

以運動計畫演算法則自動產生傳統皮影戲動畫

許書璋
政治大學資訊科學系
g9204@cs.nccu.edu.tw

李蔡彥
政治大學資訊科學系
li@nccu.edu.tw

摘要

皮影戲是我國十分具特色但未能數位化的傳統民間藝術。本論文中，我們嘗試設計一個使用機器人學中運動計畫演算法的動畫系統，根據使用者的高階輸入，自動產生出皮影戲動畫。皮影戲中一個影偶的運動，除了有意圖的主要運動外，還包含了被動的次要動作。我們以 RRT-Connect 運動計畫演算法來自動產生主要運動，並設計了兩個方法來減少產生次要運動所需費的功夫。針對影偶上半身手臂的部分，我們讓運動計畫器在產生運動軌跡同時，盡可能的帶有我們偏好的運動樣式。而對下半身部分則是以後處理的方式，將下半身的運動軌跡套用單擺物理模型，加上碰撞偵測修正，產生出真正皮影戲中影偶下半身的晃動及順應地形的效果。

關鍵詞：運動計畫、隨機擴展樹、皮影戲動畫、次要運動。

Abstract

Shadow play is a typical form of Chinese art that is not yet preserved in a digital form. In this paper, we attempt to develop a shadow play animation system that utilizes motion planning algorithms to generate Chinese shadow play animation automatically according to user's high-level input. The motions of a character in a shadow play show usually consists of intended primary motion and reactive secondary motion. We propose to use the RRT-connect algorithm to generate primary motions and design two methods to facilitate the generation of secondary motions. For the character's upper body, we try to bias the search of the planner to generate a path with our preferred motion pattern as much as possible. For the character's lower body, we modify its motion path by simple pendulum physical model and some collision correction mechanism in a post-processing step. With this method, we are able to generate realistic reactive and complaint motions for a character in a shadow play animation.

Keywords: Motion Planning, Rapid-exploring Random Tree (RRT), Chinese Shadow Play, Secondary Motion

1. 前言

皮影戲(如圖 1 所示)是中國傳統藝術中一項富有特色及趣味性的表演。在皮影戲的演出過程中，操縱影人(即皮影戲偶)的方式，是透過影人身上的

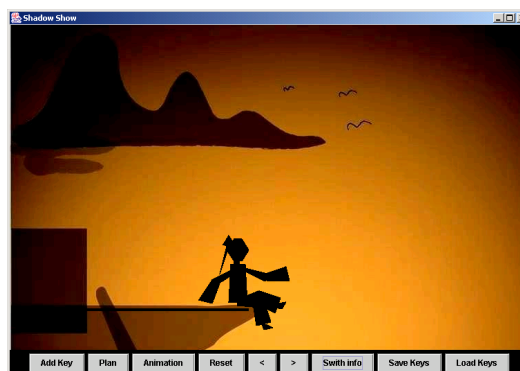


圖 1：以電腦產生的皮影戲動畫

操縱桿，來變換位置或轉動關節，藉此帶動影人肢體的變化，演出各種不同的動作[10][11]。另外，在皮影戲演出過程中，通常也會在影窗上布置一些景，例如桌子、椅子、小橋等等。由於場景是平面的，這些場景中的物品，在演出的過程中，可能會和影人發生重疊的現象；例如，影人走過一張椅子旁邊，或從一棵樹旁邊經過。但也有些情況，影人和場景中的物品不能重疊在一起；例如，影人從小橋上方走過，或坐到一張椅子上。這種需要避開碰撞的操作，如要在電腦動畫中產生，是一件很繁瑣耗時的工作。然而，我們認為這個產生影人運動的問題與人工智慧學中的運動計畫(Motion Planning)問題[3]十分相似。因此，我們設計了一個動畫系統，嘗試用運動計畫的方式來自動產生並模擬皮影戲的動畫。

一般而言，影人的動作可以分成兩類：身體的主要運動(Primary Motion)及肢體的次要運動(Secondary Motion)。主要運動通常是為了表現出影人某些高階的意圖。隨著意圖的難易及其背後所要表達的知識的多寡，產生動作時所需要指定的資料量也不同。譬如說要讓影人從一個地點走到另一個地點，理論上只需要指定起點和終點的動作就可以了，但如果說要讓影人表現出類似在練拳的動作時，就還是必須找出一連串練拳動作中的關鍵動作，再把每一段動作串連出來，這樣看起來才會像是在練拳。而次要動作指的是被動性的運動。例如，在皮影戲中，走路時影人的手會前後的擺動，而下半身則是隨著移動會出現晃動效果，以及下半身可以順應著地形而改變等等，都算是次要運

動。不過一般運動計畫器並不易產生帶有次要運動的主要運動，而往往需要靠使用者輸入許多中間的關鍵格，並將問題切割成一連串小的運動計畫問題，來產生這些效果。

為了減少我們需要輸入的關鍵格數目，我們嘗試使用兩個方法來自動產生這些皮影戲中需要的次要運動。在影人上半身部分，我們修改了運動計畫器，使它在產生運動軌跡的同時，能讓特定關節帶有特定的運動樣式(motion pattern)，像是週期性的運動(cyclic motion)等等。把這個機制套用在影人的手上，就能夠在行走時，雙手就會自動的擺動。另外，我們希望所加入的機制只是偏好，而不是限制。換言之，如果影人目前所處的地方，雙手剛好是無法前後擺動的，運動計畫器會放棄這個偏好，直到影人所在的地方是又可以允許雙手擺動時，再繼續使用這個產生運動軌跡的偏好。我們所採用的方法，是將偏好的運動樣式以狀態(states)及規則(rules)的模型來表示，讓運動計畫器在搜尋路徑的同時，參考這些偏好的模型來產生運動軌跡。

關於下半身的次要運動，我們針對運動計畫器產生出來的運動軌跡，將影人下半身的動作重新處理。主要是透過單擺的物理模型，讓影人在移動過程中，下半身會產生晃動的現象，以及遇到障礙物時採用搜尋的方式，修正碰撞的情況，讓下半身同時可以順從障礙物的表面產生動作的變化，而不會陷入障礙物中。我們稱這種運動為「順從運動(Compliant Motion)」。

2. 相關研究

角色動畫(Character Animation)產生的方式，粗略可以分為三種，一種是以程序性(Procedural)的方式來產生角色動作，也就是根據所要產生動畫的角色的特性，使用不同的運動模型，加上控制演算法，計算出角色人物的動作。像是[1]使用了物理模型來模擬人在跑步、騎單車及跳馬的動作。這種方式的優點是擁有較多對角色運動的控制能力，缺點是要做到逼真較為困難。第二種則是以樣本資料為基礎來產生動畫的方法。例如，將運動抓取(motion capture)資料直接套用在動畫角色身上，或是銜接片段的運動抓取資料以產生動畫。這種方式的優點是容易產生逼真的動畫，而缺點在於運動的重用性較低，且較難用在人以外的角色上。第三種方式則是由動畫師以電腦動畫軟體設計動畫中的關鍵格，再將關鍵格與關鍵格之間作內插，以產生動畫。這種方法的缺點則是需要經驗豐富的動畫設計師，且製作成本昂貴。在[8]這個研究中，也是利用電腦來產生皮影戲動畫，但其目標主要是將皮影戲利用電腦呈現出來，而在動作產生上則是採用動畫設計軟體來製作的。

本研究的目標是以運動計畫法則，自動產生皮影戲偶的運動。運動計畫的技術常被用在機器人學的領域，主要目的是替機器人或機器手臂，從我們

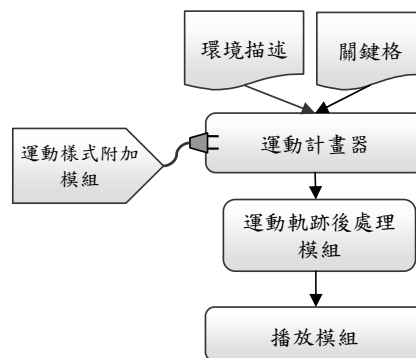


圖 2：系統架構圖

指定的起始狀態到目標狀態，規劃出一條不會與環境中障礙物碰撞的路徑，使得機器人或機器手臂可以順利的移動。此外運動計畫所發展出來的技術也被廣泛的應用到電腦動畫的領域上。在[9]中，使用運動計畫方式，產生用手拿起眼鏡並戴上這種精確且有目的動作。[7]也是利用運動計畫的方式產生出物體運動軌跡，不同的是他在過程中也同時將機構學與動力學的限制同時考慮進去，因此產生出來的路徑能夠符合所設定機構動力學的限制。在[5]中，則是透過隨機街圖(Probabilistic Roadmap)[4]這種運動計畫方式，替一個具有 22 個自由度的虛擬人物產生抓取東西的動作。在運動計畫器規劃運動軌跡的過程中，也會 bias 產生的街圖，讓產生的路徑產生出所偏好的抓取物體的姿勢。

3. 系統概觀

圖 2 是本論文所提出之皮影戲影動畫產生器的運作流程圖。若我們想要產生一段皮影戲的動作，需要輸入兩種資訊。第一個是描述皮影戲演出環境的檔案，這個檔案包含了影偶的描述(或稱 robot)、環境中有哪些物品(或稱 obstacle)。另外一個輸入則是關鍵格。由運動計畫器先根據演出環境的描述將影人與場景中的障礙物設置好，然後計算出關鍵格與關鍵格間的運動軌跡，產生出影人的動作。在運動計畫器產生運動軌跡的過程中，我們可以透過運動樣式附加模組，讓產生出來的動作軌跡，帶有特定的樣式。在動作軌跡產生出來之後，及傳給負責呈像的模組來播放之前，我們會讓運動軌跡再經過一些後處理，以讓整個動作看起來更像是真實的皮影戲演出。在第四節中，我們會先簡介 RRT-Connect 演算法，我們是用這個演算法來實做運動計畫器。在第五節中，我們將詳細的敘述我們是如何建立環境的模型，以利用運動計畫器來產生動作及具有我們所偏好樣式的運動軌跡。第六節我們將介紹如何透過一些後處理的程序修改影人的下半身運動軌跡，以讓整體動畫的表現更能出顯出皮影戲的韻味。

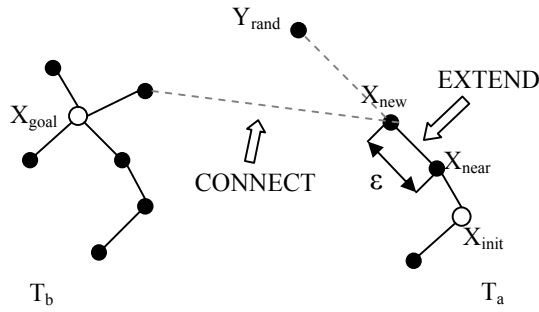


圖 3：RRT-Connect 演算法運作原理。T_a、T_b 為兩顆不同 RRT，不斷的透過 EXTEND 與 CONNECT 兩個程序，使兩顆 RRT 連結在一起。

4. RRT-Connect 演算法

快速擴張隨機樹(Rapidly-exploring Random Tree, RRT)[6]是一種能夠在空間中快速擴展的樹狀資料結構，而 RRT-Connect 演算法[2]則是被設計用來連結兩顆 RRTs，以達到運動計畫連接兩個組態的目的。這個演算法的精神是讓兩顆 RRTs 不斷的擴張自己和嘗試與對方連結，以找出路徑。以圖 3 簡單說明了 RRT-Connect 的原理。圖 3 中，T_a、T_b 代表兩顆不同 RRT，T_a 的 root 為 X_{init}，T_b 的 root 為 X_{goal}。首先由 T_a 進行 EXTEND 的動作，目的是要讓 T_a 向外擴展。EXTEND 的步驟是先在自由組態空間中任意找一個組態 Y_{rand}，做為 T_a 擴張的參考點，接著讓 X_{near} (T_a 中與 Y_{rand} 最近的節點)向 Y_{rand} 擴展一段的距離 ε，長出新的節點 X_{new}。接下來換 T_b 進行 CONNECT 步驟，CONNECT 的目的是嘗試讓 T_b 連結上 T_a。以 X_{new} 為參考點，讓 T_b 持續向 X_{new} 這個節點做 EXTEND 的動作。如果成功連上 X_{new}，代表找到一條從 X_{init} 到 X_{goal} 的路徑。如果失敗，我們將 T_a、T_b 互換，讓 T_b 進行 EXTEND，T_a 進行 CONNECT 的步驟。之後如果又失敗，則再進行交換的動作，直到交換的次數超過我們設定的上限，便宣告路徑搜尋失敗。

5. 產生皮影戲偶的動作

5.1 影人與場景

我們將影人視為運動計畫中的機器人(robot)，而和影人同一層的物品則視為運動計畫中的障礙物(obstacle)，當規劃影人動作時，必須避開這些障礙物。我們設計的影人有 12 個自由度(圖 4)，目前暫時先不考慮多個影人之間互動的情況，因此只在場景中放置一個影人，但可以放置多個物品。在系統架構中的環境描述檔，即包含了影人及場景的定義。

5.2 以運動計畫器產生運動軌跡

我們利用 RRT-Connect 演算法來實做運動計畫

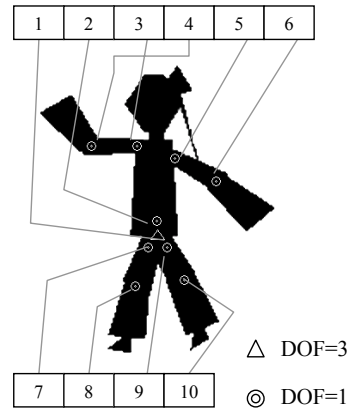


圖 4：影人的構造，方塊中的數字為關節編號，第一個為影人中心，有三個自由度(x,y,θ)，其他關節都只有一個自由度(θ)。

器。在產生動作之前，必須給定影人從動作開始到動作結束過程中的關鍵格組態，這些組態可以依動畫指定的需要從兩個到數個不等。我們的目標是利用運動計畫器，將兩兩相鄰的組態之間如何移動的軌跡規劃出來，最後再把每一段串連起來，讓影人可以從起始位置組態移動到終點位置組態，並且不會和場景中被定義為和影人同一層的物品發生碰撞。這裡我們需要中間組態的原因和傳統動畫在製作時需要關鍵格是一樣的，因為如果我們單純指定起始結束的動作，是可以產生中間無碰撞的動作，但是這個動作可能看起來不像是一般影人會演出的動作。

5.3 產生帶有週期性的運動軌跡

如同之前提過的，我們希望運動計畫器在產生整體運動軌跡時，能夠在我們指定的特定自由度上，加入具有特殊動作樣式，例如手臂擺動這一類的週期性運動，以減少我們需要輸入的關鍵格。我們使用狀態之間該如何變化的規則來描述這些我們希望附加在某特定自由度上的週期性運動。以讓手臂來回擺動為例，我們可以用兩個狀態及兩條規則來表示。這兩個狀態分別是向前擺動和向後擺動。兩條規則的第一條是如果目前是向前擺動的狀態，檢查是否到了向前的極限，就換成向後擺動的狀態，否則繼續向前擺動。第二條規則相類似，只是狀態剛好相反。

在週期性運動的動作樣式模型建立之後，我們需修改 RRT-connect 演算法，讓它可以產生帶有我們偏好的動作樣式的運動軌跡。我們需要改變的地方有兩個，第一是讓 RRT 中節點的組態，在每個自由度都加上 state 這個屬性。

第二個是修改 RRT-connect 中的 EXTEND 步驟，我們的想法是讓 RRT 在 EXTEND 時，根據我們偏好的動作樣式，來 bias EXTEND 產生出來的新

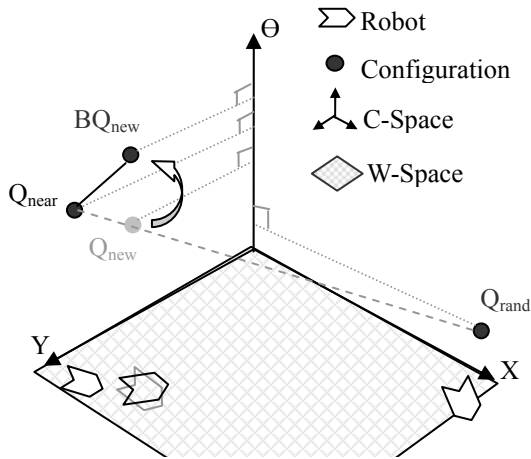


圖 5：Bias EXTEND，產生我們偏好的運動軌跡

節點。我們以圖 5 中一個具有 3 個自由度的機器人來說明我們是如何修改的。假設圖 5 的 3D 座標代表的是機器人的組態空間(C-space)，而我們直接把 XY 平面當成機器人的工作空間(Workspace)。我們希望機器人在 θ 這個自由度上，套上來回擺動的週期性運動。原本 EXTEND 是讓演算法中選出要擴展出去的節點 Q_{near} ，直接沿著 Q_{rand} 延伸的一段距離 ϵ ，產生 Q_{new} 。新的作法則是，如果我們希望我們的動作樣式的模型是套在 θ 這個自由度上，那我們就毋需使用內差的方式來決定 θ 的值，而是根據 Q_{near} 的狀態和所套用的規則，來決定的 θ 該如何變化。譬如圖 5 中，原本 EXTEND 應該產生的是 Q_{new} ，但是根據當時 Q_{near} 的狀態及該狀態下所套用的規則，發現 θ 值是要偏好要增加的，於是就去 bias Q_{new} 的 θ 值，產生 BQ_{new} 這個節點。如果我們發現 bias BQ_{new} 後會發生碰撞的情況，我們就放棄使用 BQ_{new} ，重新使用原本線性內差的方式來產生 Q_{new} 。如此一來，在 θ 這個自由度，就會盡可能的帶有我們想要的週期性運動的運動樣式了。

在皮影戲動畫系統的實做裡，我們是將影人的 3、4、5、6 這四個關節的自由度套上週期性，不同的是，我們僅用兩個狀態與兩條規則控制這四個自由度，因為 4、6(下臂) 和 3、5(上臂) 是有相依性的，當上臂向前擺時，下臂也會跟著向前擺，反之亦然。所以我們只需要兩個狀態與兩條規則(如 5.3 節第一段所描述)，就能夠讓影人的上半身在行走時，產生手臂擺動的週期性了。

6. 修飾皮影戲偶下半身運動軌跡

除了上半身的週期性運動外，我們為了讓影人的下半身的動作有更逼真的視覺效果，也對下半身的運動作了一些處理。我們主要的方法是以單擺的模型來模擬影人的下半身動作；不過，單擺的模型所產生的動作有時會讓下半身和環境有碰撞的情況發生，我們也針對這種情況作修正。

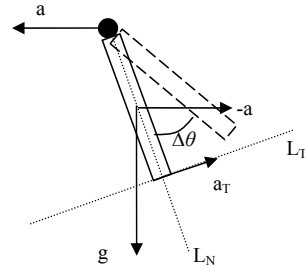


圖 6：以單擺模型重新計算下半身運動軌跡

6.1 以單擺模型重建影人下半身運動軌跡

在皮影戲中，影人的雙腳通常是沒有裝操縱桿，也就是說雙腳在操作影人的過程中會有相對運動的情況產生。如果影人沒有貼著地面，當身體向前移動時，雙腳會微微後彎，而當身體停下時，雙腳自然下垂並晃動。由於運動計畫器產生出來的運動軌跡並沒有這樣的特色，我們藉由後處理的方式，讓影人的動作具有這種特色。

對於下半身的自由度，我們放棄了由運動計畫器算出來的值，而是利用單擺模型，加上一些簡化，來修改我們下半身的運動軌跡。此外，雖然我們以後處理方式修正下半身動作時，將同時放棄了運動計畫器所計算出來的下半身軌跡，但是我們在第一個步驟進行運動計畫時仍然需要將下半身的運動也考慮進去，因為以上半身為主找的運動路徑，可能無法靠後處理修正的方式，便能找到合法的全身路徑。如果將下半身先放進去計畫，可以預留下半身的空間，當下半身重新套上單擺模型所產生的動作時，也才不會顯的太奇怪。

把圖 6 上的黑點當成是影人的關節 7，白色長條當成是影人的腿部上半部。我們以兩個組態間關節 1(影人中心點)移動的距離，根據有限差分(Finite Difference)的方法計算出關節 7 應該轉動多少。

$$\Delta S = V * t + 1/2 * a (\Delta t)^2 \quad (1)$$

首先，我們必須先求出從這次的組態移動到下次的組態後，切線加速度 a_T 的值會變成多少。我們假設從組態移動到下一個組態所需的時間是常數 ΔT ，而且是等加速運動，帶入(1)得到

$$\Delta S = V * \Delta T + 1/2 * a (\Delta T)^2 \quad (2)$$

先考慮從第一個組態移動到第二個組態，由於一開始的初速 $V=0$ ， ΔS 為影人在工作空間中移動的向量長度，透過(2)可以推出加速度 a 。 a 是作用在關節 7 上的，所以對腿部來說，等於受了一個 $-a$ 的加速度，再加上重力加速度 g 的影響，根據圖 3 力的分解，我們可以算出切線方向的加速度 a_T 。

$$a_T = r a. \quad (3)$$

再來，我們要計算 a_T 會影響腿部擺動的角度(θ)為何。由(3)得知 a_T 與角加速度 α 成正比，為了簡化我

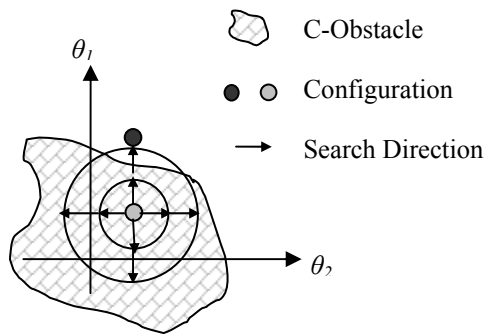


圖 7：以環狀方式慢慢擴張，搜尋出一個離原本被困在 C-obstacle 中的組態最近的新組態。

們也把 r 當成 1。因此，可以得到 $a_1 = \alpha$ 。

$$\Delta\theta = \omega * \Delta T + 1/2 * \alpha (\Delta T)^2 \quad (4)$$

有了 α 後，假設一開始角速度 ω 也為 0，加上時間為 ΔT ，即可推出 $\Delta\theta$ 。把 $\Delta\theta$ 加到關節 5 上，便完成了這次的計算。過程中需要維護 V 和 ω 的值，之後重複上面的步驟即可。我們把這種單擺的模型套用在影人身上的 7、8、9、10 這四個關節的自由度上，這時候影人的下半身，就會隨著身體的移動，而產生出晃動的效果。

6.2 修正下半身軌跡的碰撞

在上節中以單擺來模擬下半身動作時，我們暫時忽略和環境中的障礙物發生碰撞的情形，但實際上當碰撞情形發生時，我們需要修正下半身的擺動以避免碰撞。我們以圖 7 來說明修正的方法。我們將影人的雙腳分開處理，每隻腳有兩個自由度，分別是 θ_1 及 θ_2 。我們檢查經過單擺模擬後的組態，如果其中任一隻腳有碰撞情形，就代表該隻腳的組態 (θ_1, θ_2) 是落在 C-Obstacle 裡面的。我們從 θ_1 和 θ_2 組成的組態空間，以原先的組態為圓心，逐次向外擴展，搜尋合法的組態，以替換目前不合法的組態。這樣的目的是希望找一個離碰撞組態最近的合法組態，以避開碰撞。經過這樣的處理，當影人的下半身可以和障礙物接觸時，看起來的效果是影人會隨著地形和身體的起伏，貼在障礙物上面，而不會陷進去。

7. 結果分析與討論

首先我們對有無使用週期運動來產生影人上半身動作作了一些比較。實驗的環境為圖 9(a) 的場景。我們比較了三組，第一組我們只用 2 個關鍵格，第二組用 7 個(中間五個目的是為了產生雙手擺動的效果)，第三組也是用 2 個，但是套上自動手臂的週期性運動的模組。圖 8 是產生出來的軌跡中，關節 5 的自由度的變化值(影人左手手臂)。我們可以清楚的看出第二組和第三組的變化效果類似，而第一組沒有使用中間關鍵格也沒有套用週期性運

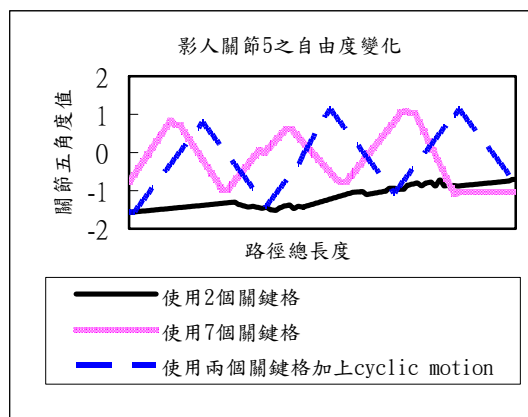


圖 8：三種不同方法所產生的自由度變化

動，所以無法產生讓手臂擺動的現象。

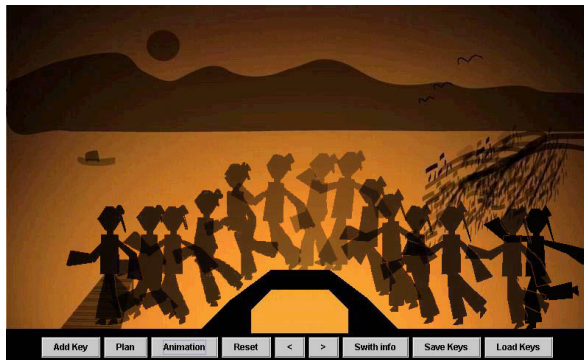
再來我們對修改後的演算法是否能夠適合的環境下，盡量採用我們設計的運動樣式偏好做了一個比較。首先在圖 9(a) 中，地面和橋是屬於和影人同一層的障礙物，剩下的皆為背景。我們的目標是產生一段影人走過橋的動畫。在移動過程中，並沒有障礙物阻擋到手的運動，所以從頭到尾，手臂都能夠持續的擺動。而在圖 9(b) 中的動畫呈現的是被抓到妖洞中的影人從鍋子中逃出。我們在鍋子的兩側放了兩個柱狀的障礙物來產生鍋子邊界的效果。影人一開始在鍋子中的時候，由於鍋子內空間不夠，手部無法揮擺，直到出去之後，才開始擺動。從這兩個例子可以看出運動計畫器能根據我們的偏好產生手臂的週期性運動。最後，圖 9(c) 說明了戲偶下半身使用單百模型及適應地形的順從運動。

以使用狀態與規則來輔助運動計畫器在規劃運動軌跡這種方式，能夠讓我們在指定的自由度上加入一些我們喜歡的運動樣式。以我們設計手臂的週期性動作來說，只需要兩個狀態與兩條規則就可以產生，但是如果產生較複雜的運動樣式，就需要比較多的狀態與規則來描述。此外，由於目前沒有考慮時間的維度，所以在規則的設計上，只能根據自由度的值變化到一定的程度，然後跳到另一個狀態，設計這類型的規則。目前系統的實做尚無讓一個狀態持續多久的時間這類的規則，這是目前很大的限制。

另外在第 6.1 節我們提到將全身的自由度都放到運動計畫器中做運動軌跡的規劃的理由。雖然將下半身考慮到運動計畫裡面是必須的，但實際上可以做一些簡化，譬如將腳的兩個關節先簡化成一個，不過由於系統在實做上便利性的考量，這部分並沒有採用簡化的方式來實做。

8. 結論

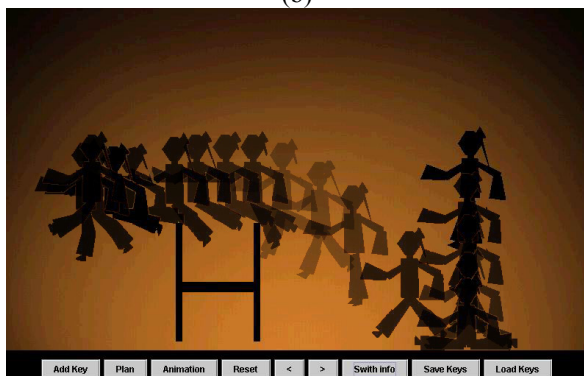
由於傳統藝術皮影戲的特色與趣味性，使我們興起利用運動計畫的方式，來產生皮影戲偶動作，製作出皮影戲動畫的想法。在這個皮影戲動作產生



(a)



(b)



(c)

圖 9：圖(a)為運動過程中，影人的手臂可以一直揮擺。圖(b)為影人一開始在鍋中，雙手無法自由擺動，直到逃出之後才能夠擺動。這兩個例子說明運動計畫器會盡量滿足我們設定的運動偏好。圖(c)表現的是影人的下半身，會順應地形而改變。

系統中，系統能夠自動產生像是手臂揮動、下半身的自然晃動及順應地形這類的次要運動，以減少我們所需輸入的關鍵格。在影人上半身部分，我們利用狀態與規則來描述我們偏好的運動樣式，然後修改 RRT-Connect 運動計畫器，讓它能夠根據我們的狀態與規則去 bias 產生的新節點，讓產生的運動軌

跡能盡量帶有我們偏好的運動樣式。在影人下半身的部分，我們利用了單擺模型，重新計畫影人下半身動作，並且再經過碰撞修正，讓影人的下半身，可以跟隨著移動過程自動產生晃動的效果，以及和地上障礙物碰撞時，能夠順著地形產生出對應的動作，而不會有陷入的情況發生，讓整體看起來更像是真實的皮影戲演出。

9. 致謝

此研究在國科會 NSC 92-2213-E-004-001 計畫的支助下完成，特此致謝。

參考文獻

- [1] J.K. Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brien, "Animating Human Athletics," *Computer Graphics (Proc. of SIGGRAPH)*, pp. 71-78, Aug, 1995.
- [2] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," *Proc. of IEEE International Conf. on Robotics and Automation*, pp. 995-1001, 2000.
- [3] J. Latombe, *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
- [4] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic Roadmaps for Fast Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, 12:566-580, 1996.
- [5] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann, "Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping," *Proc. of Eurographics*, pp. 313-322, 2003.
- [6] S.M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Technical Report 98-11, Computer Science Dept., Iowa State University, 1998.
- [7] S.M. LaValle and J.J. Kuffner, "Randomized Kinodynamics Planning," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 378-400, 2001.
- [8] Y.-B. Zhu, C.-J. Li, I. F. Shen, K.-L. Ma, A. Stompel, "A new form of traditional art: visual simulation of Chinese shadow play," *Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*, 2003.
- [9] Y. Koga, K. Kondo, J.J. Kuffner, and J.C. Latombe, "Planning motions with intentions," *Proc. of SIGGRAPH '94*, pp. 395-408, 1994.
- [10] 邱一峰。台灣皮影戲。1998。漢光文化。
- [11] 秦振安, 洪傳田。皮影戲珍藏圖典。2004。知識風出版社。