

行政院國家科學委員會專題研究計畫 成果報告

由人類表演之三維運動數據中自動挖掘基本動作與行為模 組

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-009-064-

執行期間：93年08月01日至94年07月31日

執行單位：國立交通大學資訊工程學系(所)

計畫主持人：陳稔

報告類型：精簡報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中 華 民 國 94 年 10 月 29 日

由人類表演之三維運動數據中自動挖掘基本動作與行為模組

Automatic Discovery of Motion Primitives and Behavior Units from 3D Human Performance Motion Data

計畫編號：NSC-93-2213-E-009-064

執行期限：93年8月1日至94年7月31日

主持人：陳稔教授 / 國立交通大學資工系

E-mail: zchen@csie.nctu.edu.tw Tel: 03-5731875

1. 摘要

本計畫是針對人類表演時所做的動作進行分析及再利用。首先我們利用前期計畫所建立 Vision-based Mo-Cap 系統收集人體運動的數據。其次，我們對每套表演動作的人體 3D data 表示成一個 stick figure 的 tree structure。一個 input motion stream 是依序列的 frame sequence。我們將此 frame sequence 做分段工作(segmentation)。為了降低運動空間的維度 (motion space dimensionality)，我們採取分割全身為雙手雙腳等四個肢節(substructures)。我們分段工作是針對各個肢節去做。獲得分段後的各個 motion segments，我們開始進行基本動作(motion primitives)的自動挖掘 (automatic discovery)。我們將採用維度降低 (dimension reduction) 及多重解析度分析 (multi-resolution analysis) 的方法。在抽取出基本動作後，我們利用線性動態系統 (linear dynamic system) 描述基本動作，評估運動參數後，作為一個基本程序 (generic motion procedure)。我們更進一步將各肢節的基本動作，同步起來去組合一個動作單元 (body action units)，然後再分析全部 motion stream 中有哪些動作單元擁有某種經常的時間連接順序關係 (temporal relation)，去構成一個更高階的行為模組 (behavior units)。至此，我們初步對整個表演動作做了註解 (annotation)，可供將來製作動畫時重複使用。

關鍵詞：運動分段、人型支架、維度降低、運動相似度、基本動作、動作自動挖掘、行為模組、運動限制

Abstract

This research project is concerned with the analysis and re-use of 3D human performance motion data. We will use the vision-based motion capture system we developed in the previous NSC research

project to collect human motion data. An input motion stream consists of a sequence of frames. Each motion data represents a sequence of 3D stick figure in various body postures. We need to decompose the whole sequence into several motion subsequences called motion segments. In order to derive these segments from the input stream we shall further break the whole body into four substructures (i.e. left/right arms and left/right legs.) and then consider the motion sequence of each individual substructure. In this way, we can reduce the motion space dimension significantly. We apply some segmentation technique to separate the entire motion sequence into several motion segments. To discover the motion primitives in the motion segments, we will propose to use both dimension reduction technique and multi-resolution analysis to facilitate the task. After the motion primitives are extracted, we will model each motion primitive as a linear dynamic system (LDS) to get the meaningful motion parameters. This motion parameter set of the LDS model for the motion primitive will serve as a generic procedure or module for the latter reuse purpose. We proceed to coordinate all substructures in a coordinated and synchronized manner to derive the whole-body action units. We can pursue to the next higher level units of temporal organization of a number of action units called behavior units. Up to this stage, we have analyzed the motion stream and come up with some sort of motion descriptions or annotations for the further use in animation creation.

Keywords: Motion Segmentation, Stick Figure, Dimension Reduction, Motion Similarity, Motion Primitive, Action and Behavior Discovery, Motion Constraints

2. 緣由與目的

近年來，電腦動畫技術日趨成熟，在虛擬實境

或動畫電影中，展現逼真的人體運動已成為重要的研究課題。由於人體動作捕捉設備(Motion Capture, Mo-Cap)的普及，使用 Mo-Cap 拍攝真實演員所表演的動作序列(motion sequence)，將其套用到虛擬環境中的人體模型上，已成一趨勢。此法可快速產生逼真的人體運動，並模擬演員動作之特色(characteristics or styles)。其最大缺點在於，Mo-Cap 所取得的資料，難以重複使用(re-use)，一旦虛擬環境的場景或人物模型改變，或是所需的動作略有不同，經常就需要重新拍攝。

為提高重複使用(re-use)的能力，有以下幾個步驟：

- (1) Derivation of motion or behavior primitives [1]-[7]：此步驟著重在將 Mo-Cap 所取得的動作序列，作分段(segmentation)與歸類(classification)的動作，例如將一連串武打動作，分割為出拳、踢腿、格擋、跳躍等代表性的動作單元。使用者即可根據這些基本的動作單元，重新組合出與當初拍攝時不同的動作序列。
- (2) Motion warping [8][11][12]：將一段動作序列，根據使用者所設定的限制條件，例如：人體模型的移動路徑(path)、速度、肢體的位置與方向、或是預計產生出的動作序列時間長度等等，修改為符合條件的新動作序列，但仍保有原始動作序列的特色。
- (3) Motion blending [9][10][14]：將二或多段不同的動作序列(motion examples)，根據限制條件作 warping 及 interpolation/synthesis，融合為一段新動作序列。這包括了套用 graph search [10]及 signal processing [24]的方法等。
- (4) Motion retargeting [13]：Mo-Cap 取得的動作資料，套用到不同身型比例(anthropometry)的人體模型時，會有失真的問題。此步驟即是提供方法，在套用的同時，盡可能保有原始動作資料的特徵。

以上四點中，(1)又是其他三點的基礎，本期計畫也將著重在(1)。我們將使用前期所建立的 Vision-based Mo-Cap 系統捕捉表演者的動作，並透過 Dimension Reduction、2nd-order Auto Regression Model、Clustering、Hidden Markov Model 等技巧，建立一套能將人體運動自動歸類為代表性的動作單元的系統，而這些動作單元將可提供給動畫師編輯或產生新型態的動畫使用。

3. 研究方法

我們進行人體運動分析的流程，可分為以下幾個步驟：

- 建立 3D Motion Data Set：使用本實驗室自行研發的 Vision-based Human Motion Capture System，由兩台攝影機所攝得的人體動作，計算得到人體各肢節點的三維座標(3D Joint position)。而這些隨時間變化的肢節點座標，便稱作 Motion Stream。
- Motion Segmentation：將全身肢節分成雙手雙腳四個 Substructure，分別對各 Substructure 的 Motion Stream 作分段(Segmentation)。首先將原本分佈在三維空間中的動作，作 Dimension Reduction 粗分為數個平面運動，再配合 2nd-order Auto Regression Model 作進一步的分段，得到 Substructure Motion Segment。(今年度紀文原碩士論文[18])
- Motion Primitive Discovery：我們將對 Substructure Motion Segment 做分群(Clustering)的動作。首先將同屬於同一平面的運動歸為同類，再將同一平面運動中，具有相似 Linear Dynamic System 參數(即 AR Model 的參數)的運動做更細緻的分群，即可得到各 Substructure 中具有相同特徵的 Motion Primitive。
- Body Action Unit / Behavior Unit Discovery：將四肢的 Motion Primitives 所構成的 Body Action Unit 轉換成多特徵觀察值序列後，我們要利用事先訓練出的全身動作單元描述模型 Hidden Markov Model，對其進行辨識並標籤。如此便將原本的連續動作轉換成具有語義的陳述，有了動作單元的描述序列後，使用者可以利用它針對不同的應用問題去建立適合的行為描述。(今年度顧正偉碩士論文[29])

以下分各小節對上述步驟做詳細的描述。

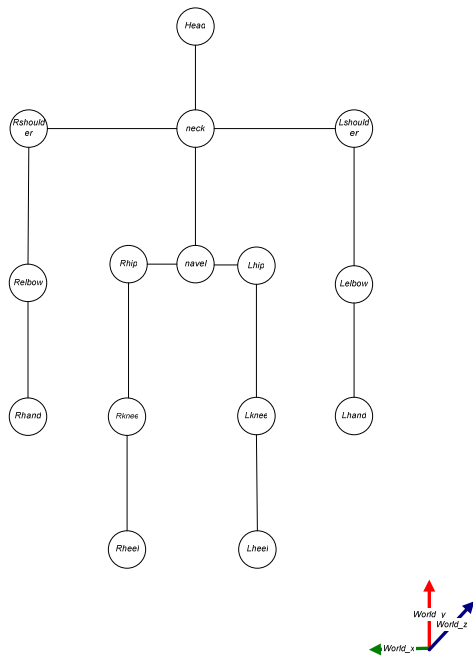
3.1 建立 3D Motion Data Set

我們使用前期計畫所研發的 Vision-based Human Motion Capture System [15]，由兩台攝影機自不同方向拍攝表演者動作之影像序列，如(圖 1)。

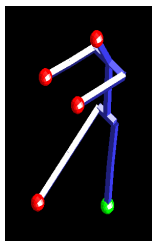


(圖 1) 使用兩台攝影機拍攝表演者動作

根據預先定義的人體骨架結構(圖 2)，由影像中取得各肢節點的三維座標(joint position)，而這些初步取得的肢節點座標會受到相機校正的誤差、肢節點標定不精確等因素的影響，因此還必須經過 Inverse Kinematics 演算法的修正，由 end-effector (末端的肢節點)配合人體各關節活動範圍的 physical constraint，可以估算出更精確的肢節點位，此部分的理論請參考[16][17]。(圖 3)展示的就是(圖 1)之原始影像經 Inverse Kinematics 計算得到的人體肢節模型，以 stick figure 方式呈現。(今年度蘇詠迪碩士論文[17])



(圖 2) 人體骨架與各關節點的定義



(圖 3) IK 修正後得到的人體 stick figure

3.2 Motion Segmentation

由於人體多數的動作都是透過四肢的運動來表現，且四肢的構造又有相似性，都是一個 3-DOF (Degree of Freedom)的關節接上一個 1-DOF 關節再

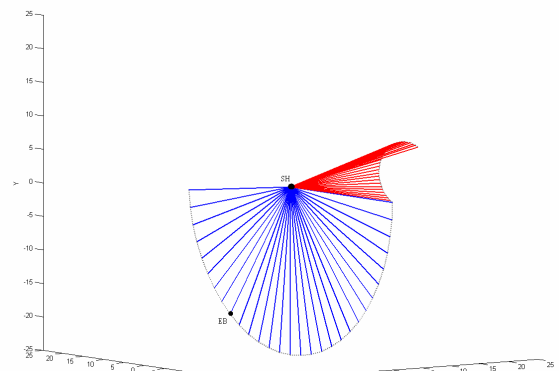
接上 end-effector。以左手而言，參考(圖 2)，其結構為 LShoulder (3-DOF)—Lelbow (1-DOF)—Lhand (End-effector)，右手及左右腳也都類似。因此我們可以將全身的肢節分為左右手及左右腳等四個 substructure，先分別對 substructure 的動作粹取出基本的動作單元(substructure motion primitive)，然後再將四肢的動作單元整合為全身的動作(body action unit)。

在粹取基本動作單元之前，必須先將前一步驟所得的 substructure motion stream (隨時間變化的肢節點座標，joint position trajectory)，切分為有意義的段落，稱作 substructure motion segment。以下便以手的動作為例，簡述 motion segmentation 的作法，細節請參考[18]。至於腳的動作可比照辦理。

3.2.1 以平面作初步分割

以手部運動為例，手肘是 1DOF 的關節，其所連接的肢節只能在固定的平面上運動，所以要計算手肘關節轉動角度只需要看兩個肢節(前臂與上臂)的夾角，即可表示手肘的旋轉角度；而肩膀是 3DOF 的關節，可以在三維空間中自由地轉動，但若找出旋轉時的旋轉軸，即可將 3DOF 降為 1DOF，再將前後時刻上臂向量依 quaternion 計算法，即可求出肩膀的旋轉角度。

上臂在三維空間中的運動軌跡根據肢節與轉軸垂直與否主要可分為共面(肢節與轉軸垂直)、共圓錐(肢節與轉軸不垂直)等兩種模式，見(圖 4)。藉由這兩種模式的組合而產生出各種不同的運動，而共圓錐模式在短時間內亦可視為共面。所以第一個工作即是要找出動作中的共面運動。



(圖 4) 運動平面示意圖：由共面運動(藍色線段)與共圓錐運動(紅色線段)組成。灰色虛線為手肘運動軌跡、SH：肩膀關節點、EB：手肘關節點。

3.2.1.1 Plane Segmentation

在運動過程中，利用肩膀到手肘的肢節向量來

尋找動作的旋轉平面，該平面的法向量即是動作的旋轉軸；而要求空間中兩個向量的旋轉軸最快的方法莫過於對這兩個向量做外積。但是，實際情況並非如此簡單，除了轉軸未必與肢節垂直外，從捕捉系統得到的 3D 點位並不是非常準確，些微的誤差都會讓做外積得到的向量發生偏移，此此我們只能改求最適合平面，使得所有點位到平面的距離最短，即求解一個平面 $aX + bY + cZ + d = 0$ ，使得

$$D = \sum_{t=1}^N |ax(t) + by(t) + cz(t) + d| \text{ 最小。}$$

由於我們並無法事情得知哪些時刻的關節點位同屬一個平面運動，因此只能採取 Greedy Approach 做時間軸長度(time resolution)的逐段切割，所遵循的原則是「點位到其所屬的平面距離必然遠小於到其它平面的距離」，設定一個容許值，來限定距離多少以內者視為同一平面，而在距離大於容許值時，將動作切開。

此外，還需要考慮到兩個平面動作間 Smooth Transition 的少數 Frame。因為動作才剛發生變化，其點位與到上一平面的距離可能會小於我們設定的容許值，如此一來會被視為動作捕捉上的誤差而將加入上一個運動平面中。要等到動作發生一段時間後，點位與上一個運動平面的距離大於容許值時才會進行切割。如此一來，得到的分段點會與真實的分段點會有時間的延遲，且將不屬於平面的點位加入，會導致擬合出來的平面發生偏移，此一偏移會連帶影響到下一個平面的準確性，而使得切割的結果愈來愈不精確。為了避免這樣的誤差一直傳遞下去，將這少數影格捨棄不列入平面擬合的計算中。我們提出的演算法如下：

Plane Segmentation Algorithm

Input: a set of points $S = [p_1, p_2, p_3, \dots]$, and distance threshold d_{th}

Output: a segmented sets of points $S_1 = [p_1, p_2, \dots, p_{i_1}]$, $S_2 = [p_{i_1+1}, p_{i_1+2}, \dots]$, ...

Method:

1. **Form the initial planes** $E = [E_1, E_2, \dots, E_n]$
 - (1) For the input set S, assume that the first m points to form an initial plane E_1' .
And initiate $S_1' = \{p_1, p_2, \dots, p_m\}$.
 - (2) Check if a point $p_i \in \{p_{m+1}, p_{m+2}, \dots\}$ belongs to E_1' ,
i.e., if $d(p_i, E_1') < d_{th}$, then add p_i to S_1' ,
and refit E_1' with S_1' .
Until $d(p_i, E_1') > d_{th}$. Let this point be p_{i_1} .
 - (3) Then remove the last k points,

$$S_1 = S_1' - \{p_{i_1-k}, \dots, p_{i_1-1}, p_{i_1}\} = \{p_1, p_2, \dots, p_{i_1-k-1}\},$$

, and refit $E_1 = E_1'$ with S_1 .

- (4) For the remaining points of the input set $S = S - S_1'$,
repeat part (1)~(3) to obtain E_2, \dots, E_n .

2. Refine $E = [E_1, E_2, \dots, E_n]$ to remove outlier points.

- (1) Choose l outlier points whose distances E_i are above the threshold. $i = 1 \sim n$.
- (2) By removing them, refine E_i with the remaining points.

3.2.1.2 Plane Merge

透過前一小節的演算法雖能將共面的點位集合切割出來，但第二種共圓錐模式的運動卻會被切割成多個平面(over-segmentation)。因此，我們還需要對切出來的平面進行檢查，將能形成共圓錐模式的平面合併(merge)，找出其真正的運動平面。

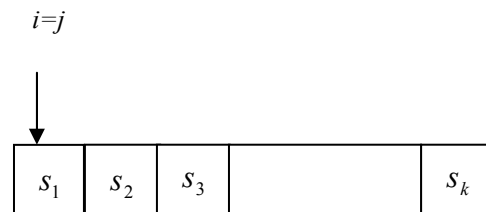
一組散佈在三維空間中的點位集合，其維度自然為三。但如果可以在空間中找到一個平面，該平面包含這組點位集合，則可推論這組點位的維度從三維退化(degenerate)到二維。共圓錐模式的運動在上述的方法中被過度切割成數個小平面，將這些小平面合併起來後，因為合併後的點集合確實屬於同一平面，所以其維度還是會退化到二維。若是不屬於共圓錐模式，則合併後維度會變成三維。以此條件便可以決定是否將兩個相鄰的平面合併為一個共圓錐運動。演算法如下：

Plane Merge Algorithm

Input: A series of segmented sets of points $s_1, s_2, \dots, s_k \in 2D$, each one with sorted eigenvalues $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_1 \geq \lambda_2 \geq \lambda_3$. And a threshold of eigenvalue, e_{th} .

Output: A merged sets of points $S_1, S_2, \dots, S_m, m < k$

Method:



1. Initial i points to s_1 ($i = 1$).
2. For the set s_j and s_{j+1} , merge them, and recalculate its dimension.
i.e.,
<i> Find out the eigenvalues $\lambda_1, \lambda_2, \lambda_3$,
<ii> Normalize $\lambda_1, \lambda_2, \lambda_3$, such that

$$\lambda_1 = 1, \lambda_2 = \frac{\lambda_2}{\lambda_1}, \lambda_3 = \frac{\lambda_3}{\lambda_1}$$

- <iii> Compare if $\lambda_3 < e_{th}$.
- (1) If $\lambda_3 < e_{th}$, then the dimension of the merged set is 2.
j set to j+1, and repeat part 2..
 - (2) Else, the dimension of the merged set is 3.
then split s_j and s_{j+1} , merge the sets $s_i \sim s_j$ into the larger one.
i.e., $S_m = \{s_i, s_2, \dots, s_j\}$.
i set to j+1, j set to i, m set to m+1, and repeat part 2..
- Until $s_i = s_k$.

3.2.2 對平面運動中的轉動角度做進一步切割

找出上臂運動的平面之後，就可以將 3DOF 的肩膀關節，改以單一轉軸(即平面的法向量)來表示，對該轉軸的轉角則可用 quaternion 運算法則求出。再加上手肘關節的角度(即上臂與下臂的夾角)，就可以描述使用兩個角度，描述整隻手的運動。

根據[19][20][21]，我們可以用 2nd-order AR (Auto-Regression) Model 來描述角度的變化，並藉由觀察參數矩陣的變化來進行動作切割。AR model 是用來描述一組資料中，其 t 時刻狀態 $x(t)$ 與先前時刻狀態 $x(t-1), x(t-2), \dots$ 有相關的一種動態模型(Dynamic Model)，二階 AR 模型可表示為 $x(t) = c_1 x(t-1) + c_2 x(t-2) + e(t)$ ，為了分析上的便利，令 $X(t) = \begin{bmatrix} x(t-1) \\ x(t) \end{bmatrix}$ ，則可將二階 AR 模型以矩陣的型式表示成：

$$X(t) = A \cdot X(t-1) + B \cdot e(t), \text{ 其中, } A = \begin{bmatrix} 0 & I \\ A_2 & A_1 \end{bmatrix}$$

用來表示模型中的決定性參數, $B = \begin{bmatrix} 0 \\ B_0 \end{bmatrix}$ ，用來表示模型中的隨機性參數， e_i 為 zero mean 的高斯雜訊。

以手部運動為例，狀態向量 $x(t)$ 就是 t 時刻時肩膀與手肘的關節角度，即 $x(t) = \begin{bmatrix} \theta_{shoulder}(t) \\ \theta_{elbow}(t) \end{bmatrix}$ 。

3.2.2.1 AR Model Segmentation

在一個穩定的運動中，關節角度應該呈現出穩定增加或減少的情況。故在此一穩定運動過程中，其 AR 模型中的參數矩陣 A 應該保持不變。換句話說，當動作發生變化時，參數矩陣 A 也必需修正以符合新的變化模式。如此一來，動作切割的問

題就轉變成偵測參數矩陣 A 變化的問題，並在參數矩陣 A 發生劇烈變化時予以切割。對於這樣的問題，可以採用 recursive system estimation 的方法來求解[22]。

利用 $W(t)$ 來表示參數矩陣變化的部分，參數矩陣 A 變化在 State Space Model 中表示如下：

$$\begin{aligned} A(t+1) &= A(t) + W(t) \\ X(t) &= A(t) \cdot X(t-1) + E(t) \end{aligned} \quad (3.1)$$

透過最小化觀察值 $X(k), k=1 \sim t$ 與預測值 $\hat{X}(k) = A \cdot X(k-1), k=1 \sim t$ 的誤差，可以計算出參數矩陣 $A(t)$

$$\begin{aligned} V_t(A(t)) &= \sum_{k=1}^t \beta(t, k) \|X(k) - A(t) \cdot X(k-1)\|^2 \\ \beta(t, k) &= \prod_{j=k}^t \lambda, \lambda \text{ is a constant} \end{aligned} \quad (3.2)$$

對(3.1)式的 A 為偏微分，可以得到估算的參數矩陣 $\hat{A}^T(t)$ ，

$$\hat{A}^T(t) = \left(\sum_{k=1}^t \beta(t, k) X(k-1) \cdot X^T(k-1) \right)^{-1} \sum_{k=1}^t \beta(t, k) X(t-1) \cdot X^T(t) \quad (3.3)$$

再將計算相鄰時刻參數矩陣 A(t) 的差異 $d(t) = \|\hat{A}(t) - \hat{A}(t-1)\|$ ， $d(t)$ 的大小反映出相鄰矩陣的差異程度，同時也表現出系統的變化程度。如此一來，可以取 $d(t)$ 中的區域最大值(local maxima) 當作動作的切割點。演算法總結如下：

AR Model Segmentation Algorithm

- Step 1. 將輸入的角度資料 $X(t), t=1 \sim n$ 利用 low pass filter 去除雜訊，並減去平均值，得 $Y(t), t=1 \sim n$
- Step 2. 將角度資料 $Y(t)$ 代入公式(3.1)中估算參數矩陣 A，並計算相鄰時刻的差異矩陣 $D(t) = \hat{A}(t) - \hat{A}(t-1)$
- Step 3. 計算差異矩陣的 norm $d(t) = \|D(t)\| = \|\hat{A}(t) - \hat{A}(t-1)\|$ ，在這裡利用 MSE (mean square error) 來計算 matrix norm， $A = (a_{ij})_{m \times n}, \|A\| = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2$
- Step 4. 將 $d(t)$ 代入 1D RLS
- Step 5. 從 Step.4 中的結果，取出序列中的最大值，該處即是切割點
- Step 6. 將切割出來的片段，進行動態模型擬合(Dynamic Model Fitting)，若是模型計算出的資料與觀察到的資料誤差超過容許值時，回到 Step.5，對此一片段重新計算切割點

經過上述平面與 AR Model 的分段後，我們得到一系列的動作片段(motion segment)，而每一片

段都可用其平面方程式與 AR 模型之參數來描述其特徵。

3.3 Motion Primitive Discovery

Motion Segmentation 完成後，先將所截取到的點位資料，根據其是否在同一平面上先做初步分群 (Clustering)，並做 Labeling；而在同一平面上的動作，則利用二階 AR 模型做更細部的 Motion Primitives 的切割與分群，並做更細微的 Labeling。細節如下所述：

3.3.1 針對運動平面做初步分群

- (1) 將平面從 World Coordinate System 轉換至 Local Coordinate System：利用平面將動作片段點位做初步分類以後，由於所有的平面都處在 World Coordinate System，這樣會造成兩個一樣的動作，因為表演者面向角度的不同，或是位置的不同，將兩個動作視為不同類型的運動。為了解決這問題，我們必須將每個平面由 World Coordinate System 轉至該 Substructure 的 Root 肢節點的 Local Coordinate System，以左手為例就是 LShoulder 的 Local Coordinate System，見(圖 2)。
- (2) Clustering：每個平面可表示為參數式 $aX + bY + cZ + d = 0$ ，我們可將平面參數 (a, b, c, d) 表示為一 feature vector，用 K-mean clustering 演算法對此各個運動平面的 feature vector 做初步分群。

3.3.2 對屬於同一平面的運動做細部分群

而在同一平面上的運動中，可能會混合有幾種不同的動作，因為動作的角度變化方式，或是角速率變化不同。所以，我們依據二階 AR 模型所獲得的參數，經過下列的處理產生出特徵向量去做更細微的分群及 Labeling。

- (1) 描述 Motion Segment 的參數矩陣：

如前所述，每個 motion segment 都可以用一個二階 AR 模型表示為

$$X(t) = A \cdot X(t-1) + B \cdot e(t)$$

矩陣 A 可用來代表此段動作片段的角速度變化模式。我們可進一步解出矩陣 A 之 eigenvalues 用來代表其特性。

- (2) 將 Motion Segment 描述為 Damped Harmonic Motion：

幾乎所有人類的動作均為 Damped Harmonic Motion，且二階 AR 程序是種特別的 Gaussian Markov 二階程序。而此種程序如在 1D 這樣的狀況下，是一種 Damped Harmonic Oscillation，亦即：

$$x(t) = \alpha \exp^{-\beta t} \cos(2\pi f t + \phi) + T$$

α : amplitude, β : damping_rate, f : frequency, ϕ : phase, T : offset

而最主要描述此種運動的兩個參數為 β 與 f 。而 α, ϕ, T 可由 minimize 重建出來的資料與原本資料的誤差而獲得，在這邊我們不會用到。

假設我們從矩陣 A 中獲得兩個 eigenvalues, λ_1, λ_2 ，且給定取樣頻率為 τ 。則根據[19]，可獲得：

$$\begin{cases} \beta_m = \frac{1}{\tau} \log \frac{1}{|\lambda_m|}, m=1,2 \\ f_m = \frac{1}{2\pi\tau} \arg \lambda_m \end{cases}$$

之後，對於每個 Motion Segment，便可得到兩組的 β 與 f 。

- (3) Clustering & Labeling：

利用 K-Means algorithm 做為 Clustering 的基礎演算法，輸入為每個 Motion Segment 所獲得的 β 與 f 。為了方便，轉換成以下的特徵向量： $[\beta_1 \ f_1 \ \beta_2 \ f_2]_i, i=1 \dots n \dots$ 。Cluster 完成後，將每群給一個 Label，即用此 Label 代表一個 Motion Primitive。

3.4 Body Action Unit / Behavior Unit Discovery

3.4.1 對於全身性的 Motion Primitive 做 Labeling

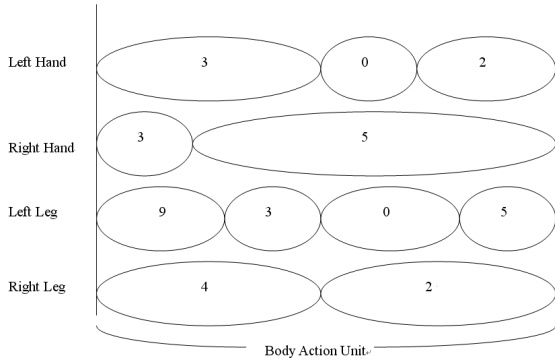
3.3 節所提到的 Labeling 方法，只針對單一肢節 (Substructure) 的動作去做 Motion Primitive 的 Labeling。而我們所需要的是全身性 (Body Unit) 的 Motion Primitive 的 Label 對應。我們提出的方法如下：

- (1) 將四肢的 Motion Primitive 所對應之 Label 以及該動作所持續的時間，依橫軸為時間的關係排列，如(圖 5)。
- (2) 將時間軸做固定長度的取樣，分割成許多個小時間觀察。
- (3) 在同一小時間點上，若四肢的 Primitive 結束時間接近於某一範圍內(如(圖 5)之 Left Hand(2), Right Hand(5), Left Leg(5), Right Leg(2))，則將該時間點視為一 Body Action Unit 的結束時間。
- (4) 將一 Body Action Unit 依照開始時間跟結束時間取出，並將新的 Body Action Unit 的開始時間設定為上一 Body Action

Unit 結束時間。

(5) 做到所有資料均處理完為止。

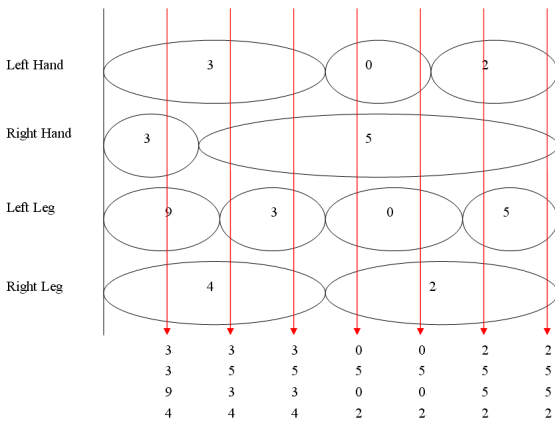
之後，我們便可以針對所獲得之 Body Action Units 做 Label 的對應。



(圖 5) 四肢之 Motion Primitives 依照時間關係所排成之圖形

3.4.2 資料型態轉換

至此我們已描述了如何對身體的四肢動作分別進行了 segment 和 cluster，並藉由四肢的運動單元 (motion primitives) 的共同週期切割出全身動作單元 (body action unit) 後，我們必須將這些資訊整合為擁有相同時間單位的多個符號序列，以進行更高階的後續分析，因此我們設定一間格 λ (單位為 frame 數)，每隔 λ 對四肢的運動單元作 sampling，取得當時各肢節所標記的符號做為我們的觀察值，來產生出多特徵觀察值序列 (multiple feature streams)，以上述的 body action unit 為例，下圖表示了此轉換過程。



(圖 6) 動作單元轉換多特徵觀察值序列示意圖

此全身運動單元所得的 multiple feature streams 即為

Left Hand	3	3	3	0	0	2	2
Right Hand	3	5	5	5	5	5	5
Left Leg	9	3	3	0	0	5	5
Right Leg	4	4	4	2	2	2	2

而我們會在接下來的章節說明如何分析轉換後如上述形式的資訊。

3.4.3 全身動作單元的多特徵觀察值序列特性分析

經由觀察數各由數個同一類型的全身運動單元轉換後的多特徵觀察值序列內容後，可以發現：即使是由同種類型的動作單元轉換而來，其內容也會有相當的變異性 (因為隨著不同的人，不同的時間，作出來的動作樣子總會略有不同)，而沒有辦法用單純的序列比對來加以辨識，然而我們相信同一類型的序列仍然擁有某種隱含的性質可以被模型化。在本文中我們挑選了 HMM 此雙重程序隨機程序模型，希望可以利用機率的特性來吸收這些變異性，以具彈性的方式描述出各個動作單元的特性。

3.4.4 基本型 HMM 介紹

隱馬可夫模型 (Hidden Markov Model, HMM) 是一個常常被拿來做時間序列資料的分析、辨識的模型 [23][24]。目前在語言、手寫文字、手勢的辨識上，都被使用的非常廣泛 [24][25][26]；而在動作分析上，也常常被拿來使用 [27][28]

HMM 的基本設計，就是由內部「狀態」(states) 之間的轉換以及各狀態呈現的「觀察值」(observations) 來描述整個資料的模型。而實際上，所能得到的資訊只有觀察值，而無法知道內部狀態的轉換的情況；這就是 HMM 中 hidden 的觀念。利用這個 hidden 的觀念，HMM 更能夠接受觀察資料的變化、誤差，讓 HMM 本身的通用性更高。

HMM 是用多個狀態所構成的，而對於模型的描述，則是透過狀態之間的轉換以及各狀態呈現的觀察值來描述。因此，在 HMM 中擁有下列的主要元素：

- (1) $S = \{S_1, S_2, \dots, S_N\}$ 為狀態的集合，裡面的 S_j 為狀態 (state)， $j = 1, 2, \dots, N$
- (2) $V = \{v_1, v_2, \dots, v_K\}$ 為所有可能觀察到的值的集合，其中 v_k 為可觀察到的值， $k = 1, 2, \dots, K$ 。
- (3) q_t 於時間 t 時的狀態， $t = 1, 2, \dots, T$ ， $q_t \in S$ 。
- (4) O_t 於時間 t 的觀察值， $O_t \in V$ 。 $O = O_1 O_2 \dots O_T$ 代表由 $t = 1$ 到 $t = T$ 的觀察值序列。
- (5) π ，起始狀態機率矩陣， $\pi = \{\pi_i\}$ ， $i = 1, 2, \dots, N$ ； $\pi_i = P(q_1 = S_i)$ ，代表 $t = 1$ 時，狀態是 S_i 的機率，所以 $0 \leq \pi_i \leq 1$ ， $\sum_{i=1}^N \pi_i = 1$ 。

(6) A , 狀態轉換的機率矩陣, $A = \{a_{ij}\}$,
 $i = 1, 2, \dots, N$, $j = 1, 2, \dots, N$;
 $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$ 代表由狀態 i 轉變到
 狀態 j 的機率, 所以 $0 \leq a_{ij} \leq 1$, $\sum_{j=1}^N a_{ij} = 1$ 。

(7) B , 特定狀態下的觀察值分佈機率矩陣,
 $B = \{b_j(v_k)\}$, $j = 1, 2, \dots, N$, $k = 1, 2, \dots, K$;
 $b_j(v_k) = P(O_t = v_k | q_t = S_j)$ 代表在狀態 j 的情
 形下, 觀察值為 v_k 的機率, 所以 $0 \leq b_j(v_k) \leq 1$,
 $\sum_{k=1}^K b_j(v_k) = 1$ 。

而在一般情況下, 會把 π 、 A 、 B 這三個機率矩
 陣合在一起, 稱為 HMM 的參數 λ ,
 $\lambda = (\pi, A, B)$; 而實際上, 只要有 λ , 就足以代
 表一個完整的 HMM 模型了。

在使用 HMM 時, 會有三個主要的問題, 分別
 是:

(1) 機率計算

計算在給定 $\lambda = (\pi, A, B)$ 的情形下,
 $O = O_1 O_2 \dots O_T$ 這組觀察值序列發生的機率
 $P(O | \lambda)$; 也就是要去計算 O 這組觀察值序列對由
 λ 構成的 HMM 模型的可能性。

(2) 最佳狀態轉換序列

在已知 $\lambda = (\pi, A, B)$ 和觀察值序列
 $O = O_1 O_2 \dots O_T$ 的情形下, 去推算出一組最有可能
 的狀態轉移路徑 $Q = q_1 q_2 \dots q_T$ 。

(3) 調整模型參數

根據觀察值序列 $O = O_1 O_2 \dots O_T$ 調整 HMM 的
 參數 $\lambda = (\pi, A, B)$, 讓 $P(O | \lambda)$ 的值越大越好; 也
 就是要讓 HMM 模型最能描述觀察值序列 O 。這也就
 是訓練 HMM 時所要使用的方法。

關於此三個問題的演算法, 請詳見[24]。此
 外, 在實作上因為要避免機率連乘造成浮點數
 underflow 的解決方法, 也就是所謂的 scaling
 problem, 請詳見[29]。

3.4.5 擴充標準型 HMM 至多觀察值隨機程序模型

標準型的 Hidden Markov Model 中, 在每一個
 狀態下一次只能處理一個觀察值, 如此構成單一觀
 察值序列。但是在全身運動單元的辨識上, 每個狀
 態都必須處理來自四肢的觀察值符號, 因此我們需
 將基本型的 HMM 擴充成能在每一個狀態中處理

多觀察值的隨機系統, 我們可以考慮以下幾種可
 能的擴張方法。

(1) 合併為單一序列

將 M 個序列中同一時間的觀察值組合成一個
 新的值, 並利用組合出來的值的序列來當作新的單
 一觀察值序列。這個方法的缺點是所組成的可能觀
 察值會是原來不同序列的可能觀察值的所有組
 合, 總共的數目會非常大。假使原來的 M 個觀察
 值序列是由可能觀察值數目分別為 K_1 、 K_2 、 \dots 、
 K_M 的觀察值集合所組成, 那所組合成的觀察值的
 可能總數目將會有 $\prod_{s=1}^M K_s$ 個; 這樣不但需要相當大
 的記憶空間, 也需要相當長的運算時間。

(2) 多個 HMM 組合

對於 M 個觀察值序列各別建立 HMM 模型,
 並個別訓練、識別; 最後要計算其整體機率時, 在
 將各序列的機率乘起來。而對於 M 個序列, 也可
 以對 M 個模型加入權重, 來修改其對整體的重要
 性。這種想法很簡單, 對於原始的 HMM 方法也不
 需要做出太大的修改; 但是這樣的缺點是沒辦法考
 慮到同一時間下, M 個序列中觀察值之間的關係。

(3) HMM with Multiple Feature Streams

對於 HMM 中的每一個狀態, 都同時對 M 個
 序列中觀察值紀錄個別的機率分佈; 整體的觀察值
 機率, 就由 M 個觀察值的機率乘積來組成; 這樣
 可以考慮在同一狀態中的 M 個觀察值, 而所需的
 記憶空間和運算時間都不會增加太多。而他的特
 色就是各觀察值的關係是機率獨立的(無法考慮之
 間的條件機率)[30]。

我們採用的是(3)的方法, 以其 Hidden Markov
 Model with Multiple Feature Streams (HMM/MFS)
 為主要的模型架構。

3.4.6 HMM/MFS 的修改

由於基本的 Hidden Markov Model 無法夠同時
 接受多組觀察值, 所以要將多特徵序列的觀念套用
 在 HMM 上, 對於變數的定義需做出以下修改:

- (1) M 為同一個時刻所觀察到的觀察值數目。
- (2) 可能的觀察值集合 V , 每組觀察值序列都有
 自己的可能觀察值集合, 總共有 M 項,
 $V = \{V^{(1)}, V^{(2)}, \dots, V^{(M)}\}$ 。其中 $V^{(s)}$ 為第 s 個觀
 察值序列的所有可能觀察值集合,
 $V^{(s)} = \{v^{(s)}_1, v^{(s)}_2, \dots, v^{(s)}_{K_s}\}$, $1 \leq s \leq M$; K_s 為
 第 s 個觀察值序列中各時刻的所有可能觀察

值的個數。 $v^{(s)}_j$ 是第 s 個觀察值序列的觀察值， $1 \leq j \leq K_s$ 。

(3) 觀察值序列 $O = O_1 O_2 \dots O_T$ ，代表時間長度為

T 的觀察值序列。 $O_t = \begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix}$ ，代表在時間

t 的 M 個觀察值；其中 $O^{(s)}_t \in V_s$ 代表於時間 t 時，第 s 個序列中的觀察值， $1 \leq s \leq M$ ， $1 \leq t \leq T$ 。

如此，原來 HMM 中的觀察值機率 $b_j(O_t)$ 則變為

$$b_j(O_t) = b_j \left(\begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix} \right),$$

其中 $b_j \left(\begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix} \right)$ 代表在狀態 j 時，觀察值為

$\begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix}$ 的機率， $1 \leq j \leq N$ 。

而整個狀態的觀察值機率可由 M 個觀察值的機率乘積來表示

$$b_j \left(\begin{bmatrix} O^{(1)}_t \\ O^{(2)}_t \\ \vdots \\ O^{(M)}_t \end{bmatrix} \right) = b^{(1)}_j(O^{(1)}_t) \times b^{(2)}_j(O^{(2)}_t) \times \dots \times b^{(M)}_j(O^{(M)}_t) \\ = \prod_{s=1}^M b^{(s)}_j(O^{(s)}_t)$$

， $1 \leq j \leq N$

其中， $b^{(s)}_j(O^{(s)}_t)$ 代表在狀態 j 時，在第 s 個序列所觀察到的觀察值為 $O^{(s)}_t$ 的機率；而 HMM 中的觀察值分佈機率矩陣 B 也將由 $\{b_j(v_k)\}$ 變為 $\{b^{(s)}_j(v^{(s)}_k)\}$ 。此外，training 的演算法中也需要若干修正，詳見[30]。

3.5 Motion Editing

從上述方式所獲得之全身動作單元序列後，可嘗試將這些基礎模型的 motion representation parameters 配合 retargeting 及 cloning 等應用做修飾，以便產生新動作。但是此一工作由於沒有拿到商用的 3D Motion Animation 軟體(如 MAYA)，既很費時、又

不漂亮，所以目前所建立出來的全身動作單元數據，只提供動畫師做編輯或其它後置處理用。

4. 實驗結果

目前拿來作為實驗中事先訓練好的全身動作單元之類型共有八種，包括彎腰、彎腰直立、蹲下、蹲下起身、踢腿、出拳、上勾拳、正步共八種，每一種動作都有四十個以上的樣本。樣本中的動作分成數次拍攝，有些是由不同的人來做；而每一個動作速度、樣子也有所不同。(圖 6)~(圖 13)為所有動作的雙視圖，(表 1)則是樣本數目列表。

(表 1) 各種動作類型的樣本數目

動作名稱	樣本數目
彎腰	43
彎腰起身	43
蹲下	40
蹲下起身	40
踢腿	41
出拳	66
上勾拳	42
正步	50
總和	365

我們對每種動作單元列舉四個多特徵觀察值序列的樣本(使用四肢 primitives 的分群數分別為 Left Hand: 10, Right Hand: 10, Left leg: 7, Right Leg: 7 之樣本)

(表 2) 動作一：彎腰 多特徵觀察值序列 範例列表

Sample1	9, 9, 3, 0, 0, 5, 2 4, 4, 4, 2, 2, 2, 2 3, 3, 3, 0, 0, 2, 2 3, 5, 5, 5, 5, 5, 5
Sample2	9, 9, 4, 3, 0, 0, 5 4, 3, 4, 2, 2, 3, 2 4, 4, 4, 3, 0, 2, 2 6, 1, 5, 5, 5, 5, 5
Sample3	9, 9, 9, 9, 3, 3, 0, 5, 5, 1 4, 3, 3, 5, 4, 2, 2, 2, 5, 2 1, 1, 1, 4, 4, 3, 0, 0, 2, 2 4, 6, 6, 6, 6, 5, 5, 5, 5, 5
Sample4	9, 9, 9, 3, 0, 5, 1 4, 4, 4, 3, 0, 5, 2 1, 4, 4, 3, 0, 2, 2 4, 6, 6, 5, 5, 5, 5

(表 3) 動作二：彎腰起身 多特徵觀察值序列 範例列表

Sample1	1, 5, 0, 3, 9, 9 2, 5, 0, 3, 3, 9 2, 2, 0, 3, 3, 3 5, 5, 5, 5, 5, 3
Sample2	5, 0, 3, 3, 9, 9, 9, 9 2, 2, 3, 3, 9, 9, 9, 9 2, 2, 0, 3, 4, 1, 1, 1 5, 5, 5, 2, 6, 6, 6, 6
Sample3	1, 5, 0, 0, 3, 3, 9, 9, 9, 9, 9 2, 5, 0, 0, 3, 3, 4, 4, 4, 4, 1 2, 2, 0, 0, 3, 3, 4, 4, 1, 4, 1 5, 5, 5, 3, 6, 2, 2, 4, 4, 4, 4
Sample4	5, 0, 3, 3, 9, 9 2, 0, 5, 9, 7, 6 2, 0, 3, 4, 1, 1 5, 5, 5, 6, 2, 6

(表 4) 動作三：蹲下多特徵觀察值序列 範例列表

Sample1	9, 9, 9, 7, 7 1, 9, 9, 3, 7 1, 1, 4, 3, 0 6, 4, 6, 2, 2
Sample2	9, 9, 9, 9, 9, 7, 7 9, 9, 3, 9, 3, 2, 7 1, 1, 4, 3, 3, 0, 2 6, 3, 6, 5, 2, 2, 2
Sample3	9, 9, 9, 3, 3, 3, 3, 7 8, 4, 9, 3, 3, 0, 3, 1 1, 1, 4, 4, 3, 0, 0, 0 6, 4, 5, 6, 2, 2, 1, 1
Sample4	9, 9, 9, 7, 7, 7 9, 1, 4, 7, 3, 7 1, 1, 4, 3, 3, 0 6, 4, 4, 5, 2, 2

(表 5) 動作四：蹲下起身多特徵觀察值序列 範例列表

Sample1	7, 7, 9, 9, 9, 9 6, 3, 9, 5, 1, 5 0, 3, 4, 4, 1, 1 2, 1, 1, 4, 1, 6
Sample2	7, 7, 7, 3, 9, 9, 9 6, 7, 7, 3, 9, 1, 5 2, 0, 0, 3, 4, 4, 1 2, 2, 1, 1, 2, 6, 1
Sample3	7, 7, 7, 9, 9, 9, 9 7, 7, 3, 9, 9, 1, 5 0, 3, 4, 4, 4, 1, 1 1, 1, 1, 1, 4, 4, 6
Sample4	7, 3, 3, 3, 3, 9, 9 6, 3, 3, 7, 9, 1, 1 3, 3, 3, 4, 4, 1, 1

	1, 1, 4, 5, 4, 1, 1
--	---------------------

(表 6) 動作五：踢腿多特徵觀察值序列 範例列表

Sample1	9, 9, 9, 9, 4, 4, 9, 9, 9, 9, 9 7, 9, 9, 0, 4, 4, 9, 5, 9, 1, 2 1, 1, 4, 3, 0, 2, 0, 3, 4, 1, 1 5, 6, 3, 0, 0, 0, 0, 5, 5, 1, 6
Sample2	9, 9, 4, 4, 4, 9, 9, 9 9, 4, 4, 4, 3, 9, 9, 1 1, 4, 3, 0, 2, 0, 3, 1 1, 3, 5, 0, 0, 0, 5, 3
Sample3	9, 9, 9, 9, 4, 4, 4, 4, 8, 8, 8, 8 4, 9, 1, 1, 3, 4, 5, 4, 8, 8, 1 4, 1, 1, 4, 3, 0, 2, 0, 4, 4, 4 4, 4, 1, 1, 5, 5, 0, 5, 1, 2, 1
Sample4	9, 9, 9, 4, 4, 3, 4, 4, 9, 8, 8, 9 8, 9, 4, 4, 4, 3, 4, 4, 2, 8, 8, 2 1, 1, 4, 3, 2, 2, 2, 0, 3, 4, 1, 4 4, 3, 3, 5, 0, 0, 0, 5, 2, 2, 4, 2

(表 7) 動作六：出拳多特徵觀察值序列 範例列表

Sample1	8, 8, 6, 6, 8, 8, 8 7, 8, 3, 3, 8, 8, 8 1, 4, 3, 4, 4, 4, 1 1, 3, 3, 3, 3, 3, 1
Sample2	9, 8, 6, 2, 6, 8, 8, 9, 9 7, 8, 6, 3, 3, 8, 7, 9, 7 1, 4, 3, 0, 3, 3, 4, 1, 1 1, 3, 3, 0, 3, 3, 3, 1, 4
Sample3	8, 8, 6, 6, 2, 6, 6, 6, 8 7, 8, 6, 5, 2, 6, 7, 9, 7 1, 1, 1, 4, 3, 3, 4, 1, 1 1, 3, 3, 3, 0, 3, 3, 4, 1
Sample4	9, 9, 9, 8, 6, 2, 2, 6, 8, 8, 8, 8, 8, 8 9, 7, 9, 3, 6, 2, 2, 6, 8, 9, 8, 7, 8, 8, 7 1, 1, 4, 4, 3, 0, 0, 0, 3, 4, 4, 1, 1, 1, 1 6, 1, 3, 3, 3, 0, 3, 3, 3, 3, 2, 2, 4, 3, 6

(表 8) 動作七：上勾拳多特徵觀察值序列 範例列表

Sample1	8, 8, 8, 6, 2, 6, 6, 8 8, 7, 3, 6, 2, 3, 7, 8 1, 1, 3, 3, 3, 4, 4, 1 2, 1, 3, 3, 3, 3, 4, 1
Sample2	8, 3, 3, 8, 6, 6, 8, 8, 8, 8, 8 6, 3, 3, 3, 3, 7, 8, 8, 8, 9, 6 1, 4, 4, 0, 0, 3, 3, 3, 4, 4, 1 6, 2, 3, 3, 3, 3, 3, 2, 1, 6, 1
Sample3	8, 8, 8, 3, 8, 6, 6, 8, 6, 6, 8 8, 7, 7, 3, 8, 6, 6, 9, 8, 7, 8 4, 1, 1, 1, 4, 3, 4, 4, 4, 1, 1 2, 6, 1, 6, 3, 3, 4, 4, 4, 1, 2
Sample4	8, 8, 8, 8, 9, 9, 9, 9, 8, 8, 6, 8, 8, 8, 8, 8, 8, 8, 8 8, 7, 6, 8, 9, 9, 9, 5, 8, 8, 6, 7, 8, 8, 7, 8, 8, 7, 8, 8 8 1, 1, 4, 3, 3, 3, 3, 3, 3, 0, 0, 3, 4, 3, 3, 4, 1, 1, 1 1 6, 4, 2, 3, 2, 3, 5, 2, 1, 2, 2, 2, 2, 5, 5, 2, 2, 4, 2, 2

	1
--	---

(表 9) 動作八：正步多特徵觀察值序列範例列表

Sample1	9, 9, 9, 9, 9, 8, 4, 6, 8, 8, 9, 9, 9, 9, 9, 9 1, 9, 4, 9, 9, 9, 8, 4, 6, 8, 0, 9, 9, 4, 9, 7, 9 1, 1, 4, 4, 3, 0, 2, 2, 2, 2, 0, 0, 3, 4, 1, 1, 1 6, 1, 3, 3, 5, 5, 2, 2, 2, 2, 2, 2, 2, 3, 3, 1, 6
Sample2	9, 9, 9, 4, 6, 8, 9, 9 4, 9, 9, 2, 6, 7, 9, 1 1, 4, 0, 2, 2, 0, 4, 1 6, 3, 5, 2, 2, 2, 3, 6
Sample3	9, 9, 9, 8, 6, 6, 8, 9, 9, 9 9, 1, 9, 8, 2, 2, 8, 4, 9, 1 1, 1, 3, 0, 2, 2, 0, 4, 1, 1 6, 1, 3, 5, 5, 2, 2, 3, 3, 6
Sample4	9, 9, 9, 8, 8, 6, 6, 8, 8, 9, 9 9, 7, 7, 8, 8, 2, 6, 5, 8, 9, 1 1, 1, 3, 0, 2, 2, 0, 0, 4, 1, 1 6, 1, 3, 5, 2, 2, 2, 5, 3, 3, 6

擁有了在各種四肢動作單元分群數目下所轉換而來的多特徵觀察值序列群後，我們便可在不同的四肢動作單元分群數目下訓練出描述各類動作單元的 HMM/MFS，為了要驗證我們所訓練的模型確實能描述不同類型動作單元(即對不同類型的動作單元能有鑑別力)，我們用人工的方式產生二十組已知動作單元內容的連續動作，每一組連續動作都至少有十五個以上的動作單元，如下表所示。

		動作單元順序 編號									
		1	2	3	4	5	6	7	8	9	...
連續動作組別編號	1	正步	彎腰	彎腰起身	蹲下	蹲下起身	出拳	上鉤拳	踢腿	踢腿
	2	彎腰	彎腰起身	出拳	正步	蹲下	蹲下起身	出拳	蹲下	蹲下起身
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

接著，我們再對其內部每一個動作單元的類別至我們所擁有該動作單元樣本中的多特徵觀察值序列隨機抽取，並且加上雜訊，產生作為測試的 synthetic data。

接下來使用產生出的 synthetic data 作為辨識

系統的輸入，並且統計總和的分析結果之正確率。其中控制變因的參數組有兩組：

(1) 四肢動作單元各別的分群數目：

在 synthetic 階段時依照不同的四肢動作單元的分群數目，會到不同分群數目轉換出的多特徵觀察值序列樣本組中隨機抽取來產生，而辨識階段時也要使用在相同四肢運動單元分群數目下所訓練出的 HMMs/MFS。

(2) 代表每一運動單元的 HMM/MFS 的訓練資料數目

由於使用不同筆數的訓練資料會影響到最後產生出的 HMMs/MFS，我們分別使用不同筆數的訓練資料來產生不同的 HMMs/MFS，並且分別測試其辨識結果之正確率，來驗證訓練資料多寡對模型的適應性之影響。

我們分別調整兩個控制變因的參數組，在各類參數設定組合對二十組的 synthetic data 中的動作單元進行辨識，各類參數設定組合的整體辨識正確率如下表所示。(黃色背景區塊代表在訓練資料數目分別為 5 筆、10 筆、20 筆下對四肢運動單元進行辨識所得之最高正確辨識率，分別為 80.61%、87.3%、93.08%)

		每一動作單元訓練資料筆數		
		5 筆	10 筆	20 筆
四肢動作單元的分群個數	Left Hand : 5 Right Hand : 5	79.41%	84.33%	87.33%
	Left Leg : 3 Right Leg : 3			
	Left Hand : 7 Right Hand : 7	80.61%	87.39%	90.475%
	Left Leg : 5 Right Leg : 5			
	Left Hand : 10 Right Hand : 10	79.22%	85.36%	93.08%
	Left Leg : 7 Right Leg : 7			
Left Hand : 15 Right Hand : 15	76.89%	83.79%	90.71%	
Left Leg : 10 Right Leg : 10				

5. 結論

本計畫提出一套人體動作分析系統，可將動作擷取設備(Mo-Cap)所錄製的人體表演資訊，分段並歸類為四肢的基本動作單元(Motion Primitive)，而四肢的動作單元有可進一步整合為全身的動作單元(Body Action Unit / Behavior Unit)，這些動作單元將可提供給動畫師做編輯或產生出全新的動畫序列使用。

6. 參考文獻

[1] O.C. Jenkins and M.J. Mataric, "Automated Derivation

- of Behavior Vocabularies for Autonomous Humanoid Motion,” Proceedings of Autonomous Agents and Multi Agent Systems, Jul. 2003.
- [2] O.C. Jenkins and M.J. Mataric, “Deriving Action and Behavior Primitives from Human Motion Data,” Proceedings of IEEE International Conference on Intelligent Robots and Systems, Oct. 2002
- [3] Y. Yacoob, “Parameterized Modeling and Recognition of Activities,” Computer Vision and Image Understanding, Vol.73, No.2, pp.232-247, Feb. 1999
- [4] Y. Li, T. Wang, and H.Y. Shum, “Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis,” Proceedings of ACM SIGGRAPH 2002.
- [5] M. Brand and A. Hertzmann, “Style Machines,” Proceedings of ACM SIGGRAPH 2000.
- [6] V. Pavlovic, J.M. Rehg, T.J. Cham, and K.P. Murphy, “A Dynamic Bayesian Network Approach to Figure Tracking Using Learned Dynamic Models,” International Conference on Computer Vision, Sep. 1999.
- [7] A.J. Ijspeert, J. Nakanishi, and S. Schaal, “Trajectory Formation for Imitation with Nonlinear Dynamical Systems,” Proceedings of IEEE International Conference on Intelligent Robots and Systems, Oct. 2001.
- [8] J. Lee and S.Y. Shin, “A Hierarchical Approach to Interactive Motion Editing for Human-like Figures,” Proceedings of ACM SIGGRAPH 1999.
- [9] S.I. Park, H.J. Shin, and S.Y. Shin, “On-line Locomotion Generation Based on Motion Blending,” Proceedings of ACM SIGGRAPH 2002.
- [10] O. Arikan and D.A. Forsyth, “Interactive Motion Generation from Examples,” Proceedings of ACM SIGGRAPH 2002.
- [11] A. Witkin and Z. Popovic, “Motion Warping,” Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Sep. 1995.
- [12] F.W. da Silva, L. Velho, J. Gomes, and S. Goldenstein, “Motion Cyclification by Time X Frequency Warping,” IEEE XII Brazilian Symposium on Computer Graphics and Image Processings, Oct. 1999.
- [13] K.J. Choi and H.S. Ko, “On-line Motion Retargetting,” The 7th Pacific Conference on Computer Graphics and Applications, Oct. 1999.
- [14] A. Bruderlin and L. Williams, “Motion Signal Processing,” Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Sep. 1995.
- [15] “追蹤人體單純運動之多相機模式比對視覺系統”研究報告: NSC-92-2213-E-009-081, 國立交通大學。
- [16] 吳政峰, “人體肢節三維動作捕捉之電腦視覺技術”, 國立交通大學資訊工程學系, 碩士論文, 民國93年。
- [17] 蘇詠迪, “利用逆向動力學修正立體視覺人體動作原始資料”, 國立交通大學資訊工程學系, 碩士論文, 民國94年。
- [18] 紀文原, “手部平面動作切割”, 國立交通大學, 碩士論文, 民國94年
- [19] C. M. Lu and N. J. Ferrier, “Automated Analysis of Repetitive Joint Motion”, IEEE Trans. Information Technology in Biomedical System, 2003.
- [20] C. M. Lu and N. J. Ferrier, “Repetitive Motion Analysis : Segmentation and Event Classification”, IEEE Trans. Pattern Anal. Machine Intell., Feb. 2004.
- [21] C. M. Lu, H. Z. Liu, and N. J. Ferrier, “Multidimensional motion segmentation and identification,” Proc. IEEE Conf. Computer Vision and Pattern Recog., pp. 629–636, 2000.
- [22] L. Ljung and T. Soderstrom. Theory and Practice of Recursive Identification, The MIT Press, 1983.
- [23] L. Wang, W. Hu, and T. N. Tan, “Recent developments in human motion analysis”, Pattern Recognition, 36(3), pp. 585–601, 2003.
- [24] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proc. of IEEE, 77(2), pp. 257-285, February 1989.
- [25] M. Zimmermann, H. Bunke. “Hidden Markov Model Length Optimization For Handwriting Recognition Systems”, Proc. of the 8th International Workshop on Frontiers in Handwriting Recognition, pp. 369–374, 2002.
- [26] P. Somervuo, “Speech Recognition Using Context Vectors And Multiple Feature Streams”, Helsinki University of Technology, Master Thesis, 1996.
- [27] J. Yamato, J. Ohya, and K. Ishii, “Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model”, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 379–385, 1992.
- [28] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato, “Hierarchical Recognition of Daily Human Actions Based on Continuous Hidden Markov Models”, Proc. of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 779-784, May 2004.
- [29] 顧正偉, “Human Motion Classification by Hidden Markov Model with Multiple Feature Streams”, 國立交通大學資訊工程學系, 碩士論文, 民國94年。
- [30] P. Somervuo, “Speech Recognition Using Context Vectors and Multiple Feature Streams”, Helsinki University of Technology, Master Thesis, 1996.

