

# 利用階層式虛擬力場模擬人群運動

呂昱翰

國立政治大學資訊科學系  
s9008@cs.nccu.edu.tw

李蔡彥

國立政治大學資訊科學系  
li@nccu.edu.tw

## 摘要

隨著電腦動畫在數位內容上的應用日益普及，虛擬人群運動的模擬在近年來成了一個廣為研究的主题。在本論文中，我們提出一個結合人工生命領域中虛擬力場與階層式控制力場的方式模擬虛擬人群的運動。使用這種混合式的力場，可以模擬人群在保持階層式組織的特性下，在環境中進行不會與障礙物碰撞的群體運動。我們利用階層式控制方式，視動畫的需要，靜態或動態地幫人群在移動中進行分組或合併。我們實作了此一方法並評估其效能，而實驗的結果也能輸出至 3D 動畫軟體上呈現。

**關鍵詞：**人群模擬、虛擬力場、階層式控制、人工生命

## 1. 前言

### 1.1 簡介

在電影製作上，常會需要模擬一群人的移動。以魔戒電影為例，當半獸人大軍進攻剛鐸城，千軍萬馬的場面，便是人群模擬的具體展現。在製作動畫的過程中，要產生群體的運動，傳統的方法是以人工設定關鍵格 (key frame)，然後用動畫製作軟體以內插 (interpolation) 的方式算出每個畫格 (frame) 的狀態。但是當人數很多的時候，要設定每個人的運動軌跡非常辛苦耗時。一個較理想的方式，是允許使用者可以用較高階的方式指定；例如指定起始點和終點，而中間的運動過程就能由系統產生。

我們稱人群中的單一個體為一個機器人 (robot)。要計算單一機器人的運動過程，可以利用傳統機器人領域中的運動計畫 (motion planning) 演算法來計算；而在計畫多個機器人的運動時，便需考慮在每個時刻所有機器人的位置，以確保在運動過程中，任何機器人之間不會發生碰撞。由於計算複雜度的關係，過去在這方面的研究常將此類問題以分離 (decoupled) 的方式循序解決，但也因此而失去了計畫的完整性 (completeness)。

在文獻上，人群模擬 (crowd simulation) 有許多不同的作法，其中一種是利用人工生命中的虛擬

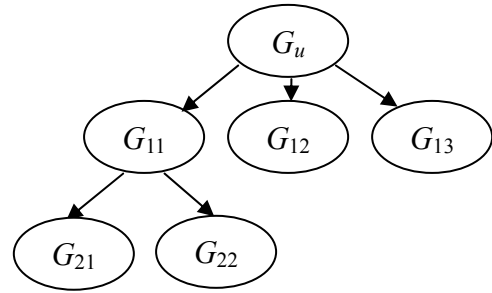


圖 1 群體階層關係圖

力場來引導機器人的運動。如此的方法可以使得群體在運動的過程中，能透過力場參數的設定，保持群聚運動的特性。然而，過去的作法都僅考慮單一群體與單一力場，至於有組織的多個群體運動與多個並存的力場，則多未能加以考慮。

我們提議以階層式控制 (hierarchical control) 的方式進行階層是群體運動的模擬。大體而言，我們希望以更高層次的角度來控制人群。想像人群像一個軍隊，我們可以指定整個「排」的大致運動目標，而實際上移動的時候整個排也會移動在一起，但是一個排裡的數個「班」會稍微的分開，不會不同班的士兵混雜在一起。當路徑有分歧時，我們可以依照班的分組，把一個排拆開來移動，而不讓路徑過於擁擠。

### 1.2 問題定義

我們假設環境中的障礙物可以被投影在一個 2D 的平面空間上，因此障礙物是以多邊形來表示。我們並假設障礙物在運動過程中是不會移動的。另外，我們以一個適當大小的圓形代表每個機器人的幾何。以圓形表示的好處是機器人偵測碰撞的檢查計算可以簡化。

在本文中我們要探討的問題是定義在一個寬度為  $W$ ，高度為  $H$  的環境中。在此環境中有著許多障礙物，分別以  $O_i, \{1 \leq i \leq N_o\}$  來表示。除了障礙物之外，還有許多機器人，分別以  $R_i, \{1 \leq i \leq N_r\}$  表示。在此環境中的機器人，在每單位時間內可以向 X 或 Y 方向移動一個單位距離，單位時間與單位距離是根據使用者賦予個體運動的速度上限而決定的，並且不能超出地圖範圍或與障礙物及其他機器

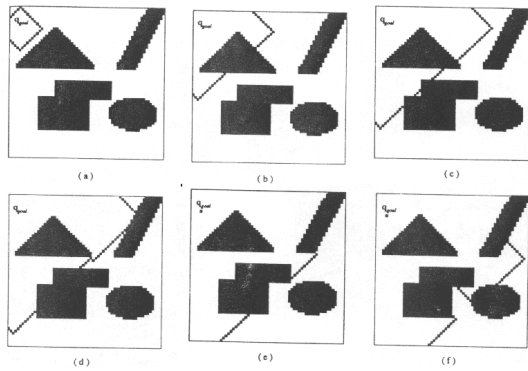


圖 2：NF1 位能場計算方法的說明[1]

人發生碰撞；也就是說任意兩個機器人相交的區域為零，而且任一機器人，也不能與障礙物相交。

在一個環境中，所有機器人所組成的集合，是最大的群體，我們以  $G_U$  來表示。而  $G_U$  包含有  $N_1$  個分解的群體，我們以  $G_{i1} \{1 \leq i \leq N_1\}$  表示；而且對於任意  $G_{i1}, G_{j1}, \{1 \leq i, j \leq N_1\}$  來說， $G_{i1} \cap G_{j1}$  是空集合，而把所有  $G_{i1} \{1 \leq i \leq N_1\}$  聯集起來，會得到  $G_U$ 。同理，我們也可以把  $G_{11}$  做分解為  $G_{2i}$  的集合，如圖 1。

而我們要研究的問題，是在給定的環境中，指定了群體的組成，以及群體間的包含關係，在指定好群體的移動目標之後，可以自動模擬出每個機器人移動的過程。在本篇論文中，我們結合了運動計畫與運動模擬的兩種方式，先利用運動計畫的技術計算出虛擬力場，然後利用此力場作人群模擬。此外，我們加入了階層式關係，兩個群體，隨著其關係的親疏，其作用力會有所不同。

### 1.3 章節架構

在第二章中，我們會整理一些相關研究，包含路徑搜尋的運動計畫以及人群模擬方法。第三章則是用階層式虛擬力場進行人群模擬。第四章是實驗設計與分析，第五章則是結論與未來發展。

## 2. 相關研究

群體運動計畫很自然地是從單一機器人的運動計畫延伸而來，我們在這一章裡首先會介紹幾種運動計畫的方法，然後擴展為多個機器人的運動計畫，最後我們會介紹傳統的群體運動模擬的方法。

### 2.1 運動計畫

對可移動的機器人來說，在環境中要從目前的位置移動到使用者指定的終點，可以用運動計畫的方法找到一條可行的路徑。通常運動計畫的假設是

在環境已知的情況下，對場景中的障礙物做分析，然後再用搜尋演算法找出必碰的路徑。分析環境的方法約略可以分為三種：區域分割法、街圖法、位能場法[7]。

區域分割法是在了解環境的狀況下，把組態空間 (configuration space) 切割成數個子空間，區分出自由空間與障礙物，用某種結構表示他們的空間關係，再根據此空間關係搜尋機器人所要經過的子空間順序。

而街圖法是在自由空間中建立出一些可行的街道，透過這些街道的連接來搜尋路徑。而街道可以根據障礙物的幾何形狀產生，也可以用隨機的方法產生。骨架圖 (voronoi diagram) 與可視圖 (visibility graph) 是以障礙物為基礎的街圖[7]，而機率街圖 (PRM, probabilistic roadmap) [6]與快速隨機擴展隨機樹 (RRT, rapidly-exploring random tree) [8]則是隨機的街圖。

位能場法則是在工作空間中仔細分析所有的地方，建立出位能場，引導機器人往終點移動。位能場就像是地形圖一樣，每個位置有不同高度，以 [1] 中的 NF1 為例，障礙物為最高點，而終點為地形最低點，向四周擴散並增加地形高度，如圖 2。配合最佳優先搜尋法 (BFS, Best-First Search)，搜尋時盡量選擇位能較低處，就可以到達終點。

### 2.2 群體運動計畫

群體運動計畫可以想像成把許多機器人的組態空間合併起來，當做是一個自由度很高的機器人來做計畫。而運動計畫法又可區分為兩種，一種是集中式，一種是分離式。

集中式的方法，是把所有機器人的自由度加起來進行運動計畫。在高維組態空間中搜尋，由於空間龐大，不用隨機的方法幾乎是不可能的。分離式的方法，首先要將所有組態的參數分組。對於多個機器人來說，最好的分法自然就是每個機器人自己的組態分成一組，也就是把群體運動計畫問題切割成多個個別的子運動計畫問題。然後按照順序逐一解決子問題，比較晚計畫的機器人會被其他較早計畫出來的路徑所限制，這是有優先順序的計畫 (prioritized planning) [4]。分離式的方法，只能幫機器人分組，沒辦法幫機器人群體再分組，這是與本文所提出的階層式架構的不同之處。

而在 [9] 的論文中，作者把多機器人動態的分組做路徑規畫，使得效能大為提升，但是這個方法的缺點就是群體無法保證一定是一起移動。

### 2.3 群體運動模擬

有很多群體運動的相關研究並不是用運動計畫的方法，例如使用虛擬力來引導機器人的運動。

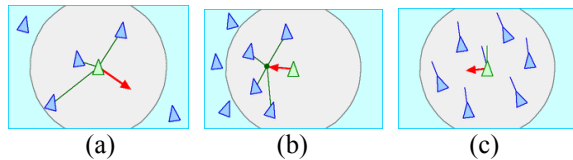


圖 3：(a)分離力，(b)凝聚力，(c)對齊力[10]

其中[10]提出使用分離力 (separation)、凝聚力 (cohesion)、對齊力 (alignment) 來影響機器人的運動，以模擬群體的運動。分散力能讓機器人與附近其他的機器人保持距離，避免擠在一起，如圖 3(a)。凝聚力使機器人向群體的中心聚集，如圖 3(b)。而對齊力則是讓機器人保持與群體一致的速度或方向，如圖 3(c)。

不過此法並沒有解決環境中有障礙物的情況，因此在[2]中合併使用了運動計畫與群體模擬的方法，先在場景中建立街圖，當群體的終點決定時，在街圖中找出一條粗略的路徑。而在[5]也提出了如何維持群體在移動中保持形狀的方法。

而本文所提出的方法，是綜合運動計畫與群體運動模擬，先計算出街圖，然後用位能場作分析，接著轉化為虛擬力場，來模擬群體的運動，我們可以自由的改變群體之間力的交互關係，讓群體可以視環境而進行合併與分離。

### 3. 階層式虛擬力場

階層式虛擬力場，是由三種力混合作用而成，第一種力是[10]中，Reynolds 提出的群體中的三種交互作用力的合力。

而第二種力是我們所設計的群體間的交互作用力，讓群體可以有階層的概念，讓機器人間的關係不是只有同群和不同群，而是有關係的親疏關係；以此出發，我們希望可以實做出群體的合併與分離。

第三種力是路徑影響力，我們預先對場景做前處理，找出地圖中的骨幹圖，當我們設定群體要移動的目標時，可以藉由骨幹圖提供路徑影響力，讓群體往目標移動，而且會盡量遠離障礙物，減少碰撞的機會。

而這三個力，依照加權的加總，可以計算出每個人要移動的方向與向量，如果移動的結果不會碰撞的障礙物或是其他機器人，那麼此移動就是合法的移動，反之則為非法的移動。至於力的加權參數，因為與場景有關，所以是按照經驗法則與實驗結果決定。以下我們將介紹這三種力的計算方式。

#### 3.1 虛擬力場基本作用力

這是在[10]中由 Reynolds 所提出的，也是人群

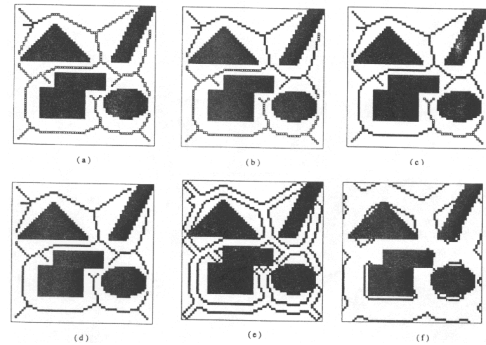


圖 4：NF2 位能場計算方法的說明[1]

模擬的基礎，原本是用來模擬鳥群的群聚飛行，不過用作模擬人群或是蟲群也很適當，原本鳥群可以在三維空間中飛翔，但是我們是要模擬人群，因此只要考慮二維空間即可。我們假設每個機器人有一定的視野範圍，沒有視角限制，可以觀察到附近的所有機器人，依照附近的機器人飛行的方向，以及與自己的距離，計算出三種力的大小與方向，然後加總起來，就是作用在該機器人上的力。

分離力是設計來為了避免機器人之間的碰撞，其方向就是自己到另一個在視野範圍中機器人之間的方向，而大小與距離平方成反比；就如同同性電子間的互斥力，當兩個機器人間的距離很小的時候，排斥力就很大，以此來避免碰撞。

凝聚力能讓整個群體有群體的感覺，可以將群體凝聚在一起。在視野範圍中有許多可見的機器人，把這些機器人的位置做平均，便可計算出此群體個體的中心；凝聚力的方向就是到此中心的方向，而大小是與中心距離成正比。

對齊力則是可以使整個群組往同樣的方向做移動而不會四散。方向的計算方式，是把視野範圍內的其他機器人的飛行方向做平均，所得出來的方向就是對齊力。將這三個力作加總，其結果就是 Reynolds 作用力的向量。

#### 3.2 群體力

群體力是架構在 Reynolds 所提出的三種力之上的力，Reynolds 是假設所有在視野範圍內的機器人，給予該機器人的力是相等的。而群體力則是會依照關係的遠近，給予不同的權重。

我們可以把這種關係，想像成是一個軍隊，而軍隊中有許多軍，軍下面有許多師，師下面分有團，團再細分成營，營分成連，連分成排，排最後分成班。任兩個士兵，可能同班不同排，如果以排的角度來看，整個排的人應該要一起移動的，但是以班的角度來看，不同班的人應該還是要有所區隔，這也是我們設計群體力的原因。

而我們調整的方式，就是在給予力的權重，關

係越近的機器人之間，其排斥力會降低，而凝聚力會提高，反之若關係越遠，排斥力會升高，而凝聚力降低，這樣一來，就可以顯示出關係遠近。原本沒有群體力加入的時候，在視野範圍內的一個機器人，可以給另外一個機器人三種力，也就是凝聚力、對齊力、分離力，我們以一個向量  $F$  來表示。而我們調整的方式，就是用一個關係向量  $R$  與  $F$  做內積，得到新的力  $F'$ 。

在沒有加入群體影響之前，我們是把關係向量  $R$  當成  $[1.0, 1.0, 1.0]$  來使用。而當我們把群體影響加入後，關係比較近的兩個機器人，其  $R$  的前兩個參數（乘上凝聚力與對齊力）會增加，而第三個參數（乘上分離力）會減少。也就是說， $R$  是反應兩個單位之間關係的向量，我們稱之為關係向量。我們由群組間的包含關係，就可以定義出任兩個機器人間的關係向量，以此來做模擬。

### 3.3 路徑影響力

在環境中，一個群組會有一個移動的目標，而在群組中的所有機器人，會盡量往該目標移動。至於移動方向的選擇，我們使用了[1]中所提出的 NF2 骨架圖，針對整個環境計算出骨架以做為街圖，如圖 4 所示。依照此骨架，當我們設定好一個群體的移動目標時，我們可以計算出一個在目的地是最低點位能場；只要順著往位能低的方向移動，就可以靠近目的地。而骨架的地方，位能會比兩側要來的低，如果按照位能場移動，那麼就可以避免與障礙物碰撞，也不會跟障礙物靠的太近，因此這個位能場能符合人類的行走模式。

骨架圖的算法是先以所有障礙物的邊緣為起點，向四周擴散，當與另外一個邊所擴散的點交會時，則該點就是骨架。我們可以將此骨架想像成是計算障礙物與障礙物間的中線或是分隔線，這條線上距離兩個最靠近的障礙物是等距的，該距離就是路面寬度。而機器人行走在此骨架上，與障礙物碰撞的機會最小。

位能圖的算法是我們先把所有障礙物的位能場設定為無限大，而目的地的位能設定為零。接下來是由目的地連接上最靠近的骨架；在連接到最靠近的骨架之後，以該點為起始點，在骨架圖上擴散，每擴散一次，位能場就遞增，直到所有相連接的骨架都擴散完畢為止。最後我們以所有的骨架為起始點，向四週擴散，每擴散一次就把位能遞增，直到把所有沒有障礙物且相連接的地方都填滿為止。此位能圖可以保證，往位能低的地方移動一定是盡量靠近骨架，且會靠近目的地，也就是我們想要的結果。

每一個群體，都會共享一個位能圖，因為群體有共同的目標，而每個機器人，會依照現在的位置，尋找與現在位置相連的地方，位能最低的方向，以此當成路徑影響力的方向。

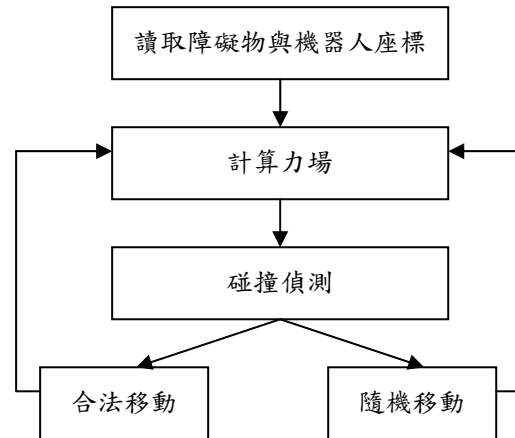


圖 5 系統流程圖

### 3.4 碰撞偵測

碰撞偵測可分成兩個部分，一部分是與障礙物的碰撞偵測，而第二部分是與其他機器人間的動態碰撞偵測。我們會把環境所處的空間切割，把連續的空間，切割成方格點。如果該方格點裡面有包含障礙物，那麼我們就會把該方格點設定為 True，反之則為 False。我們要偵測有沒有與障礙物碰撞，只要讀取該點的值，判斷是 True 還是 False 即可。而與其他機器人之間的碰撞偵測，可以直接拿兩個機器人的圓心距離和兩機器人的半徑和作比較，就可以判斷有沒有碰撞。

### 3.5 參數設定

之前的三種力加總起來，就會是影響此機器人的作用力，把此作用力的方向跟原本機器人的移動方向做平均，就是下一次的移動方向。而三種力不是直接加總起來，而是有加權的加總，因為力是我們所虛擬出來的，並不具有真正的物理意義，藉由參數調整，我們可以找出一組看起來比較擬真的模擬方式，參數的設定是依照經驗法則和實驗結果來決定的。

## 4. 實驗與分析

我們以上面的理論為基礎，實作出一個實驗系統，並做了簡單實驗，場景的大小設定為  $256 * 256$  的格狀平面。

### 4.1 系統設計

我們的實驗系統是以 Java 程式語言開發。我們

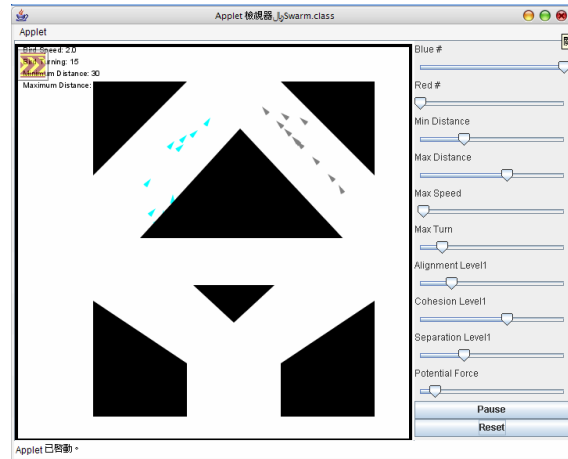


圖 6 實驗系統介面

以網路上所提供的原始 Reynold 模型為基礎[11]，加入群體力與路徑影響力，並且加入障礙物的設計，以符合我們系統的需求。為了呈現的方便，我們以 Applet 的方式實做，因此只需要用網頁就可以瀏覽。而當模擬的過程，我們會記錄每個機器人在每個時間點的座標輸出成檔案，最後我們利用 3D 繪圖軟體 Maya 內附的 Mel 程式語言，把座標讀入之後產生動畫，作為最後結果。

本論文所開發系統的流程，如圖 5 所示。系統剛開始時會讀入環境的資訊，包含障礙物位置、每個機器人的位置及機器人間的關係。之後的每一回合，系統都會先計算相互之間力的作用，然後判斷每一個體應移動的位置，在確認沒有碰撞之後才可以移動。而如果有碰撞，我們會以隨機的方式在附近移動，以免形成死結。最後我們會把座標記錄下來，到一定時間之後，就會把座標資訊輸出到檔案中。

為了方便我們實驗參數，我們讓參數可以在執行時期動態的做調整，整個程式的介面如圖 6，每個機器人以三角形表示，不同的顏色代表不同的群體，以方便表達階層式的關係。

## 4.2 實驗設計

實驗的進行分成三個部份。第一部份是小環境測試，藉由較小的環境，以及較少的群組關係，來簡單測試群組之間的力是否有正常運作。第二部份是複雜環境測試，增加場景和關係的複雜，看在這種情況下系統表現，而第三部份則是效能的測試，測試這實驗系統，最多可以應付幾個機器人的模擬。

小環境測試如圖 6，有兩群群體分別由藍色與黑色代表，人數各為 10 人。這兩群屬於同一個大群體。一開始配置在下方，中間有一分歧點，終點設定於地圖上方。我們預期兩個群組一開始可以並

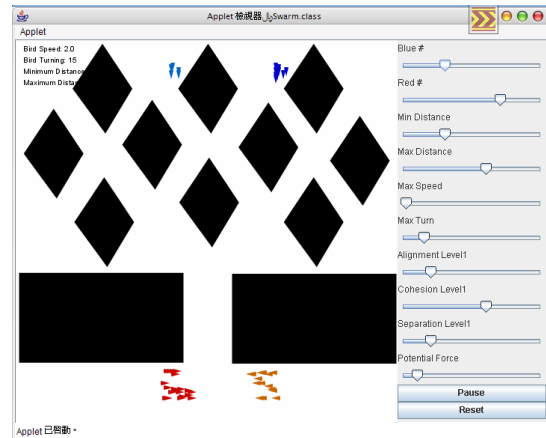


圖 7 複雜場景測試

行在下方長廊，在分歧點因作用力的作用，使兩個群組產生分離，最後在終點合併。

複雜環境測試如圖 7，我們測試了人群模擬中困難的情況，狹橋以及多分歧的道路。在複雜環境中，上方由許多分歧點形成，而下方是一狹路，在環境中有 4 群群體，下方的紅色及褐色是屬於同一個大群體，該大群體的目標在上方。而上方的藍色與亮藍色是屬於同一個大群體，其目的地是在下方。

效能測試是在小環境測試的場景中，亂數散佈機器人以及終點，在模擬過程中，我們計算每秒中的畫格數，如果以即時顯示的標準來看，15~30FPS 是可以接受的範圍，我們測試的機器系統是在 CPU AMD 1.8GHZ，而 Java 的記憶體限制是 64MB。

## 4.3 實驗結果

在小環境測試中，我們可以觀察到兩個群體有分群的行為，如我們預期的在分歧點分離，藍色以及黑色分成兩邊行進，最後會合在終點。

在複雜環境中，在狹橋路段，可以順利通過，但是在上方的多分歧點地區，由於分歧過多，而且對向也有機器人在移動，因此碰撞以及壅塞的情形比較嚴重，但是最後仍然可以移動到指定的目的地。

效能測試請見圖 8，橫軸代表環境中有多少機器人，由 20~150，以 10 個機器人為間距作量測，而縱軸代表在該數量機器人的情況下，畫面每秒更新的次數 (FPS, Frame per second)，取十秒的更新次數作平均，由於在初始化的時候，需要作一些額外的計算 (骨幹圖、位能圖)，因此一開始第一秒的 FPS 不列入計算。

由簡單場景以及複雜場景來看，在簡單環境中是可以藉由力的調整，來形成可行的動畫，但是當



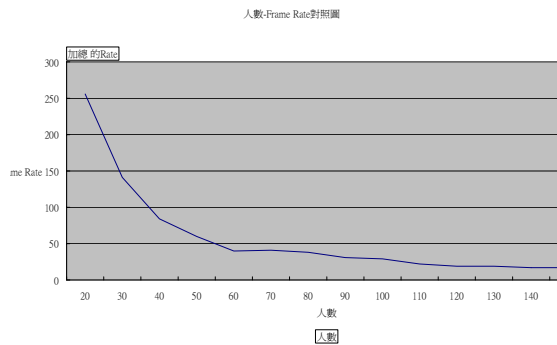


圖 8 實驗效能分析圖

場景變的複雜，如果單以力的方式來模擬，沒辦法解決相向的移動所造成的碰撞，此時必須以其他程序式的方式解決。

在效能分析圖中，我們可以觀察到，在 80 人以下，都可以維持在 30FPS 以上，而在 130 人以下，都可以維持在 15FPS，以即時應用來看，都在可以接受的範圍，場景的複雜度不會影響模擬的效能，因為碰撞偵測都是以查表的方式執行，場景的複雜度只會影響骨架圖與位能圖的計算，但由於骨架圖只需在程式一開始時計算，而位能圖只有在設定群體終點時才須計算，因此不會影響模擬時的效能。事實上，圖形顯示更新上的計算量會比群體運動模擬所需的計算量高，因此當本研究的成果是用非即時的應用上時，所能模擬的人數可以更為提高。

#### 4.4 動畫呈現

在模擬的過程中，程式會記錄每個時間點的座標，然後輸出成為資料檔案。我們利用 Maya 可程式化的特性，額外撰寫了一個 Mel 的程式，首先把障礙物讀入 Maya 中，再讀入每個機器人的每個時間點的座標位置，以此作為關鍵格，利用 Maya 的內插器，以及強大的會圖功能，輸出成影片，外加一些後製處理之後，就成為了一個短篇動畫。障礙物與機器人的 3D 模型是額外製作的，其大小與原本環境中的大小相等，我們以簡單測試場景為地圖來製作動畫，圖 9 是動畫快照。

#### 5. 結論與未來發展

在本論文中，我們提出一個能結合運動計畫與群體模擬的機制，並使用個別化的力場計算參數方式，實現階層式的群體模擬。在實驗中，我們可以看出，在 130 人以下的情況下，可以達成不錯的模擬效能，而環境的複雜度會影響模擬結果，不過可以利用調整參數的方式來達到使用者所期望的模擬效果。

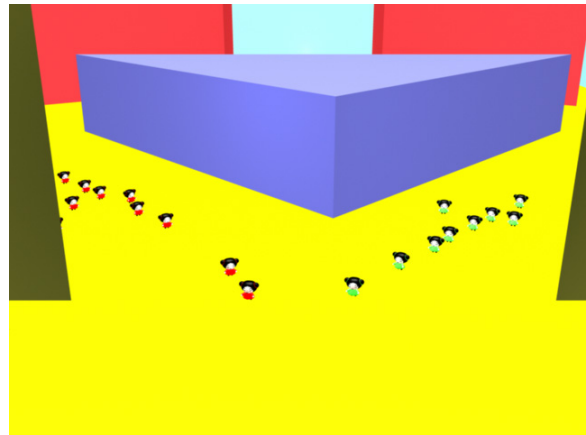


圖 9 動畫快照

未來我們希望能夠加入動態群組分配的功能，把分組變為動態的編組，或是靜態與動態的混合編組，依照路徑的品質，以及多重路徑的選擇，自動進行分組，來達成時間最佳化的群體運動模擬。

#### 致謝

能夠完成論文，要感謝 IM-Lab 全體同學的建議及國科會計畫 (NSC93-2815-C-004-002-E) 的補助。此外也感謝 Elsi 同學贊助的人體模型，讓我可以完成動畫。

#### 參考文獻

- [1] J. Barraquand, B. Langlois, and J. C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions. On Systems, Man, and Cybernetics*, 22(2):224-241, 1992.
- [2] O. B. Bayazit, J. M. Lien, and N. M. Amato, "Simulating Flocking Behaviors in Complex Environments," In *Proceedings of Pacific Conference on Computer Graphics and Application*, 2002, pp 104-113.
- [3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*, Springer.
- [4] M. Erdmann, and T. Lozano-Perez, "On Multiple Moving Objects", AI Memo No. 883, Artificial Intelligence Laboratory, MIT, 1986.
- [5] A. Kamphuis, and M. H. Overmars, "Motion planning for coherent groups of entities," In *Proceedings of IEEE International Conference on Robotics and Automation*, 2004.
- [6] L. E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration

- Spaces,” *IEEE Transactions on Robotics and Automation*, 12(4):566-580, 1996.
- [7] J. Latombe, *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
- [8] S. M. LaValle and J. J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” *Algorithmic and Computational Robotics: New Directions*, pp 293-308.
- [9] T.-Y. Li and H.-C. Chou, “Motion planning for a crowd of robots,” *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.
- [10] C. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” *Computer Graphics*, vol. 21, pp 25-34, 1987.
- [11] C. Reynolds, “Boid,” <http://www.red3d.com/cwr/boids/>