整合式電腦群體動畫運動行為控制研究

Behavior Control of Crowd Simulation with Combined Approaches

王盛祿　鐘世凱

Sheng-Ru Wang　Shih-Kai Chung

摘要

由於電腦科技的進步，在現今的電影或電視影集中，開始嘗試以電腦動畫的群體運動來模擬真人或卡通角色所組成的群體場面。然而由於群體的角色數量眾多，衍生出動作控制以及電腦硬體負荷的問題。此時若仍以傳統設定關鍵格的方法，將會耗費大量的人力，因此必需藉由電腦的特長，將角色行為準則數理化，以程式的語法來模擬群體運動的路徑來大幅節省動畫師所費的勞力。本研究以虛擬力的個體為基礎，並以一個十字路口為例，討論群體如何避免撞上障礙物，以及如何對交通號誌的事件做出反應，並加入尋路演算、設定關鍵格的個體，來進行分析與探討，並將求出的路徑與 3D 動畫軟體中的高解析度模型做結合。經由這樣的分析，對具有自主性的群體動畫模擬提供一可行的做法。

關鍵詞：群體動畫、尋路演算法、虛擬力。

ABSTRACT

As a result of Computer Graphics and technology progress, the movies and television programs start to use computer animation to simulate the crowd scene which composed by real person or cartoon characters. Because the numerous characters in a scene, it creates the problems of animation control as well as the loading of computer hardware. Traditional control methods, such as setting keyframes, can consume much animators' time and efforts. Motion path of the crowd can be computed by stylizing the behavior rules of the characters, and a lot of labor force can be saved. Our research applies the virtual force agent as a foundation, and takes a crossing as the example. It studies how the crowd avoids hitting the obstacles, as well as how it responds to the traffic-light event. Path-finding, and keyframed agent were integrated in proceeding the analysis and discussion. Finally, the computed motion path and high resolution models in 3D software are joined together to create crowd simulation. As a result, our work proposes a feasible framework in simulating the independent agents of crowd simulation.

Keywords：Crowd simulation, Path-finding algorithms, Virtual force.

一、 前言

山谷間數不清的蝴蝶，草原上無邊無際的羊群等等，在在展現出”數大就是美”的感覺。在電視影集，電影，或動畫影片中也不例外，當出現壯觀的場景，為數眾多的動物或人群，如戰爭的場面，或如動物的集體遷徙等，常會引起觀眾心理的振奮，釋放的感覺。實際運用到電影上的例子有很多，如迪士尼卡通電影「獅子王」中，一群野牛在山谷中狂奔；「花木蘭」卡通電影中蚩尤率領大軍從山上進攻下來；「魔戒」中數十萬大軍進攻；「海底總動員」中魚群排成箭頭為小丑魚指示路徑等等都是。

1.1 群體運動定義

在電腦動畫中，所謂的「群體運動」是指在畫面的場景中有為數眾多的個體而組成的群體，這個群體中的每個個體有著大致相同的造型、做著大致相同的行為，但是細看每個個體的造型或動作又略有差異。電腦擅長的是「複製」，只要能夠做一個，就能複製出好多個，但這樣複製的結果會造成大家都是一樣的動作或造型，非常的機械

王盛祿 台灣藝術大學多媒體動畫藝研所研究生
鐘世凱 台灣藝術大學多媒體動畫藝研所助理教授

化，因此必須透過動作的不一樣，貼圖的不一樣，造型的不一樣再加上隨機的排列組合來造成個體之間的差異。

1.2 研究動機與目的

群體運動在製作上會遇到兩大困難：由於群體數量眾多所遇到的動作控制問題，以及由於群體數量眾多所造成的硬體負荷。

1. 由於群體數量眾多所遇到的動作控制問題

一般在為電腦動畫的角色做動作設定時是用設定關鍵格（Keyframing）的方式，如果這個角色是主角，那麼動畫師花時間在這個角色上是值得的。但是當畫面群體的數量很多時，會使得角色的缺乏自動性，以及對不需要的動作做細節控制，需要耗費動畫師大量的勞力等問題。

2. 由於群體數量眾多所造成的硬體負荷

一旦場景中角色模型的總面數超過某一個數目後，由於記憶體的不足以及CPU運算負荷加重，還有顯示卡無法有效的即時更新在電腦螢幕上，將造成電腦無法有效率的執行甚至當機的命運，因此為了避免這樣的情形發生，必須以低解析度的模型或方塊來代高解析度的模型。

基於上述的理由，我們希望能夠產生自動化角色來做行為模擬，藉由電腦擅長「複製」以及「運算快速」的特性，來幫助動畫師自動產生群體運動所需的運動路徑以及解決其所衍生的旋轉以及動作等問題。同時動畫師以設定關鍵格的方式也可以適度的介入電腦的模擬中，在兼顧效率和動畫師的期望下，完成群體運動動畫的路徑以及動作設定部分。

二、 相關研究

2.1 群體運動計算方式

當個體的數量龐大時，由於個體與個體之間會發生影響，我們可以分為兩種方法來計算：

1. 以分離式方式計算群體運動

分離式的方法是把群體運動問題切割成多個個別的子運動問題，然後我們按照一個順序逐一解決子問題，比較晚被計算出來的角色會被其他較早計算出來的路徑所限制，避免與它們碰撞。

2. 以集中式方式計算群體運動

集中式的方法是把整個群體看成一個大的單元來看待，每一次計算只決定每個個體的一步，並根據上一步來產生下一步。例如使用位能場隨機路徑計劃演算法[5]，用空間中的位能場來引導角色，隨機選出鄰近空間中整體位能較低的組態來移動[1]。用虛擬力來模擬群體運動[10]則每一次只決定每個個體下一步的加速度。

2.2 用虛擬力來模擬群體的行為

Reynolds[11]以分子系統架構[10]為基礎來模擬群體運動，並且提出操縱群體行為（Steering behavior）的三個規則

1. 個體之間能避免碰撞（Collision avoidance）：

個體與相鄰個體之間能夠避免發生相互碰撞。

2. 個體之間速度相互對應（Velocity matching）：

個體與相鄰個體之間的速率和方向大致是接近的，確保群體能朝同方向前進。

3. 個體朝群聚中心接近（Flock centering）：

個體嘗試接近由週遭的個體所組成的小群體的幾何中心。

依據操縱群體行為規則分別作運算，會算出幾個向量，最簡單的方法就是將這幾個向量平均相加，這個向量就是這個個體的加速度，和這個時候的速度相結合，變成下一個時間格的速度。

我們也可以將這幾種力依照優先順序經過加權（比如說避免碰撞的加權最高、速度對應次之、群聚頃向最少）後相加在一起變成一個向量，這個加權的比值可以是動態的，當個體遇緊急狀態如快撞上障礙物時，避免碰撞的加權可以迅速加高，以反應這個緊急的狀態。

當群體的密度過高的時候，群體中的個體彼此會迅速產生排斥力，而朝外圍散開；相反的，當群體的密度過低的時候，群體中的個體會彼此迅速的產生吸引力，朝中心集中。特別是模擬較無自主意識的動物如魚群、鳥群時有較佳的效果。

2.3 結合尋路演算法及虛擬力

跟隨領導者（Leader following）的做法是，群體中有一個領導者，其他的跟隨者是靠近這個領導

者但要注意不要擋到領導者的路徑情況下，跟隨著領導者前進[12]。如果要讓群體以最短路徑前進，可讓領導者具有搜尋最短路徑的能力，而跟隨者是用虛擬力來跟隨著領導者，並讓領導者具有吸引力來凝聚群體。如果跟隨者被障礙物擋住，而找不到領導者時，會根據領導者留下的足跡去尋找，多個領導者的情況下，每個領導者以分離式路徑計劃方式來決定路徑，並且能夠預先保留跟隨者的空間[8]。

另一種做法是先在場景中建立街圖(Roadmap)，當目的地決定時，找出街圖上的一條路徑，路徑上只有決定街圖節點的順序，每一個個體會被最接近的節點所吸引，經過這個節點之後，再被下一個節點所吸引，最後整個群體到達終點。也可以說目的地的找尋是以街圖法來搜尋，而個體間的相互位置是以虛擬力來維持群聚的行為[6]。

2.4 使用限制以及相關問題

使用虛擬力來模擬時，在考量成本效益和行進路徑在可接受的範圍內。許多的物理法則並沒有被運用到。以鳥群為例，比如說重力、浮力、空氣阻力等。

如果在處理比如像人這種有腳的動物，在起伏的地方作群體運動時，以群體運動只能提供位置和方向的情況下，以相同的單一循環跑步的動作套用上去勢必會有不真實感，如果情況需要或鏡頭夠近的話，需要配合自動產生下半身動作的運動計劃[2][7]。

更複雜的行為跟環境，則需提供更完整的環境資料以及更多的行為資訊。如對某些特定的個體設定必須經過的或不能經過的目標點（Interest points）。動作點（Action points）需設定比單純的路徑更多的資訊，包含地點、區域、要發生動作或不作動作、旋轉的向量和關聯的動作等，當個體經過這個點後會再走到下一個目標點或動作點[9]。

如果要讓群體運動整體的運動形狀被限制在特定的幾何形狀內，而且幾何形狀內的群體又能在設定規則下自由的行動，例如「海底總動員」（Finding Nemo）中魚群排成箭頭形狀，則需要在規則之外另外加上一些限制[4]。

在組成一個群體模組的行為是依靠個體和周圍個體的區域察覺（Local perception）來決定，在兼顧運算時間及運動路徑在可接受的範圍內，可以簡化區域察覺範圍為一個球體，以加快模擬的時間。

2.5 個體角色姿態的製作

要為每個個體製作不同的動態，在群體運動上是不合乎成本的，有一種做法是通過Motion Capture方式擷取動作然後為動作設上noise讓每個個體的動作看起來有些不一樣[3]，不過這種做法比較適合個體之間的相互移動很小的情形。通常為配合電腦行為模擬，動畫師會為角色設定一組循環的動作（Motion cycle），比如人的走路循環、鳥翅膀的拍動循環等，然後再將這個動作做時間差隨機分配給個體，依據個體運動的速率來決定取樣的循環動作。

三、 路徑的計算模擬

3.1 行為的準則及力量計算

Reynolds[11]的研究中，主要針對鳥群、魚群等較無自主意識的群體給予行為的規則，群體會聚集成一個小團體前進，在處理像人等具有自主意識的個體時，第二條及第三條規則並不適用。因此，本研究針對要模擬的環境：一個繁忙的十字路口，根據觀察的結果，簡化後設定行為準則，我們依照行為準則來給予個體所需要的虛擬力，行為準則敘述如下：

1. 個體之間避免碰撞

個體與相鄰個體間能夠避免互相碰撞。

2. 個體會沿著人行道前進

個體會沿著人行道的方向前進，如果因避免碰撞短暫的轉向，當原因消失時，能夠繼續沿人行道方向前進。

3. 個體避免撞上環境中不可走的區域

個體要避免碰到牆面或走到車道上。

4. 遵循交通號誌的指示

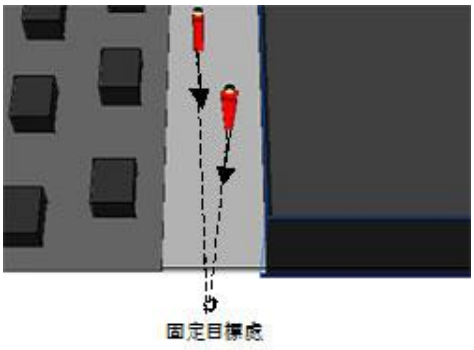
遵循交通號誌紅燈停，綠燈行的指示。

在力量及個體旋轉方面：個體所受的每一種外力我們需要求得兩個部分，一為力量的方向，一為

力量的大小。個體的旋轉角度是由atan2(ManVel.x, ManVel.z)比值反正切函數求得徑度值。

3.2 追尋固定目標

本研究中目標點就是在人行道的末端延伸處，讓個體在每一次計算的時候都受到這個目標點的吸引力所吸引，如此個體就會一直沿著人行道走，一直走到邊界處再從另一端的邊界出來。



圖一　追尋固定目標物示意圖

如圖一是追尋固定目標物示意圖。這個吸引力就是將目標點位置的座標減掉個體所在位置的座標，可以得到一個向量：

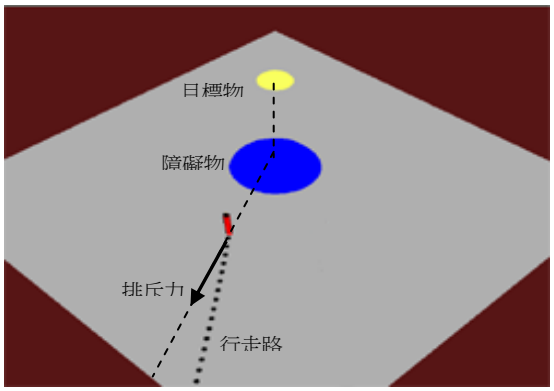
$$Goal\_Force=GoalPos-ManPos；$$

這個向量就是吸引個體往目標點走的力量，因為這個吸引的力量和距離無關，所以力量的大小應該設定為定值。

3.3 避免碰到障礙物

本研究中個體要避免碰到障礙物也就是個體要繞過障礙物而且達到所設定目標物的位置，所謂的障礙物就是場景中其他的個體。本研究中採用兩種做法：

1.　障礙物產生排斥力

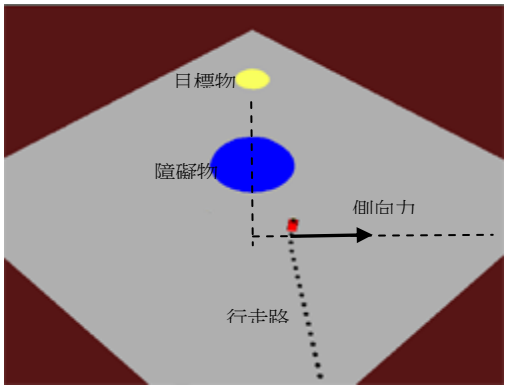


圖二　障礙物產生排斥力示意圖

如圖二是障礙物產生排斥力示意圖。如果個體和目標物的連線以及目標物和障礙物的連線夾角大於90度，而且個體和目標物的距離小於必須開始轉向的距離，則藉由障礙物產生一個排斥力將個體推開(兩個向量的夾角可依內積公式求得)。

2. 障礙物產生側向力

如圖三是產生側向力示意圖。如果個體和目標物的連線以及目標物和障礙物的連線夾角大於90度，而且個體和目標物的距離小於必須開始轉向的距離，則藉由障礙物產生一個垂直於障礙物到目標物方向的側向力量將個體推開，不過由於垂直的方向有兩邊，要處理過讓這個側向力產生在個體的同一邊。



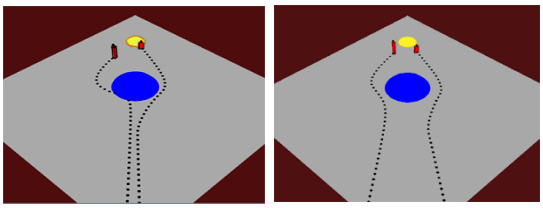
圖三 障礙物產生側向力示意圖

3. 障礙物產生排斥力及側向力的比較

用這兩種方法都可以使得個體繞過障礙物而達到目標物位置，但是用排斥力的方法當個體和障礙物以及目標物位置排列接近一直線時，由於排斥力的方向和行進的方向近乎成平行的相反方向，會造成所提供個體離開目前軌跡的力量過小，很容易就撞上障礙物，若乘以較大的係數讓排斥力增大，則當個體進入所設定的開始轉向的範圍的時，是有足夠的力量讓個體偏離目前的軌道，但是當個體越來越偏離一直線的時候，由於力量過大，很容易就造成離開障礙物太遠的情形。

相對的用產生側向力的方法，在還沒繞過障礙物的情況下，側向力的大小和個體、障礙物、目標物的夾角無關，和距離有關。尤其當個體和障礙物以及目標物位置排列接近一直線時，也能夠提供穩定易調整的偏離軌道的側向力量。

如圖四是避免碰到障礙物中排斥力及側向力的比較，左圖是個體和障礙物以及目標物位置排列較不接近一直線時，右圖是個體和障礙物以及目標物位置排列較接近一直線時。兩張圖的左邊是使用排斥力的個體，右邊是使用側向力的個體，可以看出來，尤其是當和障礙物及目標物成一直線時，使用排斥力的個體，在繞過障礙物的路徑上會有比較大轉彎的現象；使用側向力的個體，在繞過障礙物的路徑上會比較平滑。

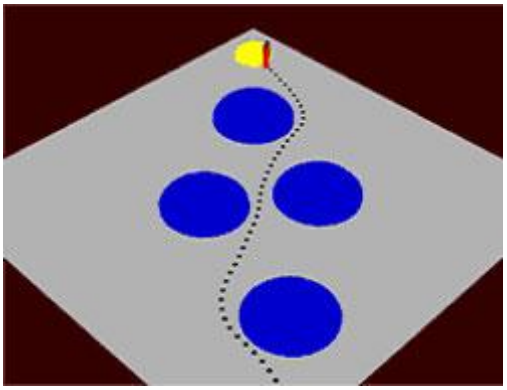


(a)　　(b)

圖四 避免碰到障礙物中側向力及排斥力的比較

4. 複數的障礙物

在個體偵測範圍內的障礙物經常不只一個，有時候在設定的偵測範圍內會有數個障礙物，個體必須找空隙繞過或一一繞過，這時候就要將在偵測範圍內的障礙物分別作上述排斥力或側向力的運算，然後再將每一個力量相加，並避免超過最大值。由於力量相加相同的方向會相加，相反的方向會相互抵消，個體會順著障礙物間的空隙繞過，到達目標物位置。測試的結果如圖五。



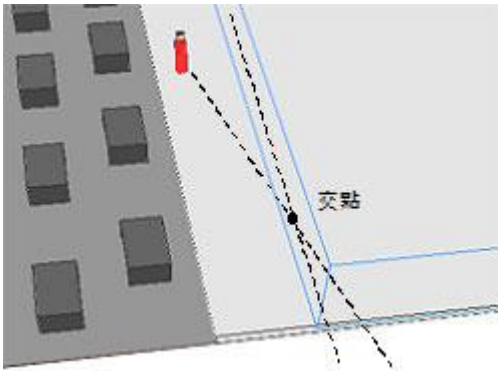
圖五 複數的障礙物示意圖

3.4 避免撞到牆面

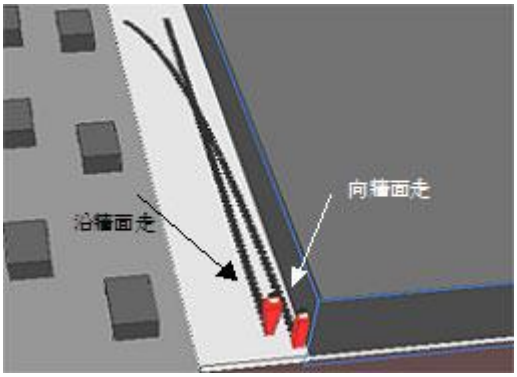
本研究由四個街廓組成十字路口，個體需要能夠不碰撞到牆面，必須要提供一個垂直於牆面方向的力。力量的大小有兩種求法：一種和牆面的夾角有關，一種和牆面的距離有關。

1. 和牆面的夾角有關的大小

如果個體向牆面的方向走，則必須施加外力讓個體轉向，如果個體的角度不是向牆面的方向走，我們可以不需另外施加外力改變方向，讓個體沿著牆面走。因此可以利用個體行走路徑的延長線，和牆面線這兩條線來尋求交點，如果個體到交點的距離越短，表示越快會撞到牆面，則牆面就必須提供越大的推力，如果個體到交點的距離越長，表示不會很快撞到牆面，牆面可以提供較小的推力。個體到牆面交點示意如圖六：



圖六 個體到牆面交點示意圖



圖七　向牆面及沿牆面走受作用力路徑示意圖

如圖七，沿牆面走的個體幾乎不會受到牆面的推力，但是向牆面走的個體會受到牆面的推力而漸漸轉向。

2. 和距離有關的大小

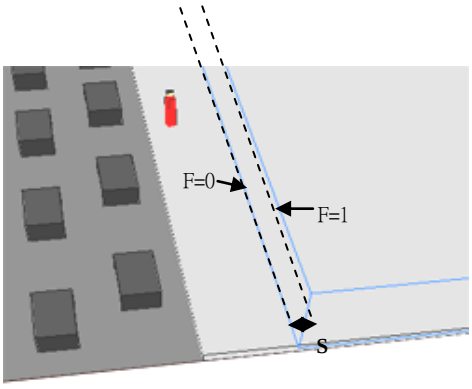
如果沒有其他力量的作用下，只有這樣的力量作用是可以防止個體撞入牆面的，但是由於個體受到許多力量的牽引，如為了避免撞上其他的個體，很有可能會因為沿著牆面行走，避免撞牆的力量過小，而被其他的力推入牆面中，所以在本研究中還加入了和個體行進角度無關，只和個體與牆面距離有關的力，強制個體退回人行道上。

當距離越小，牆面將個體推回人行道的力量越大，相反的，距離越大，牆面將個體推回人行道的力量就越小。

力量大小的運算式如下：

Magnitude=(S−Distance)*1/S；

其中Distance為個體到牆面的距離，如果距離為0，也就是個體在牆邊，則牆面推個體往人行道的力量為1，如果距離為S，則牆面推個體往人行道的力量為0。示意如圖八。



圖八 牆面推力示意圖

在十字路口的部分，為當個體穿過十字路口時，通常還會有許多其他的個體一起穿過，尤其是剛剛紅燈變綠燈的時候，大家會一起在十字路口上相遇，需要有足夠的空間讓個體往外擴散，而且十字路口沒有牆面，而可以不用給予避免撞牆的力。但是由於個體受到許多其他個體的影響，本來在靠近斑馬線外圍的有些個體就會被推到超出斑馬線太遠，導致無法順利的走回人行道上。所以在十字路口的部分，還是要給予避免撞牆的力量，要同時兼顧兩者，假想牆的邊界必須比人行道的寬度還要寬，也就是必須稍微調整Distance的算法，計算如下，D為比人行道還要寬的部分的距離：

原計算式：Distance=fabs((人行道邊界X值)

−ManPos.x)；

調整後　：Distance=fabs((人行道邊界X值+D)

−ManPos.x)；

3.5 個體受到數個外力的合成

個體受到多種外力，最後需要將這數種外力相加，得到一個合成力，再將這個合成力經過設定極大極小值後，加到個體速度的的改變上，進而改變個體的位置。

可能在分別設定各種力的時候，模擬的情況都很良好，但是當力量相加的時候，可 能有些力量就顯得過大或過小了，這個時候，就需要將某種力量其大小值再做調整，以符合個體運動的需要。

3.6 數個個體的模擬

當個體數量增多的時候，和單一個體模擬的情況有所不同。當有眾多的個體聚集在一小塊範圍的時候，會發出相當大的力量，當有其他的個體接近時，很容易因為受力過大而發生轉向或撞牆的情形，解決的方法是設定影響的上限值或增加其他外力來解決這樣的問題。

當個體因為紅綠燈停下來時，由於速度是0或接近0，這時任何其他的個體接近，都會因為發出影響力而造成個體的轉向。解決的方法是設定當個體的行進速度超過某個值時，才可以受到其他外力的影響。

當個體數量眾多的時候，如果每個個體都用相同參數的程式設定，很容易看起來過於規律，最明顯的就是每個個體行進的速度都相同，所以應該在每個個體的速度改變上，乘以每一個個體的速度係數(速度係數例如0.8~1.2間)。

3.7 交通號誌觸發的事件

十字路口還有一個重要的事件就是交通號誌紅綠燈的切換，個體在十字路口的時候，如果行走的方向是綠燈的情況就可以繼續走，但如果是紅燈的話，而且個體很接近十字路口的時候，就必須要準備停下來。

當綠燈訊息出現時，已經因為紅燈停止的個體或者準備停止的個體就可以依照類似上述的計算來增加速度前進，到達未停止前的速度。綠燈開始走時，必須要讓準備行走的個體進入延遲函數，讓個體能夠延遲不同的時間然後起步，這樣可以避免準備起步的個體在同一個時間起步，在十字路口上一字排開的現象，看起來會很不自然。

3.8 動畫師手調設定路徑個體的加入

本研究加入動畫師手調設定路徑的個體一起進行模擬。先在 Maya3D 軟體中製作好角色，以

手動設定關鍵格（Keyframing）的方式，將角色的動態，移動位置設定好。再將位移的資料存成文字檔讀進程式中，讀進的資料如下：

84.649411 0 169.508739

84.648294 0 169.49106

84.647375 0 169.468831

84.646886 0 169.442275

. . . . . .

上述的數字每一行代表一個影格的位置，第一個是 X 值，第二個 0 是 Y 值，第三個是 Z 值。不過在 3D 軟體中設定的位置改變可能要經過反覆的測試，以便跟在程式中模擬的群體速度上能夠搭配起來。

因為每個個體都有各自的編號，可以將這個資料作為手調設定路徑的個體的位置值，並讓這個個體能夠擁有影響別的個體的力量，但是不會被別的個體所影響其位置。

我們希望這個手調設定路徑的個體作為主角，也就是他的路徑具有最高的優先順序，別的個體不能夠改變他的路徑，而且在這裡做了一個設定，希望別的個體在接近主角的時候可以離主角遠一點，讓主角有發揮的空間，因此需要由主角的位置發出一點排斥力，讓其他的個體接近主角的時候，能夠讓出較大的空間來，讓主角作一些動作。

當其他的個體在接近主角的時候，不只會繞過主角，還因為主角本身會發出一些排斥力，而讓個體再稍微遠離主角一點。不過這個排斥的力量也不宜太大，因為如果太大的話，會造成其他個體的轉向，可能會導致其他個體的穿牆或走到馬路上。

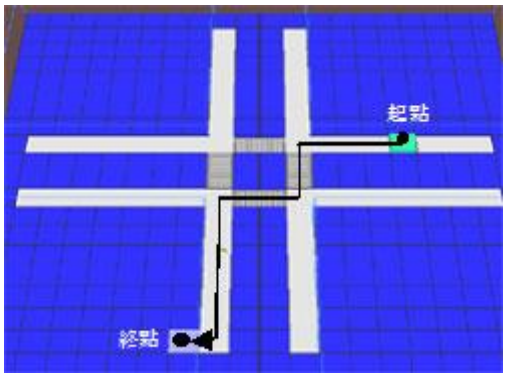
另外針對已經停在十字路口等候紅綠燈的人，我們也希望當主角接近的時候，等候的個體也可以退開讓出空間來讓主角行走。但是這個時候等候的個體速度是0，隨便一個小小的外力都可能會造成個體大大的轉向，甚至有可能不會在等候區等侯，而掉頭往回頭路行走的情況，這不是我們想要見到的。因為個體的面向是由速度的比值來決定的，因此在這個時候，本研究採用不改變個體的速度，而直接改變個體位置的方式，讓個體在位置作移動，空下空間來讓主角經過，但是個體的面向不會因為速度的變化而改變。計算如下：

ManPos+=Repulsion；

如此，則是直接將排斥力的向量不先改變到其速度上而直接改變到個體的位置上。

3.9 尋路個體的加入

本研究加入尋路的功能，期望某部分的群體能從指定的起點，到達指定的終點，以增加群體的複雜性及真實性。尋路演算採用A*尋路演算法尋找最短路徑。先將場景切割成20*20個等分，然後給予起點以及終點，讓個體能夠依據演算法避開不可行走的區域求得最短的路徑，而後根據最短路徑從起點走到終點。如圖九所示：



圖九 求得最短路徑示意圖

在本研究中，最後的表現上並不顯示具有尋路能力的個體，只是藉由尋路演算法求得的路徑，保留其中的轉折點，供其他的個體追隨此路徑而到達終點。

追隨最短路徑的個體和其他個體產生的方式一樣，一樣具有繞過其他個體，避免碰到牆面，等待紅綠燈的能力。不同的是起點和目標點位置略有不同。在尋路的過程中我們會把最短路徑的區塊編號紀錄下來，更重要的是，我們會把轉角處的編號紀錄下來，把這個最短路徑轉角處的位置作為中繼的目標點，要跟隨最短路徑的個體，他的目標就不像其他個體的目標一樣是人行道末端的延伸處，而是這些中繼的目標點。要跟隨最短路徑的個體大致上是從設定尋路的起點出發，每一個追隨的個體都要先接近第一個中繼目標點，

然後才能繼續去接近第二個中繼目標點，等到接近第二個中繼目標點後，才能接近第三個中繼目標點，如此依序接近中繼目標點後才能繼續往下一個中繼目標點前進，直到最後到達終點為止。

確定了轉角處的編號，進而得到轉角處的位置後，就可以把這個轉角處的位置當作是中繼目標點。當跟隨的個體和中繼目標點的距離小於某個設定的亂數值時，則可把中繼目標點移到下一個轉角處，個體就會轉向，並朝著下一個目標點前進。這裡要設定亂數值的目的是希望個體不要在同一條線上轉向，而是在某個範圍內轉向，會比較具有真實性。

3.10 車子的群體運動

本研究除了代表人的個體組成的群體運動外，還有車子所組成的群體運動。總共有4組車子，兩組水平運動，兩組垂直運動。車子運動的方式限定在同一條直線上，後面的車子和前面的車子要保持一定的距離，而且要遵守交通規則，黃燈就要準備停下來，紅燈停止，綠燈行進。

由於車子的運動限定在同一條直線上，因此沒有旋轉的問題，所以可以不用藉由求出速度來改變位置，可以直接對位置作改變。同時為節省記憶體，和代表人的個體一樣，當車子超出邊界的話，會從另外一端的邊界出來。車子位置的改變計算如下：(go_step為這個影格車子應該改變的距離。)

CarPos.x+=go_step；

車子的運動可以採用兩種方式計算，一種是用純數學的方法，一種是模擬人類的想法，也就是用模糊邏輯來判斷。

1. 純數學的方法

車子的運動是採用車頭帶動其後端車子的方式，每個影格車子應該改變多少距離，和每一台車子以及他前面那一台車子距離有關，如果這台車子和他前面那一台車子有較大的距離，則這台車子可以用較快的速度前進，相反的，如果這台車子和前面那台車子距離較小，則必須用較慢的速度前進。計算如下：(M 為自行設定的調整係數)

go_step=這台車和前車的距離/M；

同時應該給予設定 go_step 上限值，避免因為和前車的距離過遠，而造成車子的行進速度過快。當車距小於某個設定值時，則將 go_step 設定為0。因為車列是採用循環的方式，所以車列的頭要計算的是和位在車列的尾的車子的距離。

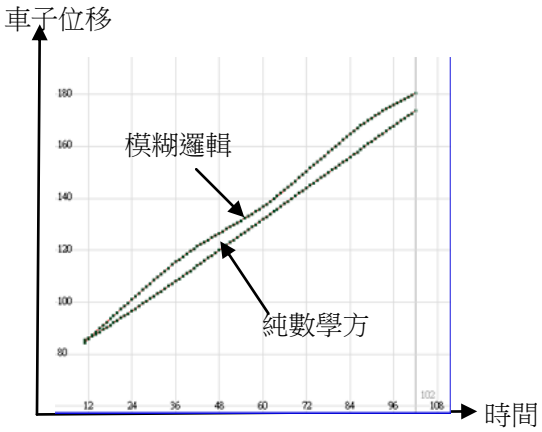
由於車子在移動的過程中，並不知道何時會遇到燈號的變換，也就不知哪一台車該首先停下來，因此應該在交通號誌閃黃燈的時候，偵測哪一台車最接近但是未超過停止線，將這一台車作為車列的頭，作為帶動其他車停止或啟動的依據。

2. 用模糊邏輯的方法

另一種計算車子的運動是採用模糊邏輯的方法，因為真人的判斷不可能像純數學式那麼細膩，可能在腦海中只有加速、減速、維持原速等簡單的判斷而已，因此我們希望最後車子是經由這些簡單的判斷來作為運動的依據。

3. 純數學法及模糊邏輯的比較

根據實做的結果，以純數學方法求得的位移比較成線性的增加，以模糊邏輯的方法求得的位移比較有快慢的變化，如圖十所示：

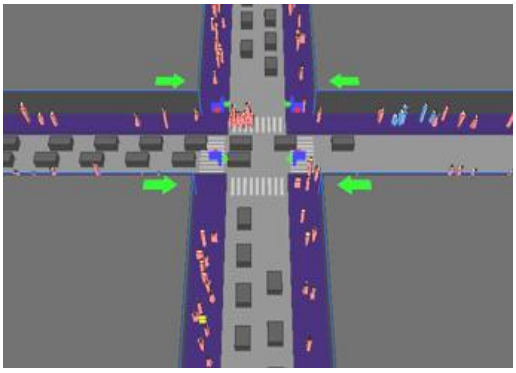


圖十 純數學法及模糊邏輯法位移比較圖

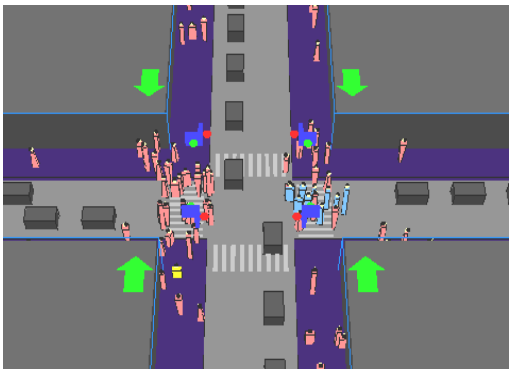
雖然以模糊邏輯法求位移會比較有快慢的變化，比較像真實的交通狀況，但是因為牽涉到的變數較多，調整較為不易，很容易造成位移突然過大或過小的情形，反而離真實的狀況更遠，因此使用的時候必須特別注意，必要時可以設定極大極小值避免這樣的狀況發生。

3.11 整體的模擬

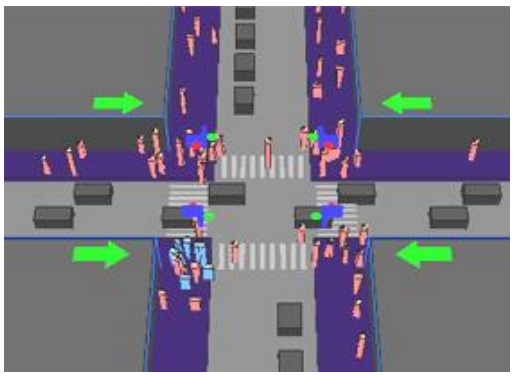
當車子加入後要進行整體的模擬，考慮車子是否有撞到穿越十字路口的行人，如果有，則要進行參數的調整，或者調整人出現的位置。最後在程式中模擬的情形如圖十一：



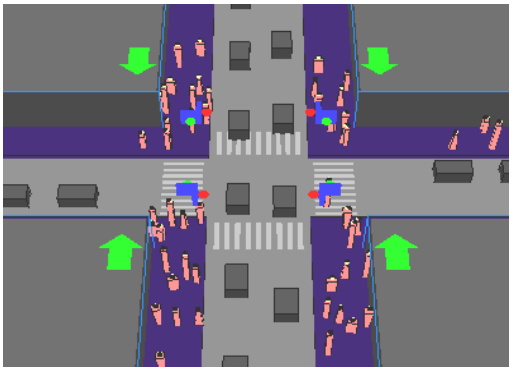
(a)　frame=1



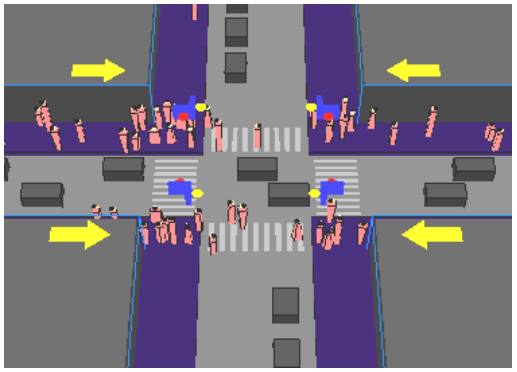
(b)frame=500



(c)frame=1000



(d)frame=1500



(e)frame=2000

圖十一 整體在程式中模擬的結果

四、群體動畫的模擬效果

匯入 3D 軟體呈現

在模擬群體運動的時候，是用低解析度的模型來模擬運動的行為，最後要呈現一段完整的動畫還是需要將模擬的數據傳到 3D 軟體中，利用 3D 軟體的建模及算圖功能，呈現高解析度的角色。本研究是以 Maya3D 軟體作為最後呈現的工具。

4.1 個體角色的製作

製作適合場景大小的人物模型，本研究共做四組模型，一個年輕男子、一個年輕女子、一個高個壯年、一個戴帽子的小朋友。將這四組模型進行複製，大約複製 20 多次，複製到讓整個場景中有 100 個人為止，以符合在程式中模擬的個體數量。

然後將這些角色身上的衣服或褲子的顏色或貼圖進行更換，避免因為複製而過於相似，同時可以對人物的造型做調整，目的在於讓整個場景中的人物遠看起來都略有不同。四組人物的造型如圖十二所示。

圖十二 四組人物的造型

人物的路徑是在程式中模擬，但是每個人物自己本身走路的動作，是在 3D 軟體中設定。因為對每個人物角色以 Keyframing 分別設定走路的動作是不合乎成本的作法，如果設定人物一個循環的動作，也不易處理人物在停下來等紅綠燈時雙腳要站立的問題，因此這裡是以程式的方式，控制人物手的擺動和腳的循環運動。也因為使用到群體運動畫面的鏡頭通常不會距離角色很近，因此一些小的瑕疵是可以在容許的範圍內。

人物的手部及腳部的運動是由反向關節(Inverse kinematics)來控制，因此只要能夠移動 ikhandle，就能控制整個手臂及腳部的運動。因此人物循環的動作主要是以 sin 及 cos 函數控制手部及腳部的 ikhandle。因為人物有停下來等紅綠燈的時候，腳部要直直的站立，因此為人物增加一項屬性”vel”，到時候要傳入模擬時人物的速度，希望當人物速度為 0 的時候，雙手可以垂下，雙腳可以直直站立。

除了 expression 之外，因為還有地面及腳步抬高時高度的限制，所以還要對於這些帶動的物體在設定極限值的地方設定極大極小值。同時參數設定會影響循環的格數，要讓循環的格數能配合從模擬程式傳過來的速度。最後人物的循環動作在 22 格左右，呈現出來如圖十三：

(a)

(b)

圖十三 (a)人物的動作循環 1~11 (b)人物的動作循環 12~22

4.2 資料的匯入

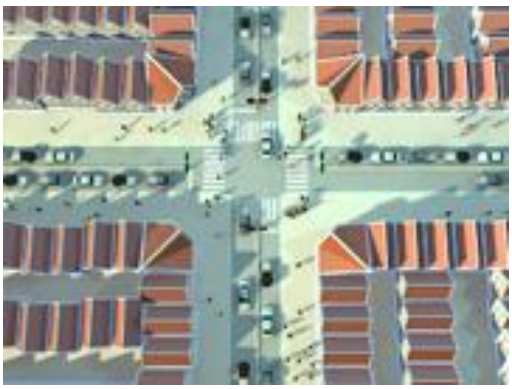
我們需要將程式模擬的結果匯入到 Maya3D 軟體中，要傳入的是人物 X 軸、Z 軸的位置、人物的 Y 軸的旋轉角度、以及人物的行進速度。由於文字檔案不小，直接將文字貼於 script 上有行數的限制，因此以 mel 的方式一次讀取較為適合。同時也不宜在 expression 中使用 if(frame==…..)的方式來判斷影格數，這樣會讓 Maya 花費很多判斷的時間。

由於利用 setKeyframe 的方式，可以讓 hip 物件在每個影格都有 key 值，所以如果對哪個人物的行走路徑或位置不滿意的話，亦可以在 Maya 中以人工的方式修改 Graph Editor 中的參數值，來改變人物的路徑或位置等。

4.3 高解析度模型算圖結果

經由上述的方式，在 Maya 中加以算圖，最後呈現的結果為一段動畫，擷取如圖十四：

(a)側視圖



(b) 俯視圖

(c)frame=1

(d)frame=400

(e)frame=800

(f)frame=1200

(g)frame=1600

(h)frame=2000

圖十四　動畫擷取的畫面

五、 結論及未來研究方向

5.1 結論

本研究針對有自主意識的群體運動中個體路徑的決定，以一個十字路口為例，提出適用的行為準則，以及簡易的實踐流程。藉由自動化角色的產生，實際對一般可能發生的狀況作探討及演算，以解決在電腦動畫上數量眾多所遇到的動作控制以及硬體負荷問題。

在個體角色的路徑規則方面，藉由行為準則的數理化，以程式語法來控制，提供個體追尋固定目標的力、避免碰到障礙物的力，應以側向力為主，排斥力為輔。避免撞到牆面的力，應以和角度有關的方式計算大小為主，和距離有關的方式計算大小為輔。受到交通號誌的觸發的事件應該產生的減速、加速等行為，並將這些力量組合，讓電腦進行計算，求得每個個體在每個影格該有的位置。

但是光是對個體角色設定路徑規則，如果完全由電腦控制，則畫面會有些呆板，因此加入手動設

定路徑以及尋路的個體。手動設定路徑的個體可以作為動畫中的主角，由動畫師來自由的設定複雜的動作或路徑等，並由程式設定讓主角身上會發出排斥力，讓其他的個體能夠離開他一些距離，讓他可以做一些誇張的表情或動作。尋路的個體可以自行設定個體的起點與終點，讓個體依照最短的路徑到達終點，紀錄最短路徑上的轉折點，作為中繼的目標點，讓跟隨者可以根據這些中繼目標點到達終點。這個起點或終點可以是某個商店或建築物，這樣可以做出看起來更有自主意識的群體角色，增加群體運動的可控制性。

車子的群體運動要將車頭去接車尾來計算距離才能達到循環的效果，用純數學的方式計算位移較平順，用模糊邏輯的方式計算位移變化較大。

個體角色的製作上，以幾組模型配合不同顏色的材質貼圖來顯示大同小異的角色，以程式設定循環的動作，同時循環的格數要配合上行走的速度，並使用速度的參數來改變循環動作的大小。

藉由上述的做法，我們可以得到繁忙十字路口高解析度模型的動畫。

5.2 未來的研究方向

經過本研究，可以提供群體運動路徑決定一個可行的方向，但是群體運動會遇到的情況十分的複雜而多樣，不同的場景，不同的真實性，可能會有不同的做法，所以還有許多值得討論的地方，未來研究的方向為：

1. 路徑決定方面

對非平面或不規則的地形如何決定路徑、如何尋找最短路徑。以及當個體的前方有巨大的障礙物或擠滿其他個體的時候，該如何尋找路徑。

2. 動作決定方面

當路徑決定以後，如何給予適當有變化的動作，讓角色的動作更生動。

3. 路徑和動作混合的決定

讓角色和角色之間能夠在程式演算的時候就產生互動的可能性，例如停下來說話，或者停下來看商店內的商品，或者如戰場上的士兵互相對抗等動作及路徑的混合。

參考文獻

中文部分

[1]周旭騏，《以階層式動態編隊的方法計劃群體運動》，台北：國立政治大學資訊科學研究所碩士論文，2003.

[2]陳培鋒，《即時自動產生人體下半身動作的運動計劃》，台北：國立政治大學資訊科學研究所碩士論文，2003.

[3]陳麗真，《電腦動畫之多人物場景建構》，嘉義：國立中正大學通訊工程研究所碩士論文,2002.

英文部分

[4] M. Aderson, E. McDaniel., S. Chenney, 2003, ”Constrained Animation of Flock.” ACM SIGGRAPH Computer Graphics (37), pp.286-297.

[5] J. Barraquand, and J. C. Latombe, 1991, ”Robot Motion Planning: A Distributed Representation Approach,”International Journal of Robotics Research.10(6), pp.628-649.

[6] O. B. Bayazit, J. M. Lien, N. M. Amato, 2002, ”Simulating Flocking Behaviors in Complex Environments”. Technical Report TR02-003,PARASOL LAB, Department of Computer Science, Texas A&M University.

[7] S.K. Chung, and J. K. Hahn, 1999,“Animation of Human Walking in Virtual Environments” Computer Animation 1999, Geneva, Switzerland, Proceedings. IEEE Computer Society, pp.4-15.

[8] T. Y. Li, Y. J. Jeng, S. I. Chang, 2001, ”Simulating Virtual Human Crowds with a Leader-Follower Model”Proceedings of the 2001 Computer Animation Conference, Korea.

[9] S. R. Musse, C. Babski, T. Capin, D.Thalmann, 1998, “Crowd Modelling in Collaborative Virtual Environments”. Proc. of ACM-VSRT’98. Taipei, Taiwan.

[10] W. T. Reeves, 1983, "Particle Systems--A Technique for Modeling a Class of Fuzzy

Objects," ACM Transactions on Graphics, pp.359-376.

[11] C. W. Reynolds, 1987,“Flocks, Herds, and Schools: A Distributed Behavioral Model.” ACM SIGGRAPH Computer Graphics. 21 (4), pp.25-34.

[12] C. W. Reynolds, 1999,“Steering Behaviors For Autonomous Characters”In Game Developers Conference (GDC), California.