

第三章

系統實現

前面我們已介紹在虛擬實境安全巡邏自動車遠端呈現與操控系統所使用之主要的策略及方法，但是要實現此系統，則需要硬體設備及軟體工具的規劃及搭配。在這章中，主要是說明對此系統的硬體設備，以及所運用的軟體工具做描述，並對硬體設備及軟體工具間之整合做一介紹。



3.1 硬體設備

要實現上述的操控系統，實驗時用到的設備包含了一台個人電腦和力回饋搖桿，個人電腦主要是呈現虛擬實境場景，並且配合智慧型人機介面系統，作路徑規畫及自動車導引的運算，以及進行資料的接收和傳送命令給力回饋搖桿，如表3.1列出此系統的周邊配備；在力回饋搖桿部分，我們利用 Laparoscopic impulse engine [32]，如圖3.1所示，當作接收位置命令和產生力回饋給使用者，這是五軸的搖桿，我們操作握柄可以做X、Y和Z方向的移動，在握柄上方有旋轉環，可以做旋轉的方向的移動，另外在搖桿的頂端，其外觀看起來像把剪刀，可以控制開和關兩個自由度，表3.2為此搖桿規格。

Personal Computer	Athlon-XP 1500 256 MB RAM
Operation System	Microsoft Windows 98
Simulation Manager	OpenGL Microsoft Visual C++
3D Modeling Package	3D Studio Max
Input Device	Keyboard Mouse Impulse Engine 5 DOF Force Reflection Joystick

表 3.1 遠端呈現系統之周邊配備



圖 3.1 力回饋搖桿 (Laparoscopic impulse engine)

Item		Specification
Degree of Freedom	Motion and Tracking	5
	Force Feedback	3
Workspace Size		10×23×23 cm^3
Position Resolution		0.0009 in.(1100dpi)
Max Force Output		2 lbs. (8.9N)
Backdrive Friction		≤ 0.5 oz (0.14N)
Bandwidth	Linear Axis	650 Hz
	Rotary Axis	120Hz

表 3.2 力回饋搖桿規格表(Laparoscopic impulse engine)



3.2 軟體工具

發展虛擬實境安全巡邏自動車遠端呈現操控系統，除了上述的硬體設備外，還需軟體工具的應用；由於視覺感受在一般人的感官中佔了最優先的角色，所產生的影響力也最大，故目前在虛擬實境的繪圖技巧裡，絕大部份均以三維繪圖的方式來表現虛擬場景，而虛擬實境和一般立體動畫兩者之間的差異，在於一般立體動畫中，它是屬於被動的形式，預先繪製靜態場景中所有的物件，接著就是使所有場景中的物件產生隨時間變化而變化的動畫設定，讓畫面中的場景變化看起來是連續的畫面，而虛擬實境的電腦繪圖，它是以及時的方式呈現，所以可讓使用者任意在場景中遊走；本篇在虛擬實境遠端呈現系統中，其軟體實現部分包括3D物件的幾何模型建構、虛擬實境的繪圖引擎以及多執行緒部分，下面會分別

針對這三部份做說明。

3.2.1 3D物件幾何模型建構

在虛擬場景中需要多種的3D物件，可製作此類的模型軟體像3D Studio Max 軟體、AutoCAD軟體等，在這我們主要運用3D Studio Max 5.0 軟體來建構，這是一套應用於美工的幾何造型中，一個很強大的軟體，除了可繪製模型和製作動畫，還可執行多種檔案的匯入及匯出的功能，此軟體應用於各種行業中，像是建築、室內或傢俱設計、工業產品設計、電玩遊戲設計、動畫製作或是傳撥影視業等。



3DS Max 5也將NURBS曲面建模功能加進來，使用NURBS曲面建立模型最大的優點就是自由曲面建立容易，它是一種以函數曲線來定義物體外型的建模技術，且近年來，由於電腦的運算能力及繪圖個方面的功能增強，使得NURBS曲面建模已經變為3D模型製作的趨勢。軟體本身內建相當多的基本物件，像是立方體 (Box)、球體 (Sphere)、錐體 (Cone)、圓柱體 (Cylinder)等，可方便提供使用者使用工具，再調整物體的基本參數來快速建立模型，等建立好模型再將某一材質指定給模型，此動作就叫貼圖(mapping)，另外如果可以加以利用貼圖技巧，就可以利用簡單的物體來表現複雜的物件，不用爲了要呈現出一非常複雜的模型，而大量的使用復合的3D物件，比如說要表現石磚結構的地面，只需選擇是當的材質，且利用簡單的幾何物體形狀，就可以製作出非常真實的感覺。

在3ds max中可以設定七種視角，其包括了上(Top)、下(Bottom)、前(Front)、後(Back)、左(Left)、右(Right)以及透視(Perspective)，由這七種視角讓使用者在製

作物件模型時，隨時切換到不同的視角，不但可以隨時檢視圖形的外觀，也可以將圖形建立在正確的位置；另外在3DS Max 5.0中定義兩主要座標系統，分別為場景座標系統和物件座標系統，如圖3.2所示。場景座標系統是採用世界座標系統的定義，基本上是個絕對座標系統，負責3DS Max 5.0場景方位的指示，3DS Max 5.0中以紅線表X軸，恆指向右為正，綠線表Y軸，恆指向前為正，藍線表Z軸，恆指向上為正。物件座標系統可充分協助使用者配合移動、旋轉及比例變化等變動指令，其是用來協助指示方位，因為要將此物件在三度空間中定位時，適時地將物件座標系統按照其需要做轉換，可產生精準的定位效果。

在 3DS Max 5.0 中共定義了幾個物件座標系統，不同的物件座標系統配合的使用時機也會不一樣，其大約可以分為檢視座標系統 (View Coordinate)，這是一種相對的座標系統，此座標系統是由螢幕座標系統與世界座標系統混合定義而成的；當作用視窗在上下前後左右六個視角時，被選取物件的物件座標軸是呈現螢幕座標系統；當作用視窗在透視視角時，被選取物件的物件座標軸是呈現世界座標，如圖3.2所示。螢幕座標系統 (Screen Coordinate)，這是屬於一種相對座標系統，其不管作用視窗在哪一個視角，被選取物件的物件座標，其軸方向永遠是X恆指向右為正，Y恆指向上為正，Z恆指向使用者為正。世界座標系統 (World Coordinate)，這是一種絕對座標系統，其不管作用視窗在哪一個視角，被選取的物件的座標均與場景座標系統一致，因此物件座標系統XYZ的三個軸向永遠跟世界座標一樣。局部座標系統 (Local Coordinate)中，當物件在空間中的角度方位沒有改變時，其物件的物件座標軸向方位與世界座標系統是一樣的；但當物件在空間中的角度有改變時，其座標軸向方位就是該物件角度改變後的方位，也就是這是物件本身的座標系統。平衡座標系統 (Gimbal Coordinate)和局部座標系統很類似，都是以物件本身作為編輯單位，可以在不影響到其他軸向的情況下，對單一軸向進行旋轉。格面座標系統 (Grid Coordinate)會以所在的格面座標為準，如要計算不同平面間的距離向量，即可以利用此座標系統。選取座標系統 (Pick

Coordinate)是將作用視角中的被選取物件，以其他物件的局部座標系統的座標軸向，作為空間定位編修用的參考軸向，也就是說，選取座標系統就是拿其他物件的座標系統指示的方向，作為它遵循的方向。

在知道所使用的座標系統後，可以開始繪製3D物件，一個3D物件的線架構模型(Wireframe Model)是將空間中很多3D的點連接成線，進而構成以線段型態資料為主的3D模型，其可視為三角形所構成的，因此建構越真實的物件或場景，則必須運用較多的多邊形，也因此帶給電腦中的處理器越大的負擔，而會降低虛擬環境即時模擬的效能，所以在這就利用貼圖的技巧，以簡化組成物件之多邊型的數目，讓計算量可以降低，提高繪圖性能。

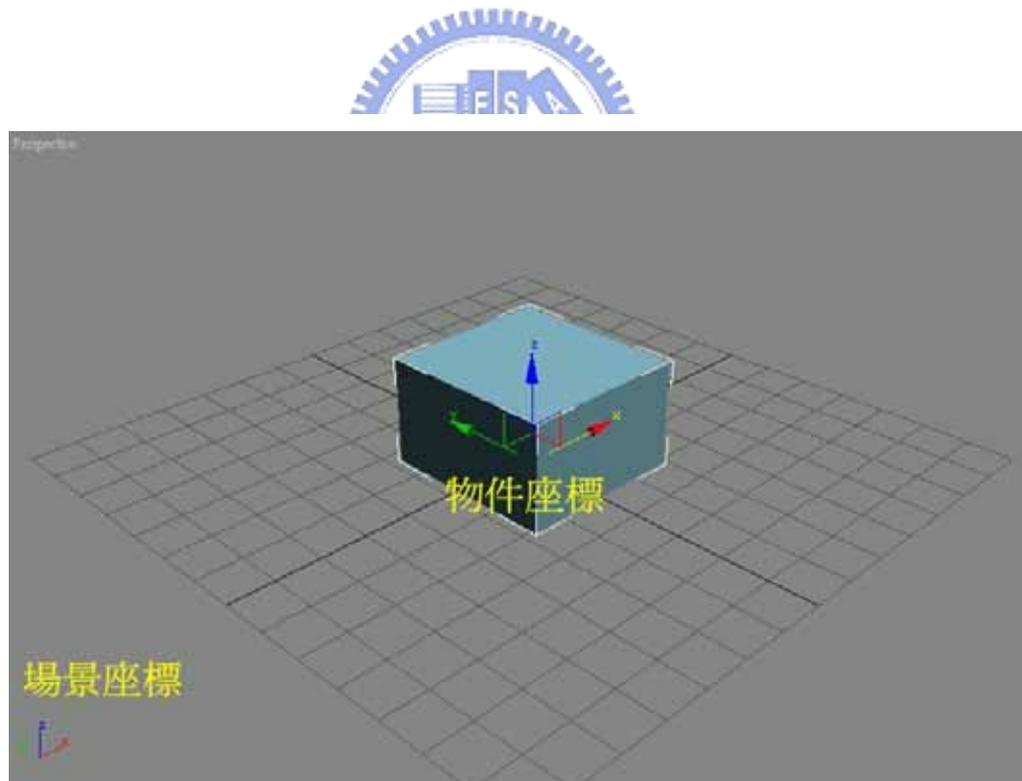


圖 3.2 3DS Max 5.0 所使用之兩主要座標系統

3.2.2 3D繪圖與模型程式庫

在本論文的實驗中，需要使用到電腦繪圖 [27] 的工具以製作場景的影像，現今電腦繪圖的運算環境，大致如圖3.3所示，使用者透過 I/O 設備與視窗系統 (Windowing system) 互動，常見的視窗系統如 Microsoft Windows 系列的作業系統及 UNIX 上的 X window 環境，應用程式及其模型則建構在圖學函式庫 (Graphics library) 之上，所謂圖學函式庫是一個常式 (routine) 的工具集，包含低階的程式碼，雖然工作與圖形的硬體設備很密切，但通常是不會限定設備 (Device-independent)，顯示設備包括顯示卡和其驅動程式以及顯示器等部分，常見的圖學函式庫有 OpenGL、DirectX、WTK 等。

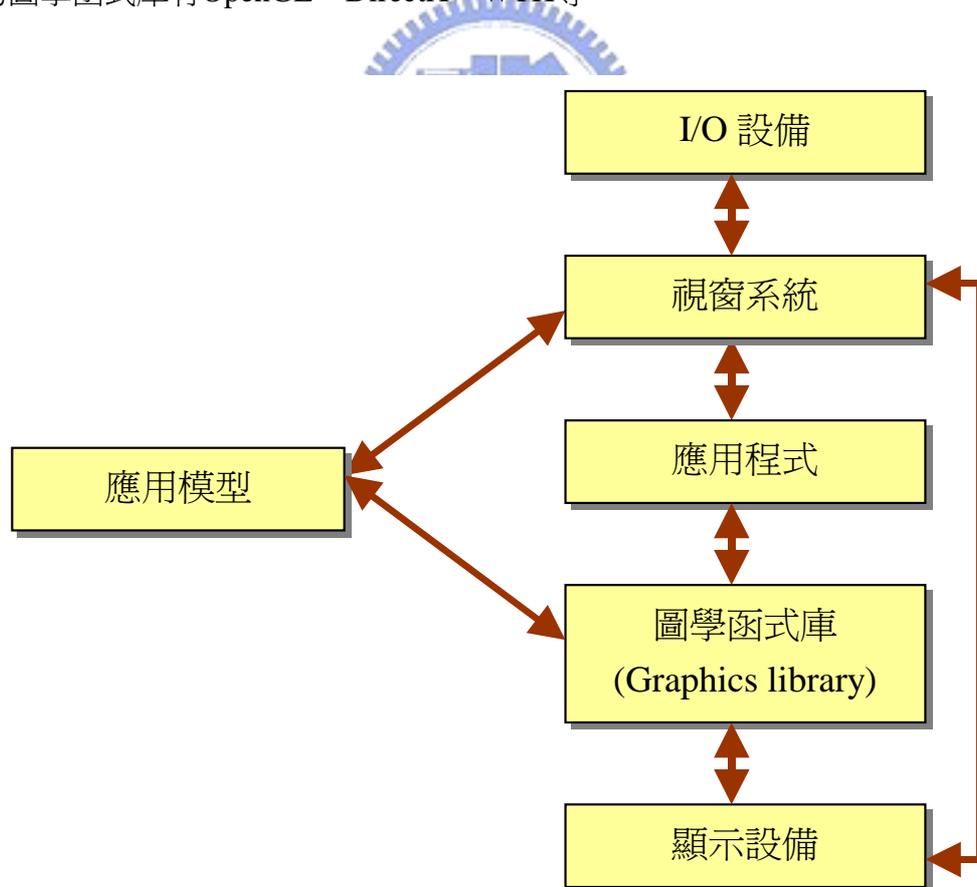


圖 3.3 電腦繪圖之運算環境

OpenGL[1,3,23]的正式定義是「繪圖硬體的軟體界面」，基本上，它是3D繪圖與模型程式庫，既迅速又可移植，OpenGL不像是C或是C++那樣的程式語言，它比較像是C執行時期的程式庫，提供許多包裝好的功能函數。同時OpenGL是文件完備的應用程式設計界面(Application Programming Interface, API)。OpenGL自1992年已成為工業標準，由獨立的OpenGL ARB (Architecture Review Board)管理其規範，其中成員包括SGI(Silicon Graphics)、Digital、IBM、Intel、及Microsoft等大廠，很多影像顯示卡及作業系統也都支援OpenGL。另外，OpenGL具有穩定、可靠、可攜性等特性，尤其是OpenGL的程式可攜性，不論在Windows或是UNIX上皆可使用，這使得用OpenGL所開發出來的程式，在未來更具有發展性，故本論文選用OpenGL作為實驗所用的圖學函式庫。



圖3.4為 OpenGL 呈像流程，當應用程式發出OpenGL API 函數呼叫時，指令會放到指令緩衝區中。緩衝區最終會填滿指令、座標點資料、材質資料等等。當緩衝區清空(Flush)時，不管是經由程式指令或驅動程式本身的設計，指令與資

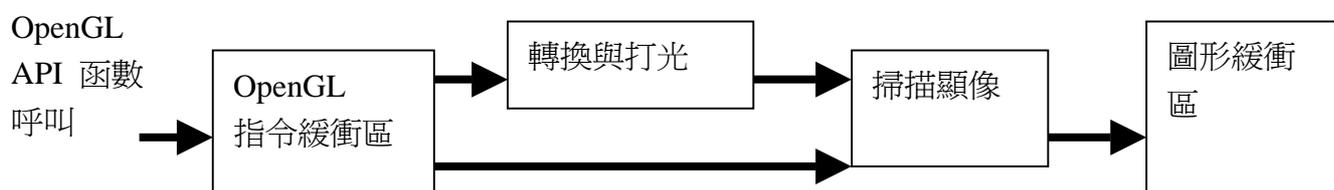


圖 3.4 OpenGL 呈像流程

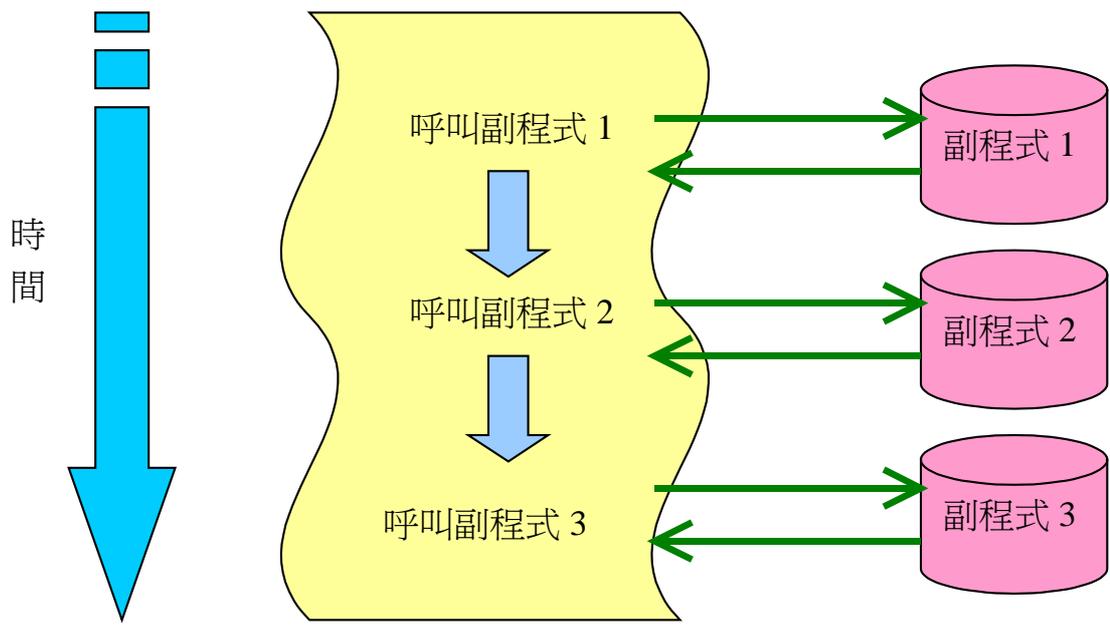
料都會被傳到下一個階段。座標點資料會先被轉換並進行燈光運算，其中用以描述物體幾何位置的座標點會針對物體的位置與方向進行運算，同時也會進行光影計算，以得出每個點上的顏色該有多亮，完成這個程序後，就把資料提供給掃描顯像部份。掃描顯像器將依幾何位置、顏色與材質資料建立影像，然後影像會放到圖形緩衝區內。圖形緩衝區是圖形顯示裝置的記憶體如：繪圖卡、顯示卡，到了這個階段會把影像顯示到螢幕上。

3.2.3 多執行緒技術

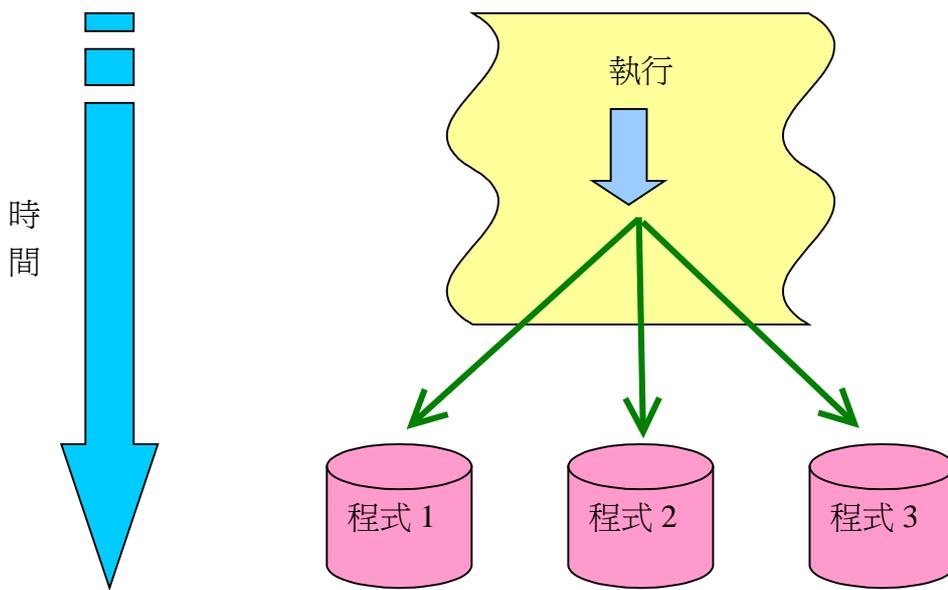


在人類的視覺感受中，爲了要讓人感覺畫面流暢，至少每秒需繪製20到30個畫面，也就是虛擬實境的更新頻率需要20-30Hz左右；在力覺呈現方面，其取樣頻率範圍約在300-1000Hz[4]；所以如果要同時在模擬迴圈中傳送並呈現出視覺及回饋力，讓它在相同的模擬控制迴圈中，會因爲無法提供足夠的頻寬，而可能會造成回饋的力不穩定[8]，讓操控者會因所回饋的力不連續而感覺不適，所以力覺上其更新頻率必須夠快，才可滿足人類對力覺的需求，因此利用多執行緒(Multithreading) 的技巧來解決這問題。

利用執行緒來設計程式，除了提昇效率外，還可以避免讓程式太過複雜，使用此技術，將可以在程式中進行兩個以上作業的平行處理，也就是使程式得以將其工作切開，獨立運作，不互相影響，如圖3.5所示，其中圖3.5(a)爲一般程式設計時的執行順序和時間的關係，圖3.5(b)則爲執行緒程式執行的順序和時間的關



(a) 一般程式設計時的執行順序和時間的關係



(b) 執行緒程式執行的順序和時間的關係

圖 3.5 (a)一般程式設計時的執行順序和時間的關係和(b)執行緒程式執行的順序和時間的關係

係；舉個日常生活的例來說，單緒程式就像大型量飯店中只有一個結帳出口，如果顧客採購東西只有一點點，那麼就可以快速結帳，但若採買的東西很多，則結帳時間就需很久，且其它想結帳的顧客也需等待，而多緒程式就像是有很多個結帳出口，就算一個出口停頓了很久，也不會影響到其它的出口路線。

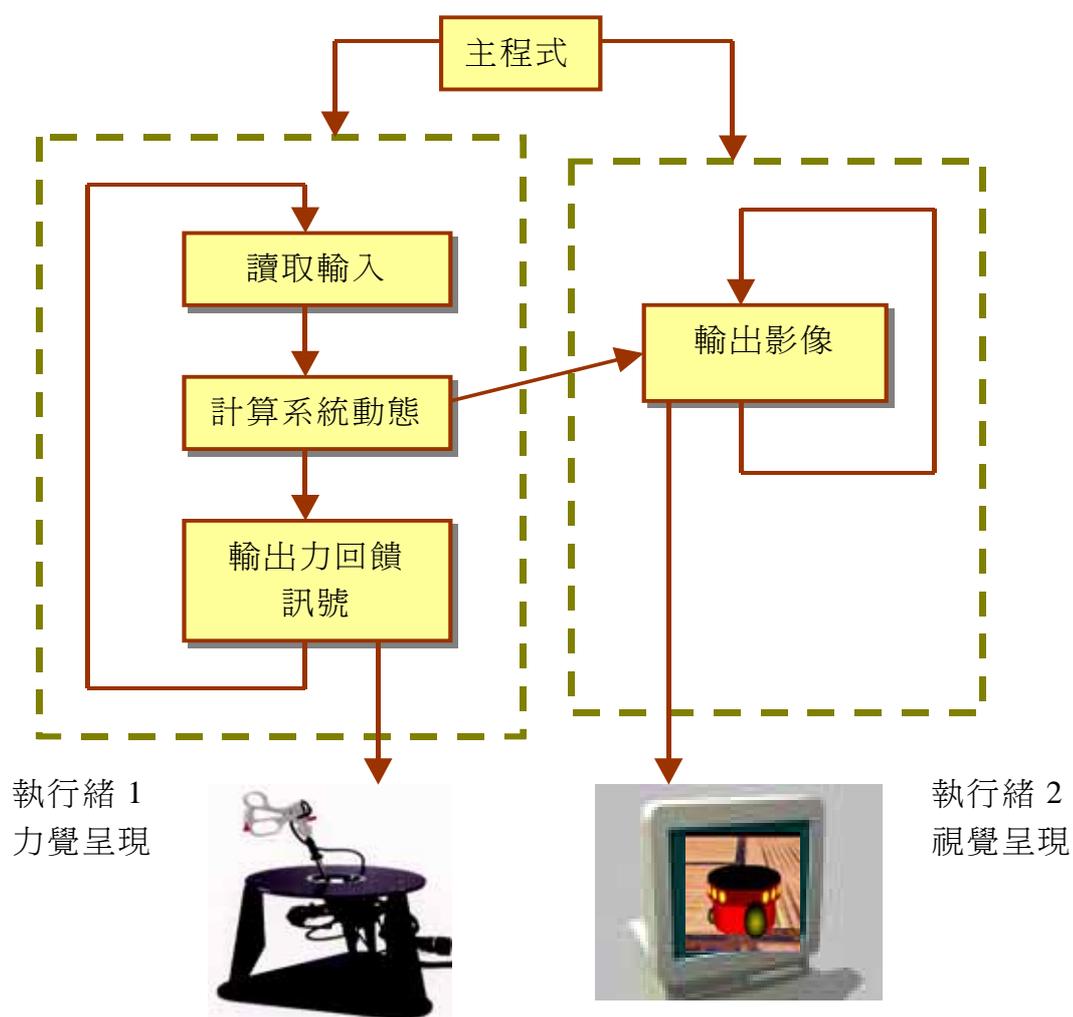


圖 3.6 多執行緒之程式流程圖

在本篇論文中，其所採用的程式流程架構，如圖3.6所示，操作者會先透過力回饋裝置或鍵盤下達控制命令，這些命令進而改變全區域變數的數值，接著處理器再執行這兩個執行緒，分別為處理視覺呈現的執行緒，另一個為處理力覺呈現的執行緒，經由處理器計算後可得場景資訊和力資訊，最後再經由人機介面裝置，將視覺和力覺的感受傳送給操作者來感受體驗。

