

上下肢協同運動的程序式動畫模擬

Procedural Animation of Four-Limb Coordinated Human Motions

陳映似 (Ying-Szu Chen), 李蔡彥 (Tsai-Yen Li)

國立政治大學資訊科學學系

Department of Computer Science, National Chengchi University

中文摘要

電腦動畫是極具發展潛力的數位內容產業之一，其應用遍及娛樂、傳播與工業等領域。目前常用的電腦動畫產生方式之一為運動捕捉 (Motion Capture)，但其所擷取到的資料運用彈性較低。另一方面，程序式動畫依照特定演算法判斷環境而產生對應的動作，運用的彈性較高。本計畫以程序式動畫的方式，針對協同上肢及下肢運動的程序進行設計，以自動模擬出人體四肢協同運動的動畫。我們透過相關資料蒐集及人體運動的觀察，設計出數個符合環境限制條件的手腳並用運動之程序，並將此程序實現在一個自行設計的3D人體動畫平台上。

Abstract

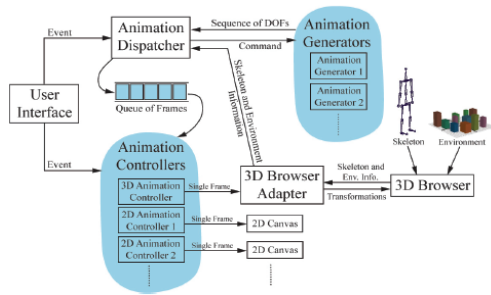
Computer animation is one of the digital contents with great development potential in many applications in entertainment, communication and industry. Currently, motion capture is one of the most commonly used ways to create animations, but the captured motions are difficult for reuse. On the other hand, procedural animation, which generates motions through appropriate algorithms, is more flexible and can be made adaptive to the environment. In this paper, we use the procedural animation approach to design algorithms to generate the coordinated motion of the four limbs for human figures. Through observation of hu-

man motions, we have designed several motion procedures that can generate four-limb coordinated motions that can satisfy human kinematics and environment constraints. We will demonstrate the experimental results in a human animation platform that we have developed.

1. 研究動機與問題

電腦動畫是數位內容產業的其中一環，內容廣泛地應用在日常生活中，是極具發展潛力的產業項目之一。商業上的應用包含遊戲、電視、電影等視聽娛樂；而在軍事、工業以及醫學領域中，藉由電腦動畫的模擬，更能有效率的達到精準的預測。

一般而言，電腦動畫的產生方式分為三類：運動捕捉(Motion Capture)[3]、動力學模擬(Dynamic Simulation)[4]及程序式動畫(Procedural Animation)[1]。目前為了製作動作自然的動畫，大多的動畫採用運動捕捉的方式製作。雖然這樣所產生的動畫自然度相當高，但所產生的動畫除了在原設計的情境下使用，較難再運用在其他情境，缺乏應用上的彈性。亦即在每個差異甚少的環境中做類似動作的模擬時，都需要重新擷取，相當耗時耗力。相對而言，程序式動畫是一個具發展潛力但尚待發展的方法，它是依據運動法則與環境條件來計算動作，自動判斷環境而產生適當



圖一：IMHAP 的系統架構

的動作[9][7]。

人體運動的模擬一直都是電腦動畫的重點課題之一。有些研究著重於上肢運動，有些則著重於下肢運動；而本計畫則強調上肢及下肢的協同運動，例如伏地跪姿爬行、攀爬梯子等動作。由於這些動作與環境限制的點較一般運動來的多，因此所需設計的程序也更具挑戰性。我們希望能透過此類人體運動程序的設計，讓虛擬環境中的數位演員能在複雜的環境中自動產生適當的運動。例如，在障礙賽的模擬中，虛擬人物將會依照障礙物的高低形狀，自動判斷需以何種運動方式越過該障礙物。

2. 背景

2.1. IMHAP 系統

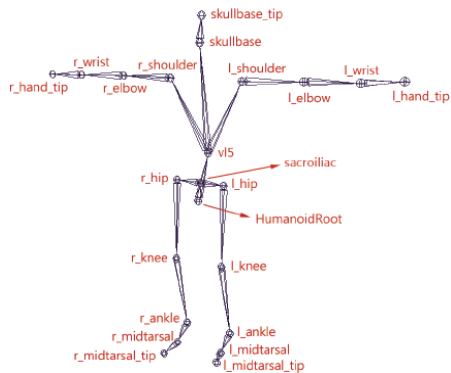
實現程序式動畫的方式很多，但多是以自行開發應用軟體的方式進行。我們則是在本過去的研究成果所開發的一個人體動畫模擬平台上進行。這是一個名為 IMHAP (Intelligent Media Lab's Humanoid Animation Platform) 的動畫平台 [5]。IMHAP 的系統架構如圖一所示。此系統的架構主要是根據 MVC 模式設計的。在 MVC 中，系統被區分成三大模組：Model、View 以及 Controller，Model 用來封裝資料，View 與使用者互動，而 Controller 則負責前兩者之間的溝通。

在圖一中，每個方塊表示一個模組，而模組間的箭頭表示資料流向，其中 Animation Dispatcher 和 Animation Generator 是整個系統的核心，它們扮演 Model 的角色。Animation Controller 負責控制動畫的播放，所以它是 MVC 裡的 Controller。剩下的 3D Browser、2D Canvas 以及 User Interface 都是與使用者互動的元件，所以它們是 MVC 裡的 View。以下段落將詳細描述各個模組的功能。

3D Browser 負責繪圖工作，它可以讀取人體模型和環境資料。3D Browser Adapter 用來封裝 3D Browser，對其他模組提供一個統一的介面，這麼做的優點是可以讓系統動態地抽換不同的 3D Browser，而不用更動到系統本身。

Animation Dispatcher 和 Animation Generator 是整個系統的核心。Animation Dispatcher 可以作為全域規劃器 (Global Planner)，它從使用者介面接收高階命令，間接指示它所掌管的多個 Animation Generator。而 Animation Generator 則是區域規劃器 (Local Planner)，負責產生個別運動。舉個例子來解釋這兩個模組之間的關係：使用者透過使用者介面輸入了一條行進路徑給 Animation Dispatcher，Animation Dispatcher 接收這道命令之後，根據環境中的障礙物，計算出何時該用何種運動，如此便將一個高階的命令拆成數個低階的指令，之後將工作分派給適當的 Animation Generator，讓 Animation Generator 產生運動細節後，才把結果交由給 Animation Dispatcher 整理成一連串的畫格。

Animation Controller 從 Animation Dispatcher 讀取畫格，並控制動畫的播放。使用者可以透過使用者介面去控制 Animation Controller，進而控制 3D



圖二：H-Anim LOA1 所定義的人體骨架

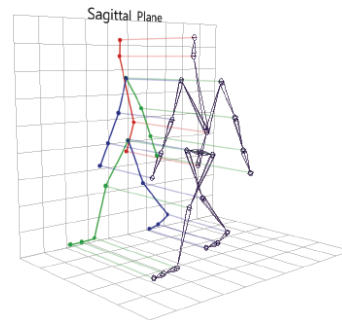
Browser 或 2D Canvas 應該顯示第幾個畫格。系統中允許多個 Animation Controller 存在，多個 Animation Controller 可以讓應用程式同時擁有多個視圖。例如，2D Animation Controller 可連接到 2D Canvas，提供像是上視圖、側視圖、或是曲線圖等 2D 視圖。而 3D Animation Controller 則可連接到 3D Browser Adapter，提供 3D 透視圖的瀏覽。

2.2. 人體模型：LOA1 模型

我們所採用的人體模型為VRML [8] 裡H-Anim 標準[10]的 LOA 1 所定義的骨架，如圖二所示。此人體模型是由十八個關節所組成。在 3D 空間中，HumanoidRoot 有 6 個自由度（平移和旋轉各有 XYZ 三個自由度），而其他關節有 1~3 個自由度。為了減輕負擔，我們在設計運動時，只考慮側視圖投影平面（Sagittal Plane），如圖 3 所示。如此一來，除了 HumanoidRoot 的自由度為 3（兩個平移自由度加上一個旋轉自由度）之外，其他關節的自由度均減為 1，大幅簡化問題的複雜度。

3. 動畫設計步驟

如前節所述，為了簡化計算，本研究



圖三：將 3D 人體骨架投影到 2D 側視平面上

在觀察人體運動之後，省略掉較細微的變化與不規則的動作，並將人體模型投影到一個 2D 平面後(如圖三)，訂立出人體行進時動作轉換的關鍵部份，也就是一般所謂的關鍵格(keyframe)。有了這些關鍵格後，接下來我們可使用單純的內插法來計算出剩餘的畫格，最後在將所產生出的畫格串連起來，完成動畫。

歸納上述產生關鍵格與內插的方法，我們的動畫程序設計的流程可分為以下四步驟：

- **觀察**：經由觀察實際動作，找出動作與動作之間的轉折點，將整體動作分段(phase)。
- **訂立關鍵格**：根據觀察的結果，將人體模型的組態依據該模型的各個關節長度及角度背景做適當的設定，使得人體模型可以做出關鍵格(key frame)之動作，並且不違背自然機制；即依據運動法則與環境限制來計算人體模型的組態。
- **設計內插過程**：利用關鍵格內插(interpolation)產生關鍵格與關鍵格之間的動作畫格。同樣需注意運動法則與環境限制。例如：直線軌跡與曲線軌跡的選擇、緩進緩出或是等速運動的決定並且避免一切碰撞的發生。
- **動畫呈現**：將內插產生的動畫畫格資料輸入 IMBrowser[6]，以 3D 立體圖



圖四：爬梯子的關鍵格

的方式呈現動畫結果。

4. 上下肢協同動作設計

人體上下肢協同之動作眾多，其中最容易在日常生活裡見到的包含攀爬梯子以及跪姿爬行。因此我們以此二個動作為例，進行動畫程序的設計。

4.1. 爬梯子

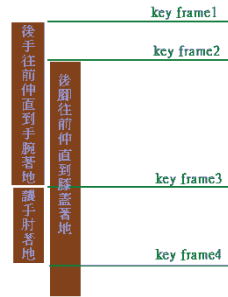
4.1.1. 動作階段分析

我們稱在空中移動而未與環境接觸的手或腳為「移動手」及「移動腳」。另外，我們稱重心所在的手及腳為「重心手」及「重心腳」。爬梯的動作十分單純，分成如下兩個階段。

- **階段一：**重心腳打直、移動手往上伸兩個梯距；移動腳曲膝。此時重心腳為跨上階梯的腳、反之移動腳為曲膝腳、移動手則為與重心腳同側的手。
- **階段二：**將移動腳往上再跨上一個階梯。

4.1.2. 關鍵格設計

- **關鍵格一：**雙手抓住梯子兩旁，重心手在上、移動手在下，兩腳分別踩在兩個階梯上，移動腳打直、重心腳踩在往上一階階梯；這個關鍵格是四肢



圖五：跪姿爬行之動作分析

唯一皆接觸梯子的時刻。

- **關鍵格二：**重心腳正好打直、移動腳則曲膝懸空，移動手向上握住梯子。這兩個關鍵格的範例如圖四所示

4.1.3. 關鍵格之間的內差方法

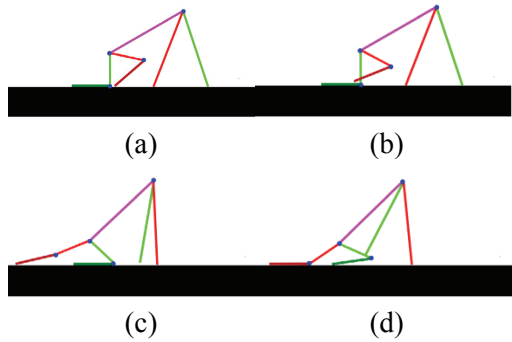
- **第一階段的內差方法：**第一與第二關鍵格之間的內插，目的為將整個身體往上撐。我們經由觀察發現，移動腳曲膝的同時，腳板為了避免與階梯產生碰撞，會自然的打直。而重心腳則會因為省力關係，減小腳踝角度，使重心造成的力矩變小。
- **第二階段的內差方法：**第二關鍵格至第一關鍵格之間的內插，目的為將移動腳腳板位置移至向上一階之階梯位置。內插過程中，需考慮腳板若直接上移，則容易與階梯產生碰撞，因此上移時的軌跡為拋物線。

4.2. 跪姿爬行

在此運動中，手和腳雖然總共只有三個動作，但由於動作並非單純的一個接續一個，而是有部份時間同時並行的，因此我們必須將整個動作切割為四個部份作內插，如圖五。

4.2.1. 動作階段分析

- **階段一：**移動手(即後手)往前伸。



圖六：跪姿爬行動作之關鍵格定義

- **階段二：**移動手往前伸不久，移動腳(即後腳)也往前伸，直到移動手手腕著地。
- **階段三：**移動手手肘下壓準備著地，移動腳繼續往前伸。
- **階段四：**移動腳繼續往前伸直到膝蓋著地，即回起始狀態。

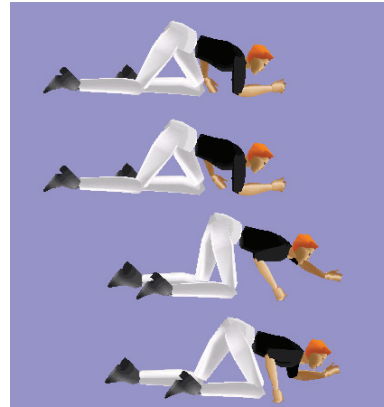
4.2.2. 關鍵格設計

所有關鍵格的 3D 截圖如圖七所示。

- **關鍵格一：**跪姿伏地。重心手(即前手)的手肘與地呈垂直，兩腿成跪姿，兩膝相距為步伐寬度。圖中左手為重心手、右手為移動手，反之左腳為移動腳、右腳為重心腳，如圖六(a)。紫色為背脊，紅色代表右側，綠色代表左側；較短的是手臂，較長的是大腿。頭、小腿、手掌和腳掌因非移動重點，故圖中略。(向左為正)。
- **關鍵格二：**移動手正在往前伸，移動腳準備離地，如圖六(b)所示。
- **關鍵格三：**移動手手腕著地，重心手手肘準備離地，如圖六(c)所示。
- **關鍵格四：**移動手手肘著地，移動腳膝蓋準備著地，如圖六(d)所示。

4.2.3. 關鍵格之間的內差方法

- **第一階段的內差方法：**第一關鍵格與



圖七：跪姿爬行的所有關鍵格

第二關鍵格之間的內插，即為移動手往前伸，直到移動腳開始移動的瞬間；實為直到手腕著地的部份內插。

- **第二階段的內差方法：**第二關鍵格與第三關鍵格之間的內插，在移動手繼續往前伸，以重心腳膝蓋為支點、骨盆則向前畫弧做位移，使移動腳往前伸；並隨著骨盆位置的改變計算背脊角度，使動作合理。
- **第三階段的內差方法：**第三關鍵格與第四關鍵格之間的內插，在移動手手肘放下準備著地成為新的重心手；重心手手肘離地準備回起始狀態並成為新的移動手，骨盆繼續移動使移動腳繼續往前伸。
- **第四階段的內差方法：**第四關鍵格與第一關鍵格之間的內插，以新的重心手手肘為支點、兩手曲肘以回起始狀態，骨盆繼續移動直到移動腳膝蓋著地。

5. 過渡動作設計

除了上述兩個動作的程序設計外，如果要讓這些動作能順利的連接起來，需要設計適當的過渡動作，而這些動作也可以視為特殊的動畫程序。例如，站姿到登上梯子，需要有一段過渡動作；另外，臥倒



圖八：由站姿到上梯子的所有關鍵格

可以視為從站立到爬行的特例過渡動作。

5.1. 由站姿到上梯子

5.1.1. 動作階段分析

- **階段一：**伸出雙手、分別搭在梯子的兩側。
- **階段二：**將移動腳踩上第一階階梯，並改變成為重心腳。
- **階段三：**打直重心腳並將移動手上移一個梯距。

5.1.2. 關鍵格

- **關鍵格一：**站直。
- **關鍵格二：**雙腳著地，雙手同高分別搭在梯子的兩側。
- **關鍵格三：**單腳著地，另一隻腳踩在第一階階梯上，雙手同高分別搭在梯子的兩側。
- **關鍵格四：**雙手抓住梯子兩旁，重心手在上、移動手在下，重心腳打直、移動腳則曲膝懸空；即為上梯子的第二關鍵格。

在圖八中我們展示這個過渡動作的所有關鍵格。

5.1.3. 關鍵格之間的內差方法

- **第一階段的內差方法：**第一與第二關鍵格之間的內插，雙手移動軌跡是以肩膀為圓心、手臂長為半徑畫出的圓弧。



圖九：臥倒的所有關鍵格

- **第二階段的內差方法：**第二與第三關鍵格之間的內插，目的為將移動腳腳板位置移至第一階之階梯位置。內插過程中，需考慮腳板若直接上移，則容易與階梯產生碰撞，因此上移時的軌跡為拋物線。
- **第三階段的內差方法：**第三與第四關鍵格之間之間的內插，目的為將整個身體往上撐、並使所有關節組態與上梯子的第二個關鍵格相同，以便與上梯子的動作連接。

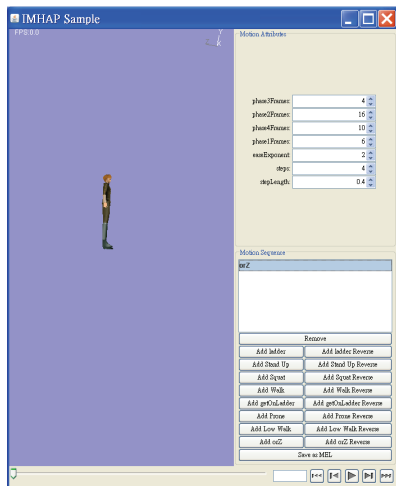
5.2. 臥倒動作的設計

5.2.1. 動作階段分析

- **階段一：**右腳向前踏一步。
- **階段二：**跪下。
- **階段三：**彎腰使兩手著地、並使右膝蓋著地。
- **階段四：**伸左手、使左手手肘著地，轉右腳腳踝、使腳背貼地，一邊開始將左腳往前伸。
- **階段五：**伸右手、使右手手肘著地，繼續伸左腳。

5.2.2. 關鍵格設計

- **第一關鍵格：**站直。
- **第二關鍵格：**身體直立，雙腳一前一後、右腳在前左腳在後。
- **第三關鍵格：**上半身直立，下半身為單腳跪姿、左膝蓋著地。
- **第四關鍵格：**四肢著地，雙腳腳尖、膝蓋、雙手手掌著地，右腳在前、左



圖十：動畫產生平台的人機介面

腳在後、雙手平行。

- **第五關鍵格：**左手在前、手肘著地，右手在後、手掌著地，右腳為重心腳、腳背貼地，左腳為移動腳、腳尖著地。
- **第六關鍵格：**右手在前左手在後、雙手手肘著地，左腳在前右腳在後、雙腳腳背貼地。

在圖九中我們展示這個過渡動作的所有關鍵格。

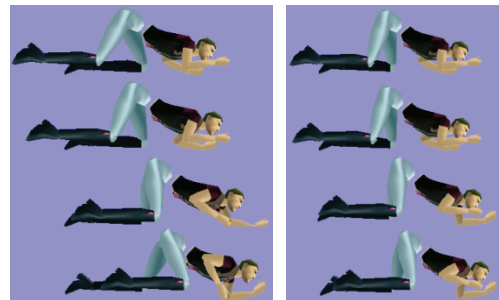
5.2.3. 關鍵格之間的內差方法

- **第一階段的內差方法：**第一與第二關鍵格之間的內插，骨盆軌跡是以左腳為圓心、腿長為半徑畫出的圓弧；右腳腳跟軌跡為貝茲曲線。
- **第二階段的內差方法：**第二與第三關鍵格之間的內插，將骨盆下移，直到左膝著地。
- **第三階段的內差方法：**第三與第四關鍵格之間的內插，骨盆軌跡是以左膝為圓心、大腿長為半徑畫出的圓弧，一邊彎腰使雙手手掌著地。
- **第四階段的內差方法：**第四與第五關鍵格之間的內插，骨盆軌跡以右腳為圓心、大腿為半徑畫出的圓弧，左手



(a) (b)

圖十一：設定不同梯距所產生的動畫結果 (a)大梯距 (b)小梯距



(a) (b)

圖十二：設定不同爬行距離所產生的動畫結果 (a)大步距 (b)小步距

一邊往前伸使手肘著地，左腳也向前伸、同時轉動腳踝。

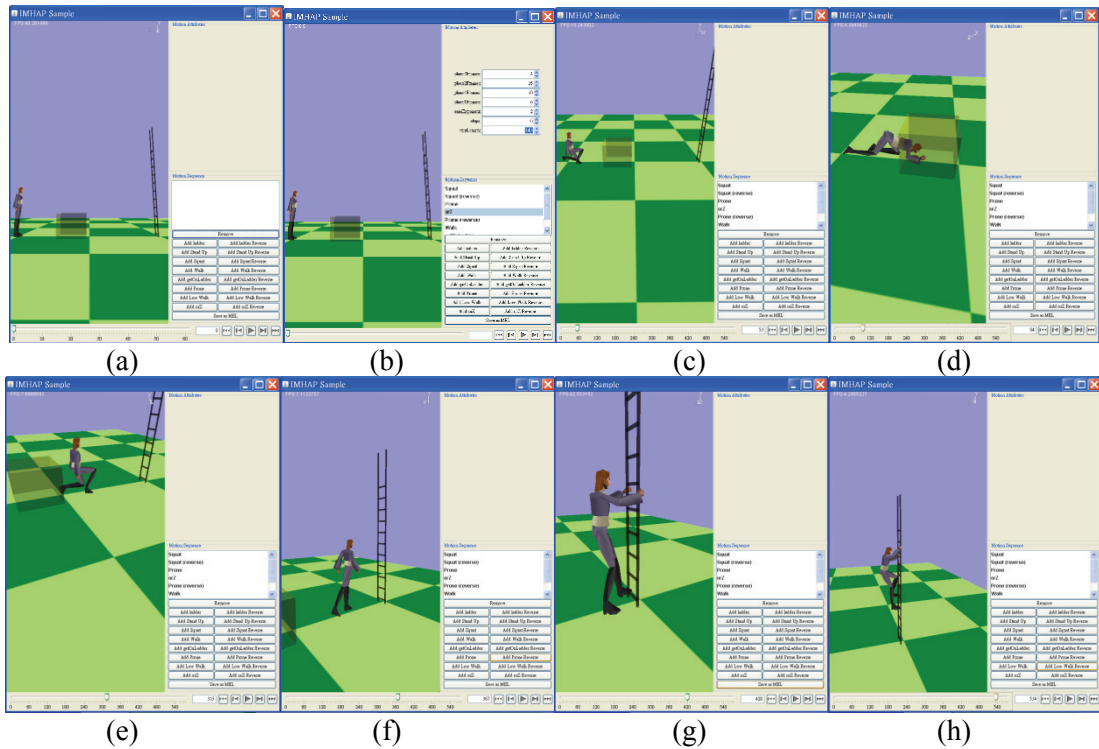
- **第五階段的內差方法：**第五與第六關鍵格之間的內插，骨盆軌跡以右腳為圓心、大腿為半徑畫出的圓弧，右手一邊往前伸使手肘著地，左腳也向前伸到定點；使得最終所有關節組態與跪姿爬行的第一個關鍵格相同，以便與跪姿爬行的動作連接。

6. 實驗結果與成果展示

6.1. 人機介面設計

我們所設計的動畫產生系統是建立在IMHAP的人體動畫實驗平台上，此系統的使用者介面視窗如圖十所示，左側為3D動畫的展示，右側則為動作參數的指定視窗。使用者可以調整每個動作的參數，以獲得類似但不完全相同的動作模擬。

6.2. 爬梯子與爬行動作的動畫結果



圖十三：整體運動的規劃與展示流程圖

我們首先以爬不同梯距的梯子為例說明自動產生爬梯子動作的結果。我們依序設定較大與較小的梯距參數，所產生的動畫結果如圖十一所示。

第二個例子是以爬行的動作的變化為例。我們依序設定較大與較小的爬行距離參數，所產生的動畫結果如圖十二所示。

6.3. 整體運動的規劃與展示

在本節中，我們將以一個綜合的例子說明四肢協同運動的程序式動畫產生方式。在這個例子中，我們指定虛擬演員先臥倒後爬行過一個通道，站起來後爬上一個梯子。在系統的使用上，我們載首先入人體模型與環境資訊，如圖十三(a)所示；在此方格地板的環境中，有兩個障礙物限制虛擬角色的活動，分別為一個半透明的箱形障礙物以及梯子。第二步，由右側面板自訂運動規劃並調整參數，使得虛擬角

色能夠以適當的動作通過障礙物，如圖十三(b)。接著開始播放所指定的動畫：圖十三(c)為臥倒的關鍵格之一，圖十三(d)為以跪姿爬行穿越箱形障礙物的片段，圖十三(e)為由臥倒起身的片段、圖十三(f)則為走向梯子、接著圖十三(g)為攀上梯子的關鍵格之一，最後圖十三(h)則為爬梯子的片段。

7. 結論與未來延展

本研究特色，並非在於強調模擬出來的動作之擬真程度，而是在於產生動畫的靈活性；即依據不同參數的輸入，可計算產生類似但不完全相同的動作，如爬行梯距不同的梯子、依據路徑長短調整步伐數等。

未來我們希望能再加入人工智慧之技術，使系統能夠在給予環境以及起點和終點後，自行計算找出最短路徑，並依照環境限制進行適當的動作。例如，這類技

術可以應用於障礙賽的模擬或礦坑逃生的模擬等，其中虛擬人物將可依照障礙物的高低形狀，自動判斷需以何種運動方式越過該障礙物。

此外，我們希望能結合虛擬人物的心理及體能等個人因素，讓每個虛擬人物在相同的環境條件下，能因為個體的狀態不同而有不同的反應。最後我們將增加人機界面的互動性，使得使用者能夠即時的改變環境，而虛擬人物也能立即的反應產生適當的動作。

8. 參考文獻

- [1] A. Bruderlin and T.W. Calvert, "Goal-directed animation of human walking," in *Proceedings of ACM SIGGRAPH*, 23(4):233-242, 1989.
- [2] A. Witkin and M. Kass. "Spacetime Constraints," *Computer Graphics*, 22(4):159-168, August 1988.
- [3] A. Witkin and Z. Popovic, "Motion warping," in *Proceedings of SIGGRAPH 95*, pp.105-108, Addison Wesley, August 1995.
- [4] P.F. Chen, *Real-Time Planning for Humanoid Lower Body Motion*, Master Thesis, Computer Science Department, National Chengchi University, 2003.
- [5] T.Y. Li, and C.H. Liang, "Simulating Human Low-Posture Motions with Procedural Animation," in *Proceedings of 2007 Computer Graphics Workshop*, 2007.
- [6] T.Y. Li, M.Y. Liao, and P.C. Tao, "IMNET: An Experimental Testbed for Extensible Multi-user Virtual Environment Systems," *ICCSA 2005*, LNCS 3480, O. Gervasi et al. (Eds.), pp. 957-966, 2005.
- [7] T.Y. Li, P.F. Chen, P.Z. Huang, "Motion Planning for Humanoid Walking in a Layered Environment," in *Proceedings of the 2003 International Conference on Robotics and Automation*, September, 2003.
- [8] VRML, <http://www.web3d.org/x3d/specifications/vrml/>
- [9] Y. Koga, K. Kondo, J. Kuffner, and J.C. Latombe, "Planning motions with intentions," in *Proceedings of SIGGRAPH'94*, pp.395-408, 1994.
- [10] H-Anim, <http://www.h-anim.org>