對應的音樂和動作，系統修改音樂的時間比率（time scale）和動作的時間（timewrap）令兩者搭配起來。[11][12]以舞蹈為例子，結合 Laban 對舞蹈動作的分析和描述[4]，分別對動作和音樂進行分析、切割片段和重組片段來達到產生對應輸入的音樂的動作。不同的是，Shiratori, Nakazawa and Ikeuchi [11]在分析合成動作中，從動作資料庫中合成具有情感的特性的動作。Torresani, Hackney and Bregler[12]則是建構動作圖以找尋相似於原有動作的動作，進而串連成新的動作。

在過往的研究中，動作的生成都是從存在的動作中修改、調整或合成產生新的動作，但尚未有研究使用程序式動畫，在沒有動作存在的情況下，產生多變化的動作。故本論文將提出使用程序式動畫方式產生動作來表達音樂。

3. 系統總覽

圖 1 為本論文的系統架構，系統分為三大部份：第一部份是音樂的處理，系統接收 MIDI 音樂後讀取並分析音樂特徵將其切割成多個音樂片段（music segments）；第二部份是運動規劃，根據我們提出的音樂特徵模型為每一音樂片段規劃右

手、左手、身體和頭部動作，包括關鍵格的產生、關鍵格之間的軌跡和運動速率，記錄於動作片段（motion segments）中，包含各部位的動作和速率；第三部份是產生動畫，使用 ikan 模組[1]由反向機構學計算肢體的組態產生動作並內插產生動畫。這些動畫最後則是在我們實驗過去開發的 3D 引擎 IMBrowser[6]上呈現。

4. 考慮機構限制的旋律切割

本研究的目的主要是在為虛擬人物規劃手部運動以表達旋律。旋律可以由音高的高低起伏和節奏長短所形成的輪廓來表達，而我們將使用手部運動來視覺化旋律的輪廓。基於人類聽覺的記憶力是有限的，將音樂切割成多個片段後，每個片段各自規劃運動。此外，考慮到手腕的活動空間有一定的限制，手臂的各關節也有一些人體限制，因此我們在切割旋律時不能只考慮音樂因素，而應同時考慮手臂動作能否表達。

4.1 組態空間的定義

我們的系統假設由使用者指定 MIDI 中的兩個音軌作為旋律音軌和節奏音軌。為了方便表示將兩個音軌的資料合成一個組態，我們使用 $melody = [n_1, n_2, \dots, n_m]$ 定義音樂組態， $melody$ 由多個音符 n_i 所組成，每個音符 $n_i = [pitch, rhythm, intensity, isBeat]$ 有四個屬性，分別是音高（pitch）、音長（rhythm）、強度（intensity）和是否節奏點（isBeat）。音高的範圍從 0 到 127，單位是半音；音長的範圍從 0 到 4，單位是拍；強度的範圍從 0 到 120。 $melody$ 中的每一個音符 n_i 的音高、節奏和強度擷取於旋律音軌，而是否為節奏點屬性則需要從節奏音軌中的資料判斷出來。例如，如果在 n_i 演奏的同時節奏音軌中亦有音符演奏，則 n_i 為節奏點，否則為非節奏點。

4.2 旋律切割器的目標函數

在考慮音樂屬性和動作限制下，我們希望對於使用者能依個人偏好產生不同的切割。例如，使用者可以偏好較少的節奏點出現在一個片段中，或者是偏好較長的音長片段等。在實作上我們會將可做為偏好的條件參數化，依據使用者的喜好做適當的加權，以做為切割時的評分依據，進而切割出不同風格的片段。

目標函數（objective function）是在旋律切割演算法中對一段旋律片段進行評分的依據。一般而言，旋律片段的組態與使用者的偏好越相似，分數越低。我們令 $s = [n_1, n_2, \dots, n_m]$ 代表一段音樂片段， s 包含一個或多個音符（ $m > 0$ ），目標函數 f 是由四個部份所組成：

$$f_1(s) = isBeat(s.n_m) \quad (1)$$

$$f_2(s) = range(s) \quad (2)$$

Algorithm: Segment(melody)

```
1. Table T;  
2. while length from minimal to melody.length  
3. begin  
4.   while i from 0 to melody.length - length - 1  
5.     begin  
6.       while mid from i + 1 to i + length - 1  
7.         begin  
8.           value = normalize( T[i][mid], T[mid][i + length]);  
9.           if value <= T[i][i + length]  
10.            T[i][i + length].segmentAt = mid;  
11.            T[i][i + length] = value;  
12.         end  
13.     end  
14. end
```

圖 2 旋律切割演算法

$$f_3(s) = \text{rhythmDensity}(s) \quad (3)$$

$$f_4(s) = \text{totalRhythm}(s) \quad (4)$$

$$f(s) = w_1 * f_1(s) + w_2 * f_2(s) + w_3 * f_3(s) + w_4 * f_4(s) \quad (5)$$

f_1 是評估片段中最後一個音符是否為休止符、節奏點或非節奏點，若為休止符或節奏點則 $isBeat()$ 將回傳 0，反之則回傳 1，所以判斷最後一個音符將得到 1 或 0 分。

f_2 是評估片段中的音域大小。音域的大小是指片段 s 中最高的音高與最低的音高之差 $pitch_{range}$ 。在樂理中音高以 12 個半音為一個音域，因此我們以 12 個半音為音域的標準值，函數回傳片段中的音域與 12 之差 $abs(pitch_{range} - 12)$ 為分數。

f_3 是評估片段中的節奏點的密度，片段中節奏點的密度 $density_{local}$ 是指片段 s 中的音符數目與節奏點數目之比，我們以全域的節奏點的密度 $density_{global}$ 為標準值做比較，此函數回傳片段中的節奏點的密度與全域的節奏點的密度之差 $abs(density_{global} - density_{local})$ 為分數。

f_4 是評估片段中的總音長，總音長 $totalRhythm$ 由片段中的所有音符的長度加總所得，我們設定參數 v 作為總音長的偏好值，使用者可根據偏好條改 v 的值，函數回傳 $abs(totalRhythm - v)$ 為分數。

目標函數組合以上四個函數，每個函數會經過正規化到 0~1 之間，並乘上由使用者指定或系統預設的權重，最後得到目標函數的定義如(5)，此函數 f 之值範圍為 0~1 之間。

4.3 旋律切割演算法

圖 2 為根據使用者偏好和考慮動作限制下的旋律切割演算法的虛擬碼。演算法使用動態規劃方式 (dynamic programming) 計算，此演算法相似於文字編輯器的換行方法，找出一段文字中多個換行點換行使得文字盡可能排列整齊。演算法中第一行的 Table 是動態規劃中把已經計算過的值儲存起來，Table 會預先計算每個片段的目標函數的分數並記錄，其後開始進行搜尋的迴圈。第 2~7 行，我

們先從長度 (length) 為最小的片段開始計算 (我們最小長度為 2)，之後慢慢將長度增加到整個旋律的長度 (melody.length)，片段的起始位置由第二層迴圈控制，對於任何長度的片段的起始位置 (i) 從 0 開始到 melody.length - length，由以上兩個迴圈取出任何位置任何長度的片段後，由第三層迴圈嘗試找出切割點 (mid)，其值為 $i + 1$ 至 $i + length - 1$ 之間。第 8 行，切割出兩片段後進行正規化使目標值為 0~1 之間，以包含片段的數目和總片段數目之比正規化。第 9~11 行，判斷起始位置為 i 長度為 length 的片段是否以 mid 為切割點後得到更好的目標值，如果是則記錄切割點並更新目標值。

旋律切割演算法使用動態規劃的方式，目的是希望找出最佳解，我們演算法使用從下而上的方式 (bottom-up fashion)，對所有元素以 2 為長度開始評估，其後長度逐漸增長到總長度。演算法由三個迴圈構成，過程中每個片段的分數已預先計算被存在 Table 中，第一層迴圈控制長度，複雜度為 $O(n)$ ，第二層迴圈控制起始位置，複雜度為 $O(n)$ ，第三層迴圈控制切斷點位置，複雜度為 $O(n)$ ，所以整個演算法的複雜度為 $O(n^3)$ 。

演算法結束後 table 中將記錄任一片段是否有切割點，之後，我們從最大的片段開始 $T[0][melody.length]$ ，如果有切割點則切割成兩個片段並重覆此動作直到片段中沒有切割點為止。此演算法最後得到的是一連串有序的音樂片段 $music\ segment = s_1, s_2, \dots, s_k, k > 0$ 。

5. 規劃動作軌跡

5.1 音樂特徵模型

在我們對應音樂特徵到動作特徵前，我們先探討人體動作和音樂之間的關係。Laban 在這方面有非常深入的研究並提出 "labanotation"。Laban Movement Analysis(LMA)是一個描述語言，用來描述和記錄人體的舞蹈動作，共分成四個部份 body, effort, shape 及 space。

根據 Laban 的理論[4]，人體動作的情感和表達能夠由其中兩個元件—effort 和 shape 來描述。effort 是細緻地描述動作特徵令人理解動作的內在意圖，如憤怒下推擠別人和伸手觸碰別人，兩個動作同樣是伸展手臂的動作但有斷然不同的意圖。使用這四個種類我們能描述一個動作的力道、快慢、自由性和方向。

人類與生俱來就有舞蹈的能力，大多數人能夠搭配音樂的節奏性活動身體，例如，嘻哈舞蹈員 (Hip-hop dancers) 能夠在聽到音樂的同時做出各種高難度的動作搭配音樂，從觀察舞蹈員動作中，我們可歸納出兩點：

- 動作的節奏與音樂的節奏是同步的。
- 動作的強度與音樂的強度是同步的。

綜合以上知識，我們使用 LMA 中的 weight

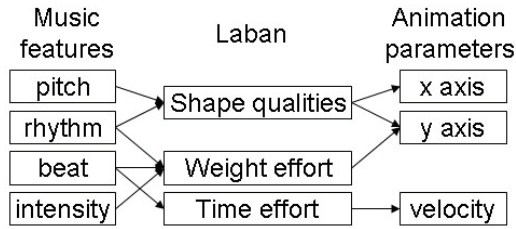


圖 3 音樂特徵模型

effort, time effort 和 shape qualities 三個元件，提出音樂特徵和動作特徵之間的關係的模型（圖 3）。由於音高和音長是刻畫旋律的輪廓，所以音高和音長對應 shape qualities，藉由手腕在 xy 空間上的位置變化來視覺化旋律。節奏的具體表現是動作的轉折，[10]指出動作會有加速減速然後停頓來表達節奏點，觀察舞蹈員的動作，會感覺動作的節奏與音樂的節奏同步的主要原因是舞蹈員在音樂的節奏點上大幅度改變動作方向和速率，較強的音用較大的幅度和力道，所以音長、是否節奏點和強度對應 weight effort，藉由軌跡的幅度和速度表達節奏和強度。此外，我們是否節奏點還對應 time effort，藉由加速減速產生停頓感來表現節奏點。

我們的運動規劃器使用這個模型規劃右手軌跡運動。規劃分成三個部份：第一個部份是根據音高和音長對應於空間位置作為關鍵格；第二個部份是根據節奏點設計關鍵格之間的運動軌跡；第三個部份是根據強度和節奏規劃運動速率。

5.2 關鍵格的產生

關鍵格的產生我們是以音符為單位，對每一個音符規劃右手手腕在空間的位置作為關鍵格（如圖 4）。x 軸方向對應時間，每更換一個音樂片段便轉換 x 軸方向；y 軸方向對應音高，以音樂片段中的中值音高（mean pitch）為標準，高於中值音高的音對應於 y 軸的正半軸，小於中值音高則對應於 y 軸的負半軸。計算的公式如(6)-(8)。旋律的第一個音符為特例，我們不規劃其空間的位置，以手腕的原位置 P_{init} 為起始位置，其後的每一個音符分別規劃 x, y 位置，x 軸的位置是以上一個關鍵格的位置 $P(n_{i-1}).x$ 加上這個音符的節奏所佔的空間 $x(n)$ ，而節奏所佔的空間的計算方法是以音符的節奏佔音樂片段的總節奏的比率乘上可利用的 x 軸空間。y 軸的位置則是以音符的音高與中值音高之差乘上 Y_{scale} ，而 Y_{scale} 可由使用者指定的偏好參數。

$$P(n_i) = \begin{cases} P_{init} & i = 0 \\ (P(n_{i-1}).x + x(n_i), y(n_i), P(n_{i-1}).z) & other \end{cases} \quad (6)$$

where

$$x(n) = n \cdot rhythm / segment_rhythm * X_{available} \quad (7)$$

$$y(n) = (n_i \cdot pitch - n_{mean} \cdot pitch) * Y_{scale} \quad (8)$$

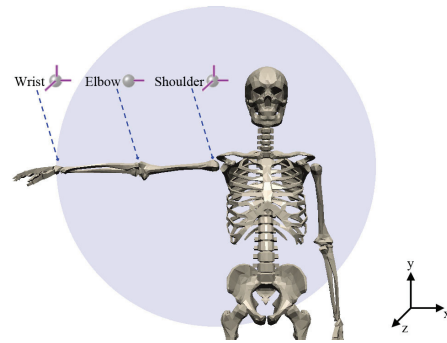


圖 4 手部關節可到達範圍和座標系統

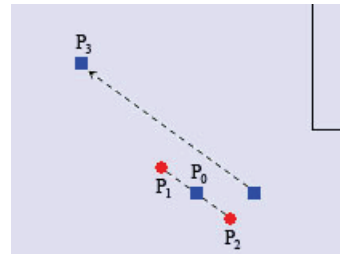


圖 5 非節奏點規劃的控制點。矩形的點為關鍵格對應的位置，圓形的點為貝茲曲線的內部控制點。

5.3 產生關鍵格之間的軌跡

我們使用三次貝茲曲線產生兩點之間的軌跡。兩個關鍵格的點分別是貝茲曲線的兩個端點 P_0 和 P_3 ，而另外的兩個內部的控制點 P_1, P_2 ，則是如我們提出的模型以音長、強度和是否節奏點規劃。音樂特徵會表現於通過關鍵格的軌跡，對每一個關鍵格我們是規劃相鄰的兩個控制點，即上一個貝茲曲線的 P_2 和下一個貝茲曲線的 P_1 ，如圖 5，非同一個曲線上的兩個控制點。以下我們將沿用 P_1 和 P_2 的表示方式。

5.3.1 非節奏點

如圖 5 所示，對於非節奏點我們令軌跡平滑地經過。所謂平滑是經過關鍵格的軌跡的切線方向是連續的。我們使用極座標來描述 P_1 和 P_2 的位置，以關鍵格為中心，從前一個關鍵格向後一個關鍵格的方向為 P_1 的方向， P_2 的方向為 P_1 的逆方向，我們使用參數 smooth 來決定 P_1 和 P_2 之間的距離，參數越大，距離越遠，則通過關鍵格的過程中切線方向的改變越小。

5.3.2 節奏點

若音符為節奏點時，則我們使用軌跡速度的改變來呈現節奏感。速度分為方向和速率，在 5.4 節我們將說明速率的規劃。在這一節中，我們將說明軌跡的長度和方向。軌跡的長度主要由 P_2 與關鍵格之間的夾角和距離控制，因此我們根據音符的音長、強度和參數，規劃 P_2 與關鍵格之間的夾角和距

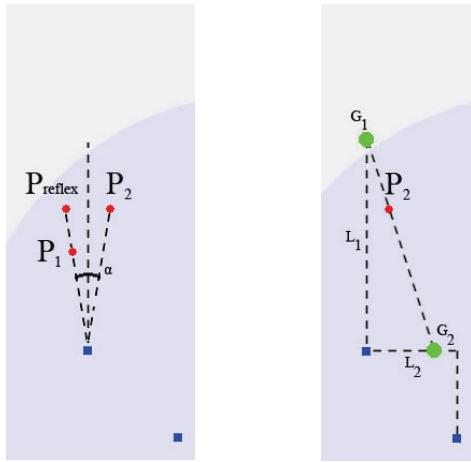


圖 6 P_1 的產生方法示意圖 圖 7 P_2 的產生方法示意圖

離。另外，我們知道 P_1 和 P_2 代表曲線的切線方向，其夾角 α （如圖 6 所示）越小則方向改變越大，所以我們使用鏡射的方式令 α 小於 180° 。

圖 7 是 P_2 產生方法的示意圖， G_1 及 G_2 分別對應到強度為 120 和 0 時的控制點，之間的強度值則在直線 G_1G_2 中線性內插求得，而 G_1, G_2 的位置是由兩個參數 L_1, L_2 所決定。 G_1 的位置在關鍵格的正上方，距離為 L_1 ；而 G_2 的位置在指向前一個關鍵格的水平方向上，距離 L_2 。除了強度外我們還需要考慮音長，例如，假設具相同強度的兩個節奏點，其中一個是一拍，一個是半拍，時間由音長決定，若令它們具相約的速率則必須控制軌跡的長度。所以我們以音長對 L_1 進行縮放來調整 P_2 的位置，從而調整貝茲曲線所描述的軌跡的長度。 P_1 的位置由 P_2 和參數 reflectance 決定。如圖 6 所示，以關鍵格的 x 座標相等的直線為對稱軸求得 P_{reflex} ，在 P_{reflex} 和關鍵格的連線上線性內插確定 P_1 ，reflectance 為 1 時 $P_1 = P_{reflex}$ ，reflectance 越小則越接近關鍵格。

5.4 產生關鍵格之間的速率函數

在動畫製作上，一個物體以先慢後快的動作進行內插的方法稱為緩入 (ease in)。而物體以先快後慢的動作進行內插的方法則稱為緩出 (ease out)。同樣地，我們以貝茲曲線來描述速率函數，橫座標表示音符的音長所佔的時間，縱座標表示曲線的內插值，透過調整控制點的位置，可改變曲線的曲度，以控制運動軌跡的內插值。我們系統中規劃的速率函數有三種。第一種是音符是節奏點，我們以類似圖 8(a) 的曲線規劃此類音符。此曲線一開始快速遞減然後平緩地進行，其後速度逐漸增加，最後瞬間降至零。這樣設計的目的在於節奏點出現之前產生夠大的速率，然後瞬間降速至靜止，藉由加速減速停頓產生節奏感。第二種是非節奏點且前一個音也是非節奏點，則我們以均速沿軌跡移動（如圖 8(b)）。第三種是非節奏點但前一個音是節奏點，則

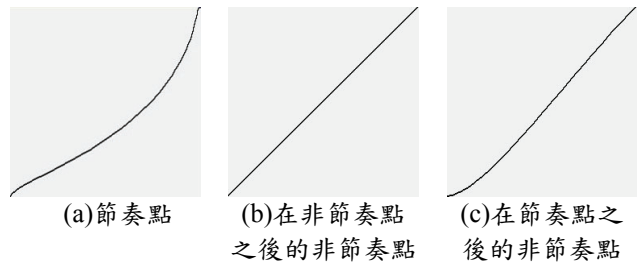


圖 8 三種規劃的速率函數

Algorithm: right hand planning(music segment)

1. P_{init} = current wrist location
2. **for each** segment in music
3. **begin**
4. plan key position for each note in segment
5. **if** position out of bound
6. scale y
7. P_{init} = the last key position
8. **end**
9. KeyPos = the first key position of second segment
10. **for each** segment in music
11. **begin**
12. plan trajectory between key positions
13. KeyPos = the first key position of next segment
14. **end**

圖 9 右手手腕軌跡規劃演算法

我們以圖 8(c) 的曲線作為速率函數。與第二種速率函數的差別在於曲線剛開始時以緩入的方式進行，因為前一個音是節奏點，所以起始處於靜止狀態，則我們以緩入的方式以免發生不自然的感覺。

5.5 手部動作軌跡規劃演算法

圖 9 為根據我們上述設計的右手手腕軌跡的規劃演算法。傳入的參數是多個音樂片段。第 1 行，我們準備一個 P_{init} 記錄每個片斷的起始位置，預設為現在手腕的位置。第 2~8 行，首先對每一個音樂片段規劃關鍵格，如果參數因為 Y_{scale} 太大出現關鍵格超過手腕能到達的範圍時，我們對 y 座標進行比例縮放 (scaling)，其後記錄最後一個關鍵格以做為下一個音樂片段的起始位置。在第 9 行。規劃非節奏點的軌跡時，我們需要前後兩個關鍵格。若音樂片段最後一個音為非節奏點時，我們需要下一個音樂片段的的第一個關鍵格。所以我們準備一個 KeyPos 記錄下一個音樂片段的的第一個關鍵格。在第 10~14 行中，我們對每個音樂片段規劃關鍵格之間的軌跡，並更新 KeyPos 的值。

5.6 身體其他各部份的動作軌跡規劃

5.6.1 左手動作軌跡規劃

規劃左手運動軌跡與右手相仿，使用音樂特徵模型規劃三個部份：第一個部份是關鍵格位置；第二個部份是關鍵格之間的運動軌跡；第三個部份是運動速率。

Algorithm: body planning(music segment)

```

1.  $P_{current} = \text{current v15 rotation}$ 
2.  $accumulate = 0$ 
3.  $direct = 1$ 
4. for each segment in music
5. begin
6.   for each note in segment
7.   begin
8.     if note is beat & note's rhythmValue isn't too short
9.        $plan(accumulate, P_{current})$ 
10.       $plan\ rotation\ for\ joint\ v15$ 
11.       $P_{current} = P_3$ 
12.       $accumulate = 0$ 
13.       $direct *= -1$ 
14.     else
15.        $accumulate += \text{note's rhythmValue}$ 
16.   end
17. end

```

圖 10 身體運動規劃演算法

左手運動的目的是彌補單手運動的不足和加強音樂特徵的表達。我們針對兩個音樂元素規劃左手運動：音長和斷奏 (*staccato*)。音樂中常常使用的音長是一拍或半拍，相對地，大於一拍的音符出現時會有延長的感覺出現，因此我們對大於一拍的音長規劃左手向外延伸的動作。其次，斷奏猶如弦樂器演奏中的連弓與斷弓，吹管樂器演奏中的連音與吐音，是鋼琴彈奏兩種基本的方法。斷奏是音符演奏後瞬間消音，聽覺上有突然切掉消失的感覺，所以我們規劃左手模仿鋼琴彈奏的舉起落下的動作，控制速率加速然後快速減速來產生停頓感。

5.6.2 身體和頭部運動規劃

規劃身體和頭部運動與手部動作的方式較不同，我們只是改變腰部關節 (*v15*) 和頭部關節 (*skullbase*) 的旋轉量，其位置不變，所以沒有軌跡，也不需要 IK。因此我們只需使用速率函數表示關節的旋轉量，以橫座標表示時間，縱座標表示關節的旋轉量 (腰部關節記錄 z 軸旋轉量，而頭部關節則是記錄 x 軸旋轉量)。

圖 10 是我們設計的身體運動規劃演算法。傳入的參數是多個音樂片段。在第 1~3 行中，我們準備了三個變數記錄目前腰部關節的旋轉量、目前累積下來的音長和旋轉方向。第 4~7 行是規劃每個音樂片段中的每個音符。第 8~13 行，判斷音符的 *isBeat* 是否為真和音長是否太短，我們認為小於一拍的音長為短，目的是避免身體快速不斷地運動。首先，我們規劃腰部關節旋轉回到起始位置，其後再規劃表達節奏點的動作。使用圖 11 預設的速率函數，使用變數 *accumulate* 比例縮放橫軸，和使用 $P_{current}$ 比例縮放縱軸。其後，我們規劃關節的旋轉量，並使用圖 12 預設的速率函數進行比例縮放。比例縮放的值是音符的音強經正規化後乘上 *swing* 和 *direct*，*swing* 是使用者指定的偏好參數，代表腰部關節的旋轉幅度，值越大則旋轉幅度越大。*direct* 是控制旋轉的方向，其值在 1 與 -1 之間交替使用，

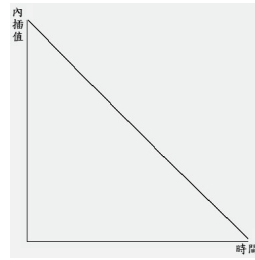


圖 11 預設回歸旋轉量速率函數

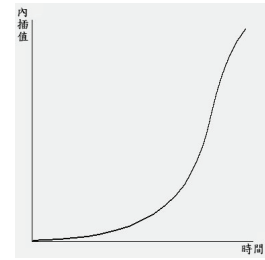


圖 12 預設旋轉量速率函數

表格 1 實驗參數

參數名稱	實驗 a	實驗 b
w1	1	1
w2	1	1
w3	1	1
w4	1	1
v	2	2
swing	7	7
nod	10	10
Y _{scale}	14	3
smooth	100	11
reflectance	100	12
L1	85	26
L2	50	58

實驗編號	第一段軌跡	第二段軌跡	第三段軌跡
a			
b			

圖 13 不同參數的動作軌跡的實驗結果

代表腰部以正反方向交替旋轉。第 14~15 行，若以上條件沒有成立則累加音符的音長。

頭部運動規劃演算法和身體運動規劃演算法相似，只需將 *swing* 參數改為 *nod* 參數，*nod* 是系統提供給使用者指定的偏好參數，代表頭部關節的旋轉幅度，值越大則幅度越大。

6. 實驗結果與討論

6.1 產生不同風格動畫測試

在這個實驗中，我們測試論文所提出的參數組的效用，測試經由調整參數是否能產生不同風格的動作來詮釋音樂。對於同一首音樂同樣的切割，我們使用差異較大的參數組產生動畫，再比對動畫之



圖 14 實驗 a 的連續截圖



圖 15 實驗 b 的連續截圖

間是否具有不同風格的差異。實驗的音樂使用巴哈的 Air on the G String，參數組如表格 1 所示。

由於空間的限制，我們只顯示三段實驗 a, b 所產生的運動軌跡。從圖 13 中顯示，實驗 a 的結果呈現較圓滑多變的軌跡，這是因為參數 smooth 很大，通過非節奏點的軌跡較圓滑所致。此外，在第一段軌跡的第二個關鍵格的軌跡顯示，通過關鍵格之前的軌跡可由 $L1$ 控制，值越大則幅度越大；在通過關鍵格之後的軌跡的方向有很大的幅度改變，這是因為參數 reflectance 很大的原因。實驗顯示以上三個參數具有明顯改變軌跡的功用，而在我們調整參數的過程中，參數 $L2$ 與 Y_{scale} 只具有微調效用。圖 14 及圖 15 是實驗 a 及 b 所產生之動畫的連續截圖。

6.2 產生不同音樂風格的測試

在這個實驗中，我們測試系統對音樂的通用性，我們從不同網站中任意選取多首 MIDI 音樂作為輸入，測試系統能否讀取音樂資料並產生人體動畫。測試音樂共 12 首，其中 4 首古典樂，4 首鄉村和 4 首爵士。

測試顯示的音樂能成功讀取並產生動畫。圖 16 中我們以曲風為分類，分別顯示每一曲風的一首音樂的測試結果。每一首我們只截取前三段運動軌跡，圖 17~圖 19 為動畫的連續截圖。

7. 結論與未來發展

曲風	第一段軌跡	第二段軌跡	第三段軌跡
古典			
鄉村			
爵士			

圖 16 不同曲風之動作軌跡的實驗結果



圖 17 古典曲風之範例的連續截圖



圖 18 鄉村曲風之範例的連續截圖



圖 19 爵士曲風之範例的連續截圖

在本研究中，我們發展了一個與音樂相關的虛擬人物動作規劃系統，規劃虛擬人物的活動，以表達音樂特徵。我們以手部動作表達旋律，身體的律動動作顯示節奏。基於人類聽覺的短暫性，我們將音樂切割成多個片段，每個片段再獨立規劃動作。切割方式以手部活動限制設計目標函數做為評分條件。我們並提出了一個音樂到動作的對應，讓3D虛擬環境中的虛擬人物可以根據這個模型將音樂特徵對應到動作特徵上。實驗結果顯示，規劃的動作有明顯的節奏存在，音長和音高較不明顯。此外，系統通用於LOA1模型和MIDI檔案，藉由改變參數組可產生不同風格的動作，以符合使用者偏好或音樂曲風。

在MIDI格式中一首音樂中往往不會明確地分出旋律音軌和節奏音軌，因此旋律可能由一個以上的樂器同時演奏，或者由一個以上相同的樂器同時演奏。我們目前是以人工選擇的方式找出旋律音軌。在[13]中提到旋律可分為前景和背景，聽眾往往會注意聆聽旋律的前景部份。未來我們希望能讓使用者選擇多個不同樂器的音軌，經由系統分析自動找出前景部份。

目前我們使用切割器切割旋律，切割的方法是根據目標函數做為最佳切割演算法的評分條件，而目標函數目前考慮的因素有音域、節奏密度和音長；未來我們希望能考慮更多音樂因素，如休止符和旋律中的基本三個模式：反復、級進和跳進，讓手部運動更有效地利用空間表達。再者，我們根據音符的特徵設計出音樂特徵模型，但音樂特徵除了低階的特徵外，還有比較高階的特徵，如調性、曲風等，未來我們也希望可以善加利用。目前對於不同曲子的音樂，我們的系統使用事先設計好的參數組，未來希望擷取音樂高階的特徵並自動調整參數組。另外，在程序式動畫中，規劃運動的演算法目前使用了身體的上半身，未來可以延伸至下半身運動，以增加動作的多樣性和豐富性。

致謝

此研究在國科會 NSC 96-2221-E-004-008 計畫的支助下完成，特此致謝。

參考文獻

- [1] M. Cardle, L. Barthe, S. Brooks, and P. Robinson, "Music Driven Motion Editing: Local Motion Transformations Guided By Music Analysis," in *Proc. of the Eurographics UK Conference*, 2002.
- [2] DeLone *et al.* (Eds.), "Aspects of *Twentieth-Century Music*," Englewood Cliffs, New Jersey: Prentice-Hall, Chap. 4, pages 270-301, 1975.
- [3] W.J. Dowling, "Scale and Contour: Two components of a theory of memory for melodies," *Psychological Review*, 1978.
- [4] R. Laban, and L. Ullmann, *Mastery of Movement*, Princeton Book Company Publishers, 1960.
- [5] H.C. Lee, and I.K. Lee, "Automatic Synchronization of Background Music and Motion in Computer Animation," *Computer Graphics Forum*, Volume 24, pages 353-362, 2005.
- [6] M.Y. Liao, and J.F. Liao and T.Y. Li, "An Extensible Scripting Language for Interactive Animation in a Speech-Enabled Virtual Environment," in *Proc. of the IEEE International Conference on Multimedia and Expo*, 2004.
- [7] S. Mishra, and J.K. Hahn, "Mapping motion to sound and music and in computer animation and VE," in *Proc. of the Pacific Graphics '95*, 1995.
- [8] J. Nakamura, T. Kaku, T. Noma, and S. Yoshida, "Automatic Background Music Generation Based on Actors 'Emotion and Motions'," in *Proc. of the Pacific Graphics*, 1993.
- [9] S. Oore, and Y. Akiyama, "Learning to Synthesize Arm Motion to Music By Example," in *Proc. of the 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2006.
- [10] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Detecting dance motion structure through music analysis," in *Proc. of IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, 2004.
- [11] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Dancing-to-Music Character Animation," in *Computer Graphics Forum*, Volume 25, pages 449-458, 2006.
- [12] L. Torresani, P. Hackney, and C. Bregler, "Learning Motion Style Synthesis from Perceptual Observations," in *Proc. of the Neural Information Processing Systems Foundation*, 2006.
- [13] A.L. Uitdenbogerd, and J. Zobel, "Manipulation of music for melody matching," in *Proc. of ACM International Multimedia Conference*, 1998.
- [14] IKAN (Inverse Kinematics using Analytical Methods). <http://cg.cis.upenn.edu/hms/software/ikan/ikan.html>