

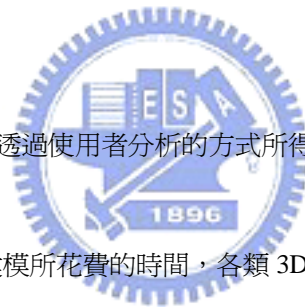
### 第三章 手勢與虛擬建模環境

## Gesture and Virtual Modelling Environment

由於人在空間中的自然行為是最好的虛擬建模操作方式，且手勢動作對於空間設計者又格外重要，因此手勢與虛擬建模環境的整合已是許多研究之研究重點。本章主要在敘述在目前市面上較為流通的虛擬建模軟體的指令行為模式，並根據 Liang(1997)對於自然手勢的描述，將手勢動作類比為建模指令。

### 3.1 虛擬建模程序

根據 Joseph 及 Stuart(1998)透過使用者分析的方式所得到的虛擬建模過程分類，將虛擬建模過程分為三個部分：



- 1) 基本創建：為了減少建模所花費的時間，各類 3D Modelling Programs 都有建立其物件資料庫。過去由於硬體及軟體技術的限制，資料庫往往只能提供方體，柱體，錐體以及球體等少數的選擇。藉由科技的快速發展，目前的 3D Modelling Programs 所提供的基礎物件已經增加為數十到數百不等。透過具規則性的分類方式以及 W I M P 使用上的特性，物件的產生就像從抽屜中拿出積木般容易(Segen 1998)，比起傳統的實體建模，大大的減少了時間及資源的浪費。
- 2) 物件內部資料的修改：基礎物件的建立，很可能不是設計者所要求的形狀，這時往往需要透過一些編修的過程來達到其需求，如物件本身已設定參數的修改，彎曲變形，子物件編修等，以完成對設計者來說最完美的型態，在建模過程中，這個程序往往是最繁瑣及複雜的，也是設計者最能發揮想像的階段。
- 3) 物件外部資料的修改：物件外部資料包括位置(Position)，旋轉(Rotation)，(比例 Scale)，材質(Material)等，在這個步驟中，大多是空間參數的決定，有時也會與內部資料做整合，基本上是建模過程的最後一個階段。

這種三階段完成的模型通常只是一個單元，數個單元的整合即完成最後的模型。這個流程可以包含所有的建模經過，其他關於建模過程的研究，大多也是以這個流程為準則。本研究亦以此流程，另外再加上"物建選取"及"環境瀏覽"兩個項目做為測試數據主要分類方式。

### 3.2 目前主要虛擬建模軟體指令測試

由於目前的虛擬建模軟體幾乎都是針對 WIMP 介面所設計的，因此本小節測試的目的在於了解設計者在使用不同建模軟體下，如何使用目前被普遍使用的 WIMP 介面輸入工具(圖 3-1)進行建模行為。其過程主要是請六名擁有 2 年以上數位建模經驗之受測者，使用目前市面上較為廣為接受的三種虛擬軟體，分別為 Form.Z4.0(RadioZity 2003)，3DS MAX6.0(Discreet Inc 2003)以及 MAYA5.0(Alias|Wavefront Inc 2003)(圖 3-2，3-3，3-4)。記錄重點在於建模的過程中，受測者對軟體介面的操控程度，而關鍵在於指令下達的行為模式。

在受測者完成建模過程中將完整紀錄其指令下達的動作，使用方式，使用特性，使用量以及花費時間。而以下為受測值平均數(單一受測值請參見附錄一)及測試模型如下(圖 3-5)。



圖 3-1 傳統的 WIMP 介面輸入工具

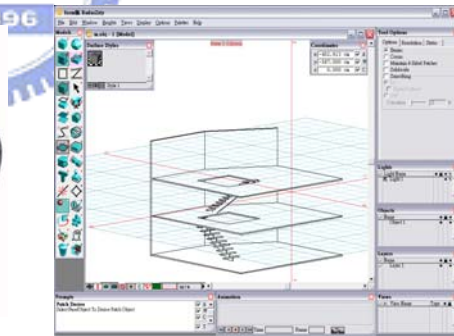


圖 3-2 Form.Z 4.0 建模環境介面

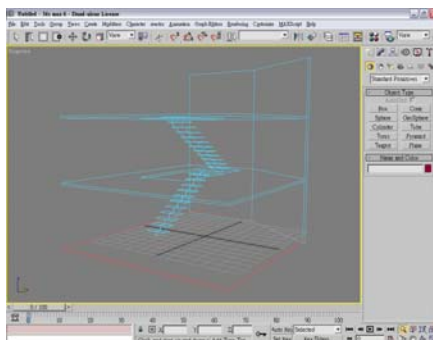


圖 3-3 3DSMAX 6.0 建模環境介面

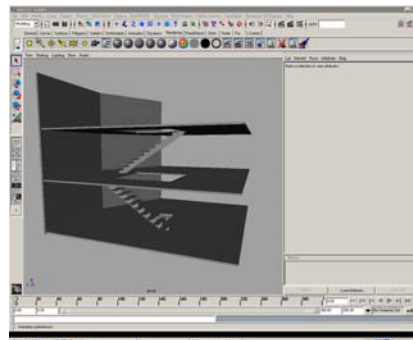


圖 3-4 MAYA 5.0 建模環境介面

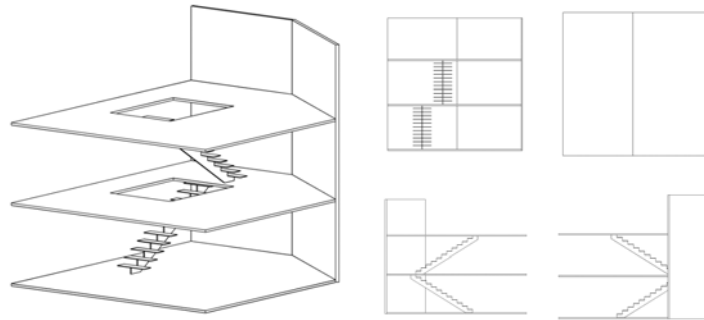


圖 3-5 測試模型

### 3.2.1 基本創建

在分析過程中，使用者動作代碼如下

滑鼠位移(M)，滑鼠左鍵(ML)，滑鼠右鍵(MR)，滑鼠中鍵(MM)，鍵盤輸入(T)，滑鼠滾輪(RB)。

介面輸入(D)，直接輸入(I)，快速面板(E)

建模環境	輸入動作	使用方式	使用特性	使用量	花費時間
3DSMAX	使用創建面板	M,ML	D	3.5	3.5 分鐘
	複製	M,ML,T	I	3.5	
MAYA	使用創建面板	M,ML	D	4	13.5 分鐘
	複製	M,ML,T	I	13.5	
	使用陣列	M,ML,T	D	0.5	
Form.Z	使用創建面板	M,ML	D	6	5.5 分鐘

表 3-1 基本創建測試紀錄

在基本創建的過程裡，由於物件資料庫太過龐大，基本的使用上都是以創建面板為主，而 MAYA 因為物件資料庫內沒有"樓梯"，所以才會大量的使用複製以及陣列，除了這點外，在時間方面並沒有很大的差異。

### 3.2.2 物件內部資料的修改

滑鼠位移(M)，滑鼠左鍵(ML)，滑鼠右鍵(MR)，滑鼠中鍵(MM)，鍵盤輸入(T)，滑鼠滾輪(RB)。

介面輸入(D)，直接輸入(I)，快速面板(E)

建模環境	輸入動作	使用方式	使用特性	使用量	花費時間
3DSMAX	既有參數輸入	M,ML,T	D	9	25 分鐘
	快速面板設定	M,ML,MR	E	6	

	使用修改面板	M,ML	D	10	
MAYA	既有參數輸入	M,ML,T	D	6	25 分鐘
	快速面板設定	M,ML,MR	E	10.5	
	使用修改面板	M,ML	D	6	
Form.Z	既有參數輸入	M,ML,T	D	6	33 分鐘
	使用修改面板	M,ML	D	14.5	

表 3-2 物件內部資料的修改測試紀錄

3DSMAX 與 MAYA 因為修改方式相近，所以在花費時間以及次數上類似，而 From.Z 雖然修改次數較少，但是卻花費更多的時間，原因是在於 From.Z 沒有快速設定的功能，每次的修改都必須在面板中選擇，而 3DSMAX 與 MAYA 將常用的修改，如子物件(sub-object)選擇，物件屬性轉換(convert to)等功能另外設置一個快速面板，藉以增加建模的速度。

### 3.2.3 物件外部資料的修改

滑鼠位移(M)，滑鼠左鍵(ML)，滑鼠右鍵(MR)，滑鼠中鍵(MM)，鍵盤輸入(T)，滑鼠滾輪(RB)。介面輸入(D)，直接輸入(I)，快速面板(E)

建模環境	輸入動作	使用方式	使用特性	使用量	花費時間
3DSMAX	位置參數輸入	M,ML,T	D	5	17 分鐘
	透過熱鍵直接修改	M,ML,T	I	6	
	點選面板移動功能	M,ML	I	2	
MAYA	位置參數輸入	M,ML,T	D	3	33 分鐘
	右鍵快速設定	M,ML,T	E	9.5	
	點選面板移動功能	M,ML	I	1	
Form.Z	位置參數輸入	M,ML,T	D	6	29 分鐘
	點選面板移動功能	M,ML	I	6	

表 3-3 物件外部資料的修改測試紀錄

這個階段的情況與物件內部資料的修改時有些類似，比起在 From.Z 上所有的修改都需要透過修改面板的點選，3DSMAX 與 MAYA 可以使用熱鍵或是直接藉由快速面板的設定，讓使用者能更快速的修改物件外部資料。表面上 MAYA 所花的時間比 From.Z 更長，但是其實 MAYA 所需處理的物件是 From.Z 與 3DSMAX 的三倍，在實際的操作上是更快速的。

### 3.2.4 物件選取

滑鼠位移(M)，滑鼠左鍵(ML)，滑鼠右鍵(MR)，滑鼠中鍵(MM)，鍵盤輸入(T)，滑鼠滾輪(RB)。  
介面輸入(D)，直接輸入(I)，快速面板(E)

建模環境	輸入動作	使用方式	使用特性	使用量	花費時間
3DSMAX	直接選取物件	M,ML	I	43	4.5 分鐘
	透過面板選取物件	M,ML	D	0.5	
MAYA	直接選取物件	M,ML	I	45.5	5.5 分鐘
	透過面板選取物件	M,ML	D	2	
Form.Z	直接選取物件	M,ML	I	42	8.5 分鐘
	透過面板選取物件	M,ML	D	3.5	

表 3-4 物件選取測試紀錄

在選取方面，所有的受測者都會有"無意義選取"的行為，就是點選物件後，在沒有執行其他的動作的情況下，又再選取另一個物件，這是建模過程中極普遍的行為，此外並沒有任何特殊的情況。

### 3.2.5 環境瀏覽

滑鼠位移(M)，滑鼠左鍵(ML)，滑鼠右鍵(MR)，滑鼠中鍵(MM)，鍵盤輸入(T)，滑鼠滾輪(RB)。  
介面輸入(D)，直接輸入(I)，快速面板(E)

建模環境	輸入動作	使用方式	使用特性	使用量	花費時間
3DSMAX	熱鍵瀏覽	M,MM,T	I	115.5	4.5 分鐘
	縮放視窗	RB	I	41.5	
	使用環境面板	M,ML	D	8	
MAYA	熱鍵瀏覽	M,MM,T	I	153	5.5 分鐘
Form.Z	使用環境面板	M,ML	I	48	8.5 分鐘

表 3-5 環境瀏覽測試紀錄

而在環境瀏覽上，From.Z 的受測者在次數上明顯較少，主要是因為瀏覽動作太過麻煩，就算如此，他們在瀏覽動作上所花的時間還是比 3DMAX 與 MAYA 的受測者更多。

### 3.2.6 分析結論

從六位測試者的使用情形來看，在每一種虛擬建模軟體中，幾乎所有的指令行為都包含了M及ML的動作，在這個情況下，可以把所有的指令行為分成"只有M及ML動作的指令"以及"除了M

及ML外還有其他動作的指令"兩種，而每一種又包含直接輸入(使用特性為I)與間接輸入使用特性為(D)兩個分類(表3-6)。

	直接輸入(使用特性為I)	間接輸入使用特性為(D)
只有M及ML	單純的選擇或抓取動作	使用介面功能
除了M及ML外還有其他動作	使用快速鍵	參數輸入

表 3-6 指令行為分類

此外，還有一些特殊的指令下達方式(如滑鼠滾輪)，由於其在使用上是各個軟體針對不同需求所設計的，因此無法對此類指令進行分類，但是基本上這些指令都是以直接輸入輸入為主。因此，我們可以把所有的指令行為分為直接輸入與間接輸入兩部分，向下細分後又可在分為五種類型：

#### 直接輸入

- 1) 選擇或抓取：選取是最常被使用的動作，幾乎每個指令都必須配合它使用；抓取動作則常被應用在物件外部指令的控制及環境瀏覽上。
- 2) 快速鍵輸入：對軟體熟悉的人才會使用的方式，一般來說都是常用的指令，不過可以根據使用者的使用情形自行調整。
- 3) 其他指令：依照軟體的不同而有不一樣的輸入方式，如滑鼠中鍵及右鍵的應用，通常為直接指令。

#### 間接輸入

- 1) 介面輸入：通常為不常用或複雜指令的輸入方式，大都為創建或物件內部時使用。
- 2) 參數輸入：與介面配合，需要精準數值時使用，一般只需用到0-9，小數點，倒退等十二種按鍵。

從三種軟體的操作方式以及時間上看，快速鍵的使用可以幫助使用者大幅的減少建模時間，但是由於鍵盤的組合方式太過多樣化，而其中又有太多無用的按鍵，對初學者來說，除了一些好記的快速鍵(比如快速面板呼叫)外，其他幾乎沒有任何幫助。為了讓虛擬建模軟體的介面更簡單易學，如何好好的利用快速鍵輸入將會是一個重要的關鍵。

### 3.3 手勢分析

手勢通常來說指的就是手指間的動作與手掌位置所組合出的一種溝通模式(Liang,1997)。我們常見到一般人們使用簡單的手勢做為彼此間訊息的傳遞。有關研究人類如何使用手勢來溝通，已有一段漫長且豐富的歷史，這段歷史可追溯至十七世紀。人類使用手勢有二個基本目的—溝通目的



及操控目的(Nemeth 1984)。一個管弦樂指揮的手勢是用傳達節拍與控制訊號給管弦樂團。鋼琴家的手指動作是用來彈奏鍵盤，而不是用來與人溝通。以手勢來溝通一般是依靠視覺來解釋其不同比劃所代表的意義(Liang,1997)。然而這些手勢在不同的環境之下，對於手勢的解譯時，會產生一些不確定性的因素。因此，自然手勢的使用就格外重要。

### 3.3.1 一般規則

Liang (1997)把手勢模式分為靜態的手勢與動態的手勢。前者能被手指，拇指，手掌的狀態來區分，也就是手的姿勢可用於拼音。後者則可經由最初的手勢，最後的手勢與動作之軌跡來區分。當手在運動中時，手的狀態會改變，但這變化不包含任何手勢資訊，因此此時手的狀態可被忽視。手勢開始時，以一個較慢的速度從起始的位置出發，繼之以一個動作的軌跡，最後結束回到手勢開始時起始的位置。Liang 分析出的一般手勢示意之動作規則如下：

- 1) 手勢開始時，以一個較慢的速度從起始的位置出發，繼續以一個較快之運動，最後結束回到手勢開始時起始的位置。
- 2) 當這隻手在運動中，假設其處於一個特定的狀態。
- 3) 當這隻手在靜止的位置時的慢動作不被認定是作手勢。
- 4) 手勢被限制在一個固定的活動範圍內。
- 5) 靜態的手勢需要在一個有限的時間固定其姿勢。
- 6) 重覆的動作可能是作手勢。
- 7) 當這隻手是靜止時，個別的手指的運動可能有其意義。

透過這些規則所建立出的手勢系統，幾乎可以傳達所有的訊息，可是大部分的手勢都必須過一定程度的訓練，才能明白該手勢所代表的指令為何，不僅不易學習，反而還會影響軟體的操作，並不是一種良好的虛擬建模操作模式。

### 3.3.2 自然手勢

為了減少學習時間以及動作與指令轉換間的不直覺，自然手勢是必要的，但是什麼是真正的自然手勢，在文獻中並無十分清楚的定義。Liang 認為基本手勢動作包含手的結構和手的運動，因此手勢動作是包括了時間上和空間上手形變化的分析。Liang(1997)依照手勢需要訓練的程度將其分為三種類型：

- 1) 手掌位置的改變與具指向性的動作(不需訓練)(圖 3-6)。
- 2) 拇指與其他四指間的交互動作(經由簡單的訓練即可使用) (圖 3-7)。
- 3) 手掌的變化配合手臂整體的動作(需要長期的訓練) (圖 3-8)。

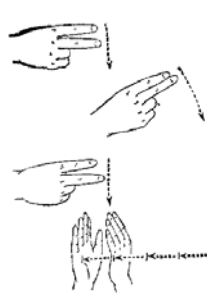


圖 3-6 不須訓練的手勢

(取自 Liang 1997)

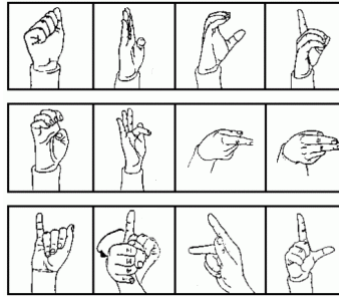


圖 3-7 經由簡單的訓練即可使用

(取自 Liang 1997)

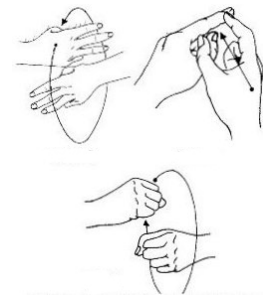


圖 3-8 需要長期的訓練的手勢

(取自 Liang 1997)

基於本研究的需求，本研究將第一與第二類型視為一種自然手勢，並作為本研究媒材手勢系統之基本依據。

### 3.4 手勢與虛擬建模軟體整合

在本研究中，手勢與虛擬建模軟體整合的主要目的，是為了讓目前較為被廣泛使用的虛擬建模軟體的使用者能更接近 Donath (1999)所提出的良好虛擬輔助環境介面操作應具備之條。因此，根據本研究所定義的自然手勢以及虛擬建模軟體指令測試的結果(詳見 3.2)，本研究將自然手勢與虛擬建模軟體整合的結果如下：

#### 直接輸入(本研究主要以手勢取代的部分)

- 1) 選擇或抓取：因為此種動作不需要明確的命令修改訊息，而且能立即的到回饋，在虛擬建模環境裡，應該將使用者的日常習慣反映在這個指令動作的使用上，所以可以用第一類型的自然手勢(手掌位置的改變與具指向性的動作)取代。
- 2) 快速指令輸入：這種動作雖然是一種直接指令的下達，但是在使用上需要預先學習，操作方法又不容易記憶。不過從使用量上看來，熱鍵輸入對於使用者的幫助確實很大，因此，本研究將以第二類型的自然手勢(拇指與其他四指間的交互動作)。
- 3) 其他指令：必須根據不同的軟體定義不同的輸入方式。

#### 間接輸入(以軟體本身的 WIMP 介面為主)

- 1) 介面輸入：介面輸入本身包含龐大的指令系統，基本上在目前的虛擬建模環境中，除了使用其基本的 WIMP 介面，並無一種良好的解決替代方案。在本研究中，會將此部分做一個保留。
- 2) 參數輸入：基本上是一種符號輸入(如數字或字母)，但是由於無法與自然手勢作類比，所以本研究定義其為一種介面輸入的指令行爲。