

在即時虛擬環境中以攝影學法則開發智慧型攝影機模組 Developing an Intelligent Camera Module with Cinematography in Real-Time Virtual Environments

鄭仲強

國立政治大學資訊科學系
g9401@cs.nccu.edu.tw

李蔡彥

國立政治大學資訊科學系
li@nccu.edu.tw

摘要

攝影機控制是即時虛擬環境中一個重要的研究課題。本研究嘗試在虛擬環境中發展符合攝影學法則的即時智慧型攝影機模組，讓使用者可以專注在化身的操縱上，而不需花額外的心力控制攝影機，便可達到追蹤目標化身的目的。此智慧型攝影機模組的能力包含了規劃不被障礙物遮蔽的跟蹤拍攝，以及當攝影機進入險惡的環境時，自動在兩台攝影機之間進行符合攝影學法則的轉切。此外，此智慧型攝影機模組並能依照不同使用者的偏好，產生不同結果的攝影機規劃。我們實作了此一智慧型攝影機的系統，並以實例說明此系統的有效性。

關鍵詞：動態運動規劃、虛擬攝影機、智慧型攝影機

1. 前言

電影向來是個很好的說故事媒介，有別於其他的媒體，電影最大的特點是透過鏡頭(Shot)來說故事，「透過一連串影像的並置，藉著鏡頭之間的對立在觀眾心底所引起的反應，來將故事往前推衍」(Theory of Montage)。這方面有關操縱攝影機拍攝鏡頭來捕捉並發覺資訊的知識我們稱之為「攝影學(Cinematography)」[1]。過去數十年來，隨著好萊塢電影業的蓬勃發展，攝影學至今已經有相當高的成熟度，且已應用在動畫中攝影機的自動擺設上[9]，然而在電腦虛擬環境中的應用仍然較為缺乏。

隨著近年來硬體與技術上的進步，使得許多3D電腦遊戲成為虛擬環境平台的典型應用。藉由畫面傳達大量的視覺資訊，玩家在遊戲中可以得到近似於現實生活中的體驗；例如，在虛擬環境中進行導覽(Navigation)，發掘場景中的事物並且觸發事件。一般的導覽系統大多使用文字、圖片來介紹，少數的導覽系統使用虛擬實境，讓使用者可以在場景中移動。而在線上遊戲中，如 World of Warcraft、Second Life 等，透過網際網路的連結，更可與其他現實生活中的其他玩家進行如對話、交易或肢體行為等的互動。

然而當使用者到了未曾拜訪的場景，很容易因為虛擬實境所提供的視野範圍太小而迷失了方向。使用者四處移動之後，常常會不認得四周的環

境，搞不清楚自己身在何處。在現有的遊戲系統中，對於虛擬環境中攝影機的操作大多透過搖桿或是使用鍵盤並搭配滑鼠的輔助來完成。我們可將目前常見攝影機系統分成以下兩大類。第一類是第一人稱攝影機。使用者控制虛擬環境中之化身，攝影機直接依附於化身身上，因此能夠看到化身眼中的世界，彷彿使用者本人處於該虛擬環境之中，但是有限的視野範圍也間接限制了攝影機的表達能力，僅能得知化身前方的資訊，對於整個場景的資訊無法立即有效的掌握。第二類則是第三人稱攝影機。攝影機以第三者的角度，隔著適當的距離拍攝虛擬環境中之化身，並且依據使用者與環境的互動來改變攝影機的角度與位置。攝影機位置的變化可能是連續的跟蹤拍攝(Tracking)，或者是在兩個不同的位置進行畫面的轉切。雖然有些研究已著力在為電腦遊戲設計攝影機的控制模組[12]，但較少能在即時環境下對預測的運動進行攝影機移動或轉切的規劃。

結合以上對虛擬環境中自動攝影機規劃的探討，本研究的目標是希望在3D虛擬環境中的場景下，實現具有自動規劃能力的攝影機模組，並且符合攝影學的準則，讓使用者在虛擬環境中操縱化身的過程中，不需要花費額外的心力控制攝影機，降低使用者操縱的複雜度，且能維持理想的拍攝效果。

2. 相關研究

在過去十年間有不少關於虛擬攝影機系統的研究，而根據其解決問題的機制，我們可將系統分成以下三類[8]：

代數式系統：將攝影機規劃的問題轉換為數學代數問題，也就是將攝影學手法轉換成數學表達式，藉此計算出攝影機的位置。此種方法的好處是能夠經由快速的計算得到解，但相對的缺點則是缺少彈性，因為問題的轉換必須依靠預先選定的攝影學手法，因此結果較為死板。相關的研究有[7]等，此研究發展出了一套高階語言 DCCL (Declarative Camera Control Language)來描述各種攝影手法，然後藉由 DCCL 的編譯器轉換成參數，再使用這些參數帶入代數式系統決定攝影機的位置。

條件限制式系統：此類型的攝影機系統將攝影機或畫面的屬性逐一轉換成數值的條件限制式

[2]，藉由尋找符合限制的屬性來規劃攝影機 [3][6]。依照條件限制的種類，還可以分成完整 (complete) 與非完整 (incomplete) 兩種。完整的方法會徹底探索整個搜尋空間，但需消耗較多的時間，然後找出一個最佳解。此類的方法還可以細分成嚴格限制與階層式限制，階層式限制相較於嚴格限制對於限制的條件較為寬鬆，因此在當有部分的條件不符合限制時仍可以找到合適的解。而非完整的方法大多採用隨機取樣的方式，因此找到的解有可能是最佳解或是有誤差的近似解。

運動計畫式系統：運動計畫的技術起源於機器人學的領域，最初的目的是為了替機器或機器手臂於 2D 或 3D 的環境下規劃運動路徑。當我們指定初始狀態以及目標狀態，運動計畫的技術便能夠規劃一條不與障礙物碰撞的移動路徑，讓我們有效率的從一個狀態轉換到另外一個狀態；早期也有人稱之為鋼琴搬運問題 (Piano Mover's Problem) [10]。整體而言，運動計畫的技術主要是將原先於 2D 或 3D 環境底下的問題簡化成 Tree 或 Graph，然後再使用搜尋演算法處理，讓複雜度較高的問題能夠在合理的時間內解決。而除了機器人自動化之外，近年來運動計畫所發展的技术也被廣泛地應用在其他領域上，例如在遊戲中輔助角色的路徑移動提示 [15]，或是和本研究相關的虛擬實境導覽上。

3. 規劃即時攝影機路徑

3.1 問題描述

當使用者在虛擬環境中操縱化身時，必須同時控制化身的移動與攝影機的視角，因此一般的方法會將一個攝影機放置於化身後方適當的角度。攝影機會一直瞄準著使用者操作的化身，因此化身會保持在螢幕中央。然而此類第三人稱的攝影機所需處理的一個問題是如何防止攝影機與虛擬環境中由幾何圖形所構成的障礙物發生碰撞。例如當我們將化身往後移動，最後將背部靠在牆壁上，假如攝影機一直保持在化身後方某個距離，最終將會跨入牆壁之中，導致畫面失真無法忠實呈現畫面，並且傳遞負面的訊息給使用者。

對此問題第一個方法是放棄任何動態攝影機規劃，而是在環境中根據可能的狀況預先架設各種類型的攝影機，每個攝影機負責拍攝地圖上的一塊區域，當使用者進入該區域後便驅動相對應的攝影機進行拍攝。這種方法所花費的代價最小，但缺點則是只能用於已知的靜態環境，若環境動態的物件太多，則無法使用。早期的電腦遊戲，如惡靈古堡，並使用這種方式架設攝影機，但成效並不好，環境中有過多的死角攝影機拍攝不到，玩家亦無法充法得知目前遊戲中的訊息。

第二個選擇是讓攝影機跨越障礙物，但是不讓幾何圖形阻擋攝影機想要拍攝物件，解決的方法便是讓攝影機和化身之間的幾何圖形經過適當的混

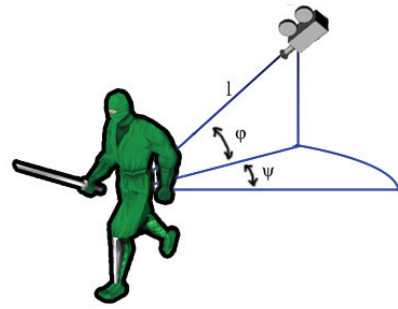


圖 1 攝影機組態的表示方式 (l, ϕ, ψ)

色 (alpha-blended)，也就是障礙物的某些部份會變成透明的。這個方法是傳統上較多人使用，但對使用者而言仍不夠直覺。

第三個方法則是當攝影機即將要跨越障礙物時，就尋找可能的攝影替代位置，然而要找到一個正確的位置並不是一件容易的事。例如我們可以嘗試提高攝影機的拍攝高度來避過牆壁，但是攝影機的俯視角度越接近垂直，所呈現的畫面勢必將損失越多，如果場景中有天花板，則將會產生超越邊界的問題。

我們希望能夠使攝影機自動依附在化身上，即時地調整位置及視角，降低操作的複雜度。規劃即時攝影機路徑的問題中，首先要考量到攝影機不可陷入障礙物之中，而攝影機與拍攝目標物之間的連線也不可遭受障礙物的遮蔽。接著我們還要考量使用者偏好的位置視角，在允許的狀況下盡可能符合這些條件，但這些部分屬於非嚴格的限制 (soft constraint)。另外，由於本目標應用有即時規劃的要求，規劃攝影機路徑可利用的時間遠低於非即時的應用，因此我們必須要有好的搜尋演算法及系統機制，加上適度的預測，才能在時間限制內完成攝影機路徑規劃。

3.2 攝影機組態空間的定義

在本論文中，我們將攝影機控制的問題轉換成攝影機即時運動規劃的問題。在 3D 環境中的攝影機可擁有 6 個自由度的組態：攝影機位置 (x, y, z) 與攝影機的拍攝方位 (roll, yaw, pitch)。但六個自由度的組態空間維度太高，勢必會花費相當多的時間搜尋路徑。但我們認為在此應用中，並非每個維度都是必要的；如攝影機滾動 (Roll) 的行為其實鮮少出現在傳統電影拍攝的過程中，因為這樣的運鏡行為容易使觀眾有喝醉酒的感覺。因此如 [13] 的研究中，便捨棄攝影機拍攝方位的三個自由度，僅由攝影機座標位置構成組態空間，且假設攝影機在任何情況下皆朝著目標物方向拍攝，也就是目標物永遠會出現在畫面的中央，而這樣的假設符合一般虛擬環境的情境。如此一來便可將組態空間降低至三維空間以加速搜尋速度，而到最後的後製階段才根據路徑的狀況加以微調攝影機方位。

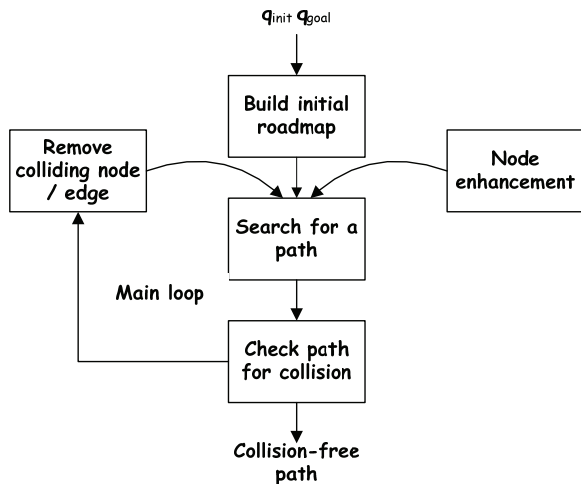


圖 2 惰性街圖(Lazy PRM)演算法運作的示意圖

我們採用類似的概念，僅考慮攝影機位置座標所組成的組態空間，但使用[11]所使用的攝影機相對於化身的表示方法，以化身為原點，組態由攝影機與目標物的距離 l 、水平夾角 ϕ 、垂直夾角 φ 所構成(如圖 1)。這樣的表示法的好處是容易維持攝影機與化身之間的相對關係。

3.3 建立機率街圖

定義組態之後，我們便可以在組態空間中建立隨機街圖(Probabilistic Roadmap)。典型機率街圖的建立方式(Probabilistic Roadmap Method, PRM)是先整個組態空間中散布節點(Node)，合法的節點不可落於障礙物之內，倘若節點與鄰居之間的直線內插不受障礙物遮蔽，便可在節點與節點之間建立一個連結(edge)。在大部分使用隨機街圖的應用中，一旦街圖建立之後，便可將路徑的起始點以及終點加入街圖中，然後使用搜尋最短路徑的方式求得路徑；如果沒有搜尋到路徑，便代表起始點與終點之間未連通，此時便可適當地再增加節點(Node Enhancement)更新街圖，然後再重新搜尋路徑。

在我們的研究中，有別於上述的隨機街圖，我們採用[4][5]所提出的的變形：惰性街圖(Lazy PRM)。惰性街圖規劃流程如圖 2 所示。此方法的最大特點是在組態空間建置街圖時，並不立即檢查所有節點的合法性，而是在路徑規劃的階段才檢查節點是否落於障礙物之內，以及節點與節點之間是否連通。如此一來便不會對搜尋期間未觸及的節點作碰撞檢查，可有效地減少路徑規劃的時間。

另一點不同的地方，隨機街圖一般是建立在整個環境，而我們所使用的街圖是建立於以化身為原點的座標系統中，僅散布於化身周遭。如圖 3 所示，隨著化身的移動，街圖也會改變其在工作空間中的位置，這樣的好處是較容易維持攝影機與化身之間的相對關係，但卻使得我們無法預先知道街圖上結點的合法與否，因此我們採用惰性街圖的精神，在

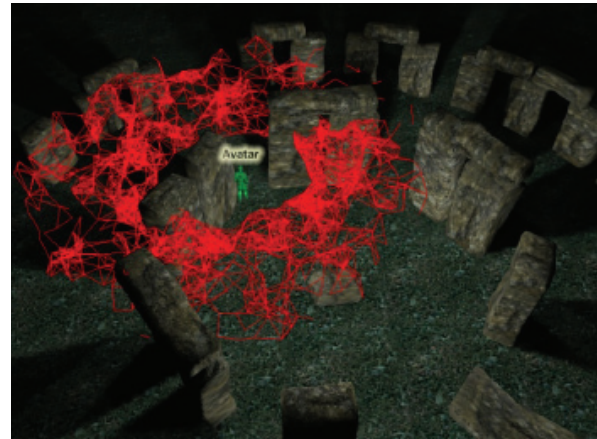


圖 3 以目標為原點所建立的隨機街圖範例

路徑搜尋的階段才檢查節點的合法性。

3.4 攝影機路徑規劃

一般的路徑規劃問題通常包含了兩個輸入資料：起始組態及目標組態，然後輸出從起始組態至目標組態的一連串轉換，也就是移動路徑。由於我們所面臨的狀況是在即時環境下進行路徑規劃，不一定有充分的時間可以保證找到目標組態，因此我們的輸入為一理想目標組態。路徑規劃的目標是在街圖上找尋一個最接近理想目標組態的組態，然後產生出一條路徑，讓起始組態能夠移動至此組態。系統並非無時無刻都在進行路徑規劃，而是當攝影機位於街圖的節點上時才觸發路徑規劃，也就是當攝影機位於街圖的連結(link)上時，並不會進行攝影機的重新規劃。

3.4.1 目標函式

我們希望對於不同使用者的偏好能夠產生不同的攝影機擺設，例如使用者可能喜歡長鏡頭的畫面、相對於化身的特定拍攝角度、或者是偏好使用攝影機的轉切大過於攝影機的跟蹤拍攝。在實作上我們會將以上有可能做為偏好的條件，依據使用者的喜好做適當的加權，以做為路徑規劃時評分的依據，進而產生不同風格的攝影機擺設。

目標函式(Objective Function)是在搜尋演算法中對街圖上每個節點的評分依據。一般而言，我們以使用者所偏好的攝影機組態為依據，在隨機街圖上與使用者所指定的組態擁有越高的相似度，其差異便越小，因此會有較低的分數。目標函式 f 可分為三個部份 f_1, f_2 , 及 f_3 ，以下我們將依序介紹。

$$f_i(q) = w_h \cdot d_h(q) + w_a \cdot d_a(q) + w_r \cdot d(q) \quad (1)$$

- 高度差 d_h ：攝影機相對於拍攝目標物所需的位移高度，與理想位移高度之差。
- 角度差 d_a ：這裡的角度指的是相對於 Line of

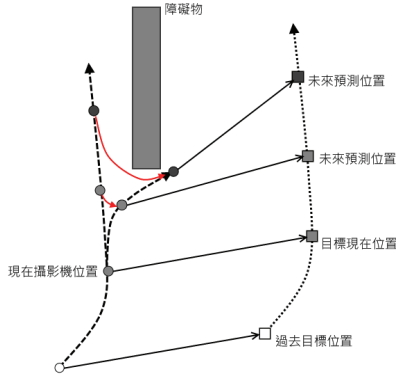


圖 4 根據目標過去軌跡對未來位置進行預測

Interest 的角度與理想角度之差。我們可以依據拍攝方位的不同區分為外部攝影機、平行攝影機以及內部攝影機，讓使用者透過導演模組下達高階的偏好指令，再轉換成實際的理想數值角度。

- 距離差 d_1 : 攝影機與拍攝目標物的距離與理想拍攝距離之差。拍攝距離可以決定目標物在畫面上的大小。

上述三個項目系統會依據導演模組所傳遞的較理想組態，與每個傳進目標函式的組態 q 求出其差距，然後乘上使用者所決定或系統事先定義的權重 (w_h, w_a, w_i)。

$$f_2(q) = C_{occ} \quad (2)$$

f_2 為能見度的成本。該節點上是否能夠拍攝到目標物，若無法則會加上固定的花費分數 (C_{occ})，反之則回傳 0。

$$f_3(q) = \text{distance}(q, q') \quad (3)$$

f_3 為移動距離的成本。移動距離指的是街圖上起始節點 (q') 移動至目前節點 (q) 所需移動的距離。移動距離的權重若調高，則會使得攝影機不容易在街圖上改變位置，因此容易保持與拍攝物的相對位置。在路徑搜尋的過程中，從不同父親節點來訪也會導致移動距離的分數有所不同，因此我們的搜尋演算法也必須如同 Dijkstra 演算法進行節點鬆弛 (node relaxation) 的動作。

最後組合以上三個函式，並且為 f_2, f_3 乘上由使用者或系統所指定的權重 (w_2, w_3)，最後我們可以得到目標函式如下：

$$f(q) = f_1(q) + w_2 \cdot f_2(q) + w_3 \cdot f_3(q) \quad (4)$$

3.4.2 時間預算

為了能夠讓程式的執行保持一定的流暢，避免攝影機規劃時因為消耗太多時間產生停頓。因此我

Algorithm: Time_Budget_Based_Path_Search

```

Let:  $OpenList \leftarrow \{node_{ini}\}$ 
1  $Timer \leftarrow 0; VisitedNode \leftarrow null$ 
2 while  $OpenList \neq \{\}$  and  $Timer < TIME\_LIMIT$ 
3    $n = \text{extractMin}(OpenList)$ 
4   if  $isVisited(n)$ 
5     then continue
6   else  $VisitedNode \leftarrow VisitedNode \cup n$ 
7   foreach neighbor  $N_s$  of  $n$ 
8     if  $isConnected(N_s) = \text{false}$ 
9     then continue
10     $new\_score \leftarrow \text{objective-function}(N_s)$ 
11    if  $score[N_s] \text{ not nil and } new\_score < score[N_s]$ 
12       $score[N_s] \leftarrow new\_score$ 
13       $OpenList \leftarrow OpenList \cup N_s$ 
14    if  $score[N_s] < score(CUR\_BEST\_NODE)$ 
15      then  $CUR\_BEST\_NODE \leftarrow N_s$ 

```

圖 5 時間預算式路徑規劃演算法

們在路徑規劃上加入了時間預算的機制，確保規劃的過程不會消耗太多的資源。在每一次路徑搜尋的過程中，將會有一段限定的時間，路徑搜尋的運作都必須在時間內完成。而路徑規劃也由於時間預算的限制，在有限的時間內只能拜訪街圖有限的節點，我們必須在時間內盡可能拜訪到分數較接近理想目標組態的節點。

3.4.3 拍攝目標未來位置的預測

有別於非即時的虛擬攝影機規劃，在即時環境下往往無法得知未來拍攝的目標物移動軌跡。然而攝影機若無法考慮未來的狀況，往往很難具有智慧的行為表現。為了能夠讓攝影機的移動更加平順，我們必須對未來拍攝目標以及攝影機的位置進行猜測，我們使用的方法是根據過去的軌跡與現在位置，以及拍攝目標與攝影機的速度和加速度，利用線性外插推測出未來 t 時間後的可能位置。當我們發現未來估計的攝影機位置無法拍攝到目標時，便要調整攝影機的位置，讓之後的攝影機軌跡都不受障礙物阻隔。如圖 4 所示，當我們預測到了未來可能有攝影機無法拍攝到目標物的狀況時，就要試圖改變攝影機的位置，好讓未來的軌跡可以拍攝到目標物。

在即時的虛擬環境系統下，我們無法完全猜出使用者未來的移動軌跡，即便是對短時間後的未來進行預測也有可能產生誤差，不過由於實際上是每次規劃便進行一次預測的動作，因此若有誤差產生也能夠很快的被修正。

我們將預測的部分納入目標函數中，街圖上的所拜訪的節點要考慮未來的位置是否能夠拍攝到目標物，若無法則視同被遮蔽，該節點在目標函數中會被加上受遮蔽所花費的分數 (C_{pred})，反之則加上 0。在我們的系統實作中， C_{pred} 使用和 C_{occ} 類似的花費分數計算方式。如此一來在路徑規劃的過程

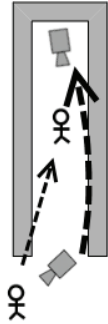


圖 6 攝影機可能陷入狹長死巷的示意圖

中便能夠透過目標函數，考慮到拍攝目標未來位置的預測。更改過後的函式 f_2 如下：

$$f_2(q) = C_{occ} + C_{pred} \quad (5)$$

3.4.4 攝影機路徑規劃演算法

如圖 5 所示，路徑規劃的演算法從來源點開始，第一步只能從來源點延伸的邊可以展開，展開後的節點將會被放入一個叫 OpenList 的資料結構中。隨著我們往前移動之後，目前展開但未被拜訪過的節點將丟入 OpenList 之中，而每次均選擇 OpenList 中花費最少的一個繼續展開。已展開過的節點則被歸入已拜訪過的群組。我們在每一拜訪過的節點中，均記錄從起點到此點的最小花費。如果在這個展開的過程中，為已拜訪過的節點找到另一個較低的花費路徑，則我們會取代舊的路徑並且存放這個抵達節點的新路線。這個逐漸最佳化的行為稱為節點鬆弛(node relaxation)。

由於我們是用時間預算式的路徑規劃，因此在時間範圍(Timer < TIME_LIMIT)內會盡可能搜尋可觸及的街圖節點，然後以分數最高的節點(CUR_BEST_NODE)作為目標組態，傳回起始組態至目標組態的路徑。

4. 虛擬攝影機的轉切

在即時環境下的攝影機規劃，因為無法事先得知完整的化身移動路徑，加上對於未來路徑的預測準確度有限，因此很容易使攝影機進入虛擬環境中較險惡的位置，例如牆角。當化身越靠近攝影機時，會導致拍攝距離過於靠近，而解決此類問題的方法乃是透過攝影機之間的轉切。當我們評估目前的攝影機已經進入險惡的環境時，便可以根據攝影學的準則，轉切至其他較為靈活的攝影機位置。

4.1 問題描述

如圖 6 所示，攝影機在跟拍目標物時，由於只能短暫預測未來，因此率先進入了狹長死巷的環境之中時，當使用者操縱的化身越接近巷底時，攝影

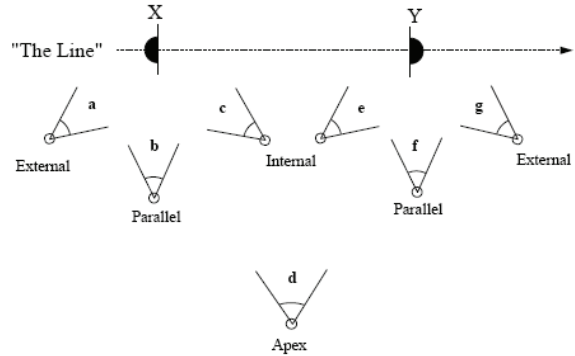


圖 7 LOI 的定義與攝影機架設的方位[7]

機在規劃過程中可展開的節點將越少。最終當化身貼緊牆壁時，攝影機甚至可能被迫沒入障礙物之中。

4.2 攝影機的轉切

解決此問題的方法是使用攝影機轉切。攝影機之間的轉切(Cut off)是一種常見的攝影技巧，藉由轉換到另一台新的攝影機來呈現畫面。轉切在攝影學上常被使用到，例如兩個演員之間的對話場景，就經常會使用轉切的方式來呈現劇情。儘管攝影機在空間關係上是不連續的，但我們仍要求在拍攝後的畫面能夠保持連續性(Continuity)，否則勢必會導致觀眾在視覺上產生錯亂。而在討論轉切的連續性前，我們必須先介紹 Line of Interest。

Line of Interest 亦稱作 Action Axis，它代表一條虛擬的軸，這個軸通常是被拍攝物體所面向或移動的方向，而在兩個人對話的場景中，Line of Interest 則為兩個人物之間的連線。一旦定義好了 Line of Interest，我便可以根據這條線與攝影機的夾角，區分出三種不同攝影機架設方位：外部(External)、平行(Parallel)、內部(Internal)。如圖 7 所示，X 和 Y 代表兩名演員互相面對著面，兩個人之間的連線為 Line of Interest，經由內部方位攝影機可以拍攝到演員正面，平行方位的攝影機可拍攝到演員側面，而外部方位的攝影機則會在畫面上呈現演員的背部。

轉切的條件乃是環境之中必須要有兩部以上的攝影機存在，才能交替輪流擔負呈現畫面的任務。因此我們可以考慮同時在街圖上放置兩個以上的虛擬攝影機進行規劃，每部攝影機使用不同的目標函式，讓攝影機的路徑不至於重疊。但是使用這樣的方法，攝影機規劃所需的成本會和街圖上攝影機的數量成正比，代價過高。

另外一個方法則是在街圖上建立虛擬連結(Virtual Edge)。虛擬連結在每次搜尋時會重新建立於起始節點上，連往街圖上其他的節點。虛擬連結建立完畢後便和原本攝影機規劃一樣搜尋攝影機路徑。有別於正常街圖上的連結(Edge)，攝影機遇

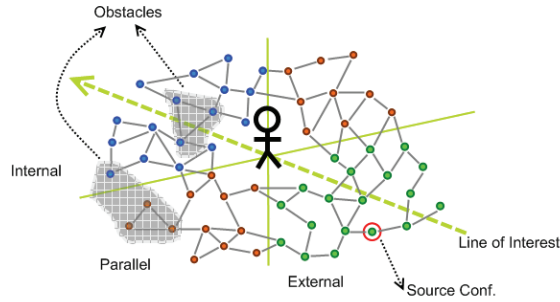


圖 8 依據相對 LOI 的角度區分街圖上節點種類的示意圖

到虛擬連結便會瞬間移轉到另一個節點上，藉此達成轉切的效果。當搜尋完畢後虛擬連結便會移除掉，下一次規劃會再重新建立新的虛擬連結。如此一來不需要額外的成本變成可以產生攝影機轉切的效果。

4.2 街圖上的虛擬連結

在建立虛擬攝影機之前，我們先由化身所面對的方向建立 Line of Interest，如圖 8 所示，接著在此線的兩端區分出內部（藍）、平行（紅）以及外部（綠）方位。根據攝影學的原則，當我們進行轉切時，不可轉切至同一個架設方位的攝影機上。因為攝影機轉切後移動的距離不大，給觀眾的視覺觀感像是同一部攝影機在附近進行跳躍的動作，所以轉切的時候必須從攝影機方位轉切到另外一個不同方位的攝影機。

區分完街圖上每個節點所屬的方位後，便可開始建立虛擬連結。虛擬連結的起始端為起始組態，另外一端的選擇必須符合以下的條件：

- 不能落於障礙物之內，也必須可以拍攝到被拍攝目標物。若轉切後卻拍攝不到目標，則失去轉切的意義。
- 必須和起始組態位於 Line of Interest 的同一邊，否則不合法。
- 新的組態必須和初始組態屬於不同的方位類型，避免畫面產生跳躍的現象。兩個組態之間的距離也不能過近，避免兩個組態雖是不同方位類型，但是兩者都位於邊界，實際距離相當接近，同樣會產生跳躍的現象。

最後我們便可以依據系統指定的個數，在符合上述條件的節點集合中，隨機挑選數個符合的組態，以建立虛擬連結供轉切使用。

加入虛擬連結後，我們可以直接使用第三節所描述的攝影機路徑規劃方法，並且在搜尋時不需區分一般連結或虛擬連結。不同的地方是虛擬連結是每次路徑規劃前便需要重新加入，依據目前組態空間障礙物的分布以及 Line of Interest 的狀況建立虛擬連結。在所回傳的路徑中，我們則需要區分一般



圖 9 測試場景之俯視圖與路線圖

連結及虛擬連結。有別於一般連結，攝影機不需要在虛擬連結上移動，直接傳送至連結的另一端即可。另外，在目標函式的部分，我們額外考慮虛擬連結進行轉切的花費分數(Cut cost)，因此我們擴充原先的 f_3 函式：

$$f_3(q) = \begin{cases} C_{cut}, & \text{if a virtual edge} \\ \text{distance}(q, q'), & \text{otherwise} \end{cases} \quad (6)$$

當路徑規劃經過虛擬連結時，我們使用公式(6)取代原本的移動距離的花費分數(Movement cost)，當路徑規劃經過這條路徑時會加上這個花費分數(C_{cut})。轉切的花費分數會設定得比移動距離來的高，並且隨著時間在一固定的範圍內遞減(如公式(7)所示)。為了避免攝影機轉切的次數過於密集，因此系統會在攝影機剛結束轉切後，調高轉切的花費分數(C_{cut}^{max})，避免馬上又進行轉切。接著再隨著時間慢慢遞減轉切的花費分數，直到最低下限(C_{cut}^{min})為止。

$$C_{cut} = \begin{cases} C_{cut}^{max}, & \text{after cut} \\ \max(C_{cut}^{min}, C_{cut} - 1) & \text{otherwise} \end{cases} \quad (7)$$

5. 實驗結果

5.1 基本場景測試

我們設計了一個場景用來測試是否能夠在充滿障礙物的虛擬中，成功避開障礙物拍攝到化身。在此實驗中，我們並不為攝影機指定特別偏好，系統只要成功規劃出閃避障礙物的路徑即可。如圖 9 所示，場景為模擬巨石陣的環境，環境中的特色是中心有一個小型空曠的廣場，攝影機在此可不受遮蔽拍攝到化身，但出了廣場後便有許多的巨石障礙物會阻擋視線。

圖 10 為實驗結果中行走路徑的取樣點，圖 11 則為攝影機在這些取樣點上的畫面截圖，代表攝影機在路徑上相對應編號的攝影機依序所拍攝的畫面。這個實驗的結果顯示能夠規劃出不受障礙物遮蔽的路徑。整體而言，除了少部分在轉彎的片刻會些微被擋到外，整條路徑上都能夠成功進行跟蹤拍攝。

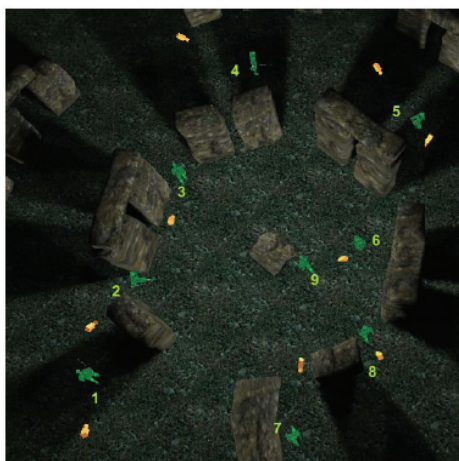


圖 10 實驗解果範例 - 行走路徑之取樣點

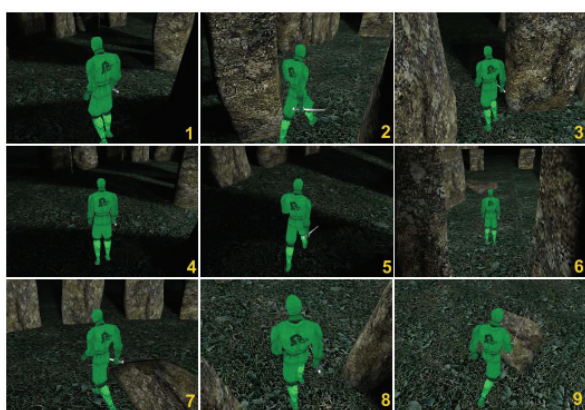


圖 11 攝影機拍攝效果 - 圖 10 中取樣點之截圖

5.2 攝影機轉切之實驗

我們設計一個如圖 6 的場景，讓攝影機率先進入死巷的環境中，然後再將攝影機漸漸逼入巷底，減少其可規劃的空間，最後觀察攝影機是否能夠利用轉切轉移至其他位置。圖 12 為此實驗之截圖，共分成四部分。一開始攝影機以外部方位的方式進行拍攝，並且率先進入死巷的環境中。隨著人物慢慢向死巷末端靠近後，在第二與第三部分的截圖可發現攝影機與化身的距離逐漸拉近。最後第四部分，當攝影機沒有足夠的搜尋空間，又預測到未來可能會陷入障礙物之中，便觸發了轉切機制，將攝影機成功轉移至合法的拍攝位置。

5.3 拍攝目標物未來位置預測深度之比較

目標物未來位置的預測深度是智慧型攝影機模組的一個設計參數。我們設計了一個實驗來瞭解預測深度對拍攝效果的影響。在這個實驗中，我們使用圖 13 中類似迷宮的場景進行實驗，因為這場景有較多的直角轉折，每當化身轉彎後，攝影機在原本的位置變可能遭到遮蔽，因此這時候對拍攝目標物未來位置的預測便顯得個外重要。我們希望使

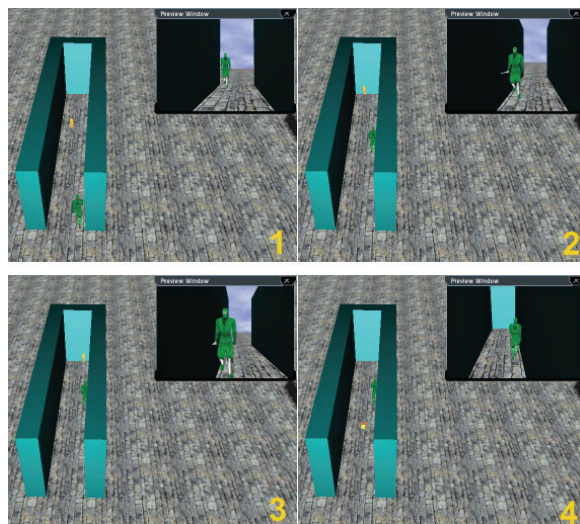


圖 12 轉切實驗結果的截圖 -

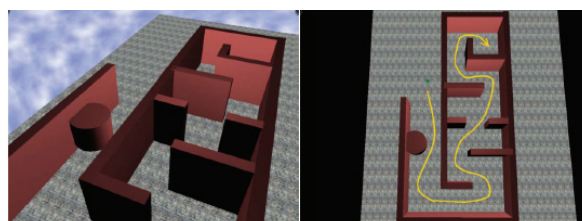


圖 13 預測深度比較的實驗場景之俯視圖與路線圖

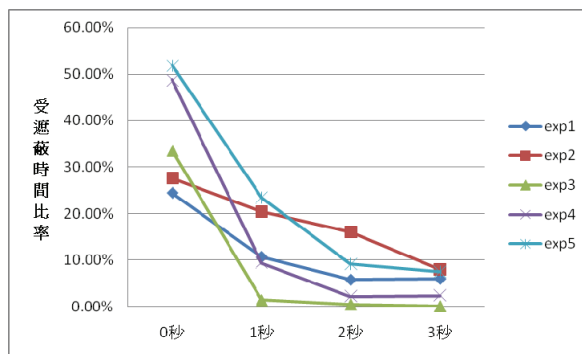


圖 14 拍攝目標物未來位置預測深度之比較的實驗結果

用這個場景，並且搭配 5 種不同的攝影機偏好參數 (exp1-5)，以及不同深度的未來位置預測 (分別為 0 秒(無預測)、1 秒、2 秒和 3 秒)。接著讓化身在場景行走同樣的路徑，觀察攝影機規劃時是否會因預測目標物未來的位置而有所差異。我們評估的方式是計算整條移動路徑上，攝影機遭到遮蔽的時間百分比。

如圖 14 所示，我們可以發現每一種攝影機參數的實驗中，有預測拍攝物未來位置的遮蔽狀況很明顯皆比無預測來的好。因為無預測的狀況下，攝影機規劃器大部分必須等到目前攝影機遭受遮蔽了才會搜尋更好的位置。而有預測的狀況則可事先猜測到將要受遮蔽，先規劃好到其他位置，因此當攝影機規劃器能夠預先知道未來一秒後拍攝物的

位置，便明顯降低了受障礙物遮蔽的時間。在不同組參數的實驗中，預測深度為1秒與2秒時，其結果仍有顯著的差異；而預測深度2秒與3秒之間的差異就不太明顯。

6. 結論

我們使用運動計畫的技術，在即時的3D虛擬環境下規劃攝影機路徑，讓使用者只需專心操縱虛擬環境中的化身，不需額外操縱攝影機，攝影機便能夠不受障礙物遮蔽，明確地拍攝到目標物。另外，此系統讓使用者可以指定攝影機不同的偏好參數，在不同的虛擬場景中，產生特定風格的攝影機架設方式。最後我們還設計出了符合攝影學原則的轉切機制，讓攝影機在被逼入類似死巷的環境中，仍然可以使用轉切的方式，轉移到其他仍有搜尋空間的攝影機位置。此系統的可用性及其有效性在初步的實驗中已經證實，但對於不同使用者的操作習慣及不同型態的場景，則需要作更多的實驗，以進一步設計一個具可適性的智慧型攝影機系統。

致謝

此研究在國科會 NSC 95-2221-E-004-015 計劃的支助下完成，特此感謝。

參考文獻

- [1] D. Arijon. *Grammar of the Film Language*. Hastings House Publishers, 1976.
- [2] W. H. Bares, J. P. Gregoire and J. C. Lester. Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds. In *Proc. of the Tenth Conf. on Innovative Applications of Artificial Intelligence*, 1998.
- [3] W. H. Bares, S. Thainimit and S. McDermott. A Model for Constraint-Based Camera Planning. In *Proc. of the 2000 AAAI Spring Symposium*, 2000.
- [4] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *IEEE Int. Conf. on Robotics & Automation*, pp. 521-528, 2000.
- [5] R. Bohlin and L. E. Kavraki. A Lazy Probabilistic Roadmap Planner for Single Query Path Planning, In *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2000.
- [6] O. Bourne and A. Sattar. Applying Constraint Satisfaction Techniques to 3D Camera Control. In *17th Australian Joint Conf. on Artificial Intelligence*, 2004.
- [7] D. B. Christianson, S. E. Anderson, L. W. He, D. H. Salesin, D. S. Weld and M. F. Cohen. Declarative Camera Control for Automatic Cinematography. In *Proc. of the Thirteenth National Conf. on Artificial Intelligence (AAAI'96)*, 1996.
- [8] M. Christie, R. Machap, J. M. Normand, P. Olivier and J. Pickering. Virtual Camera Planning: A Survey. In *Proc. of the 5th International Symposium on Smart Graphics*, 2005.
- [9] L. W. He, M. F. Cohen and D. H. Salesin. The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing. In *Proc. of the 23rd Annual Conf. on Computer Graphics and Interactive Techniques*, 1996.
- [10] J. Latombe, *Robot Motion Planning*, Klumer, Boston, MA, 1991.
- [11] T. Y. Li and T. H. Yu. Planning Tracking Motions for an Intelligent Virtual Camera. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1999.
- [12] T. C. Lin, Z. C. Shih, Y. T. Tsai. Cinematic Camera Control in 3D Computer Games. In *The 12th Int. Conf. Central Europe on Computer Graphics, Visualization and Computer Vision*, 2004.
- [13] D. Nieuwenhuisen and M. H. Overmars. Motion Planning for Camera Movements in Virtual Environment. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2004.
- [14] D. A. M. Oliveros. *Intelligent Cinematic Camera for 3D Games*. Thesis, Univ. of Technology, Sydney Australia, 2004.
- [15] B. Salomon, M. Garber, M. C. Lin and D. Manocha. Interactive Navigation in Complex Environment Using Path Planning. In *Proc. of the 2003 symposium on Interactive 3D graphics*, 2003.