

第一章

導論

在傳統教學上，學生習字都是透過教師在黑板上示範，接著學生模仿教師的書寫動作進行學習，由於在課堂上是一對多教育，加上有學習時間限制，老師無暇顧及每位學生，即使從批改學生的作業本上，也只能看出字形的外觀，無法從紙本上判斷出筆劃順序是否正確。Law 等[1]提出，學生常有因為筆畫順序的錯誤而在書寫上出現筆畫不完整的情況，為了解決上述因素，並讓學習者能自行正確的練習，我們希望設計一能立即與學習者互動的書寫系統。

在早期的研究，重心放在視覺上，多靠視覺認識中文字，而沒有實際拿起筆進行操作，這方面的相關研究有 Lam 等[2]設計的電腦輔助教學系統，此系統透過撥放文字、動畫與學習者進行互動，如圖 1-1 所示；後期開始有了以筆作為輸入設備的研究，例如 Tan [3]透過電子手寫板作為學習者的輸入工具，經由作者提出的演算法判斷學習者的軌跡是否正確，如圖 1-2 所示。近來越來越多力回饋設備被設計成似筆的形狀，許多研究者開始運用此設備加上導引法，改善先前無法實際操作的缺點，這方面的相關研究有 Eid 等[4]使用力回饋搖桿搭配所設計的導引策略，發展出線上學習平假名五十音系統，如圖 1-3(a)所示，以及 Palluel 等[5]使用力回饋搖桿搭配導引策略，教導幼稚園兒童學習基本英文字母，如圖 1-3(b)所示，以上 Tan[3]、Eid 等[4]和 Palluel 等[5]的研究領域都是屬於硬體字書寫範圍，在軟體字的領域也有相關研究發表，如 Henmi 和 Yoshikawa[6]使用二組機械手臂和二個力感測器組成的機構設計出虛擬毛筆字系統，如圖 1-4(a)所示，以及 Teo 等[7]使用市售力回饋搖桿 delta haptic device 設計出硬體字和毛筆字二種字

體的虛擬學習系統，如圖 1-4(b)所示。

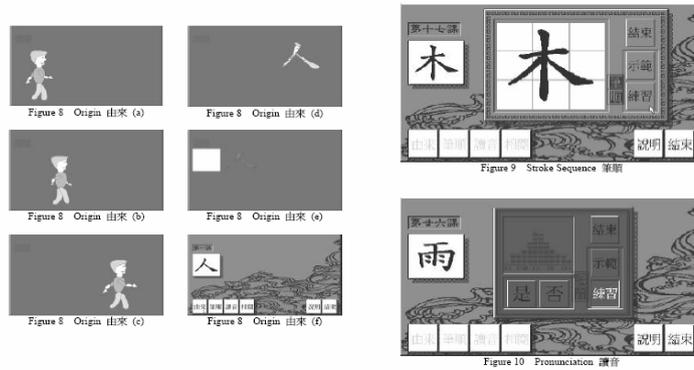
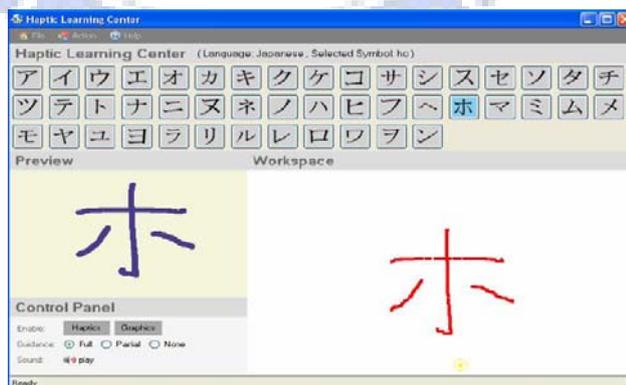


圖 1-1 電腦輔助教學系統[2]

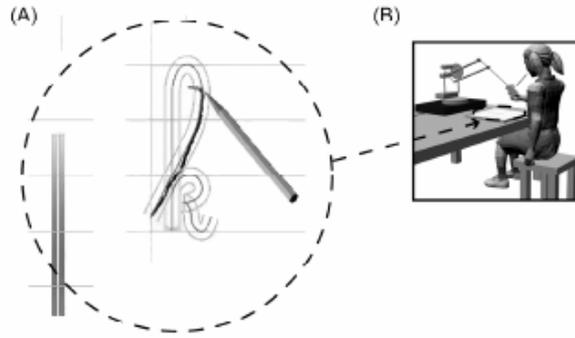


圖 1-2 中文硬體字手寫辨識系統[3]



(a) 手寫日文五十音學習系統[4]

圖 1-3 具力感輔助硬體字手寫系統(續)

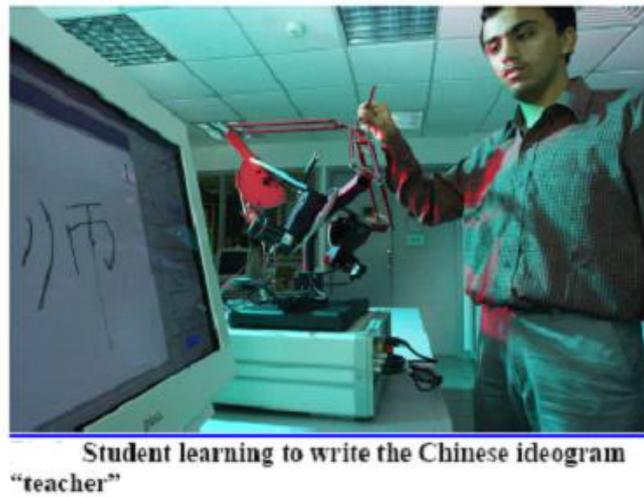


(b) 手寫英文字母學習系統[5]

圖 1-3 具力感輔助硬體字手寫系統



(a) 虛擬毛筆字系統[6]



(b) 中文字虛擬教學系統[7]

圖 1-4 具力感輔助軟體字手寫系統

在具力感輔助軟體字書寫領域的兩篇相關研究 Henmi 和 Yoshikawa [6]以及 Teo 等 [7]，在書寫紙張方面皆以系統模擬達成目的，作法為藉由設定力搖桿垂直方向的操作空間在某個高度以上，當搖桿位置在預設的高度以下時，透過力搖桿產生於施力點方向的反作用力，稱為虛擬紙(virtual paper)，而墨表現方式是以操作者穿透虛擬紙的深淺來決定像素的多寡，並顯示在虛擬場景中，如圖 1-5 所示，兩篇研究在引導方式的作法分為：(1)路徑限制(Path constraint)，操作者被限制在預設的操作路徑上，在路徑上的移動方向及速度不受到限制；(2)動作引導(Motion guidance)，學習者經由力的引導完成操作的動作，而操作的方向速度都預先規劃。此兩篇研究，在筆、墨和紙方面都以系統模擬方式呈現，相對地，本系統則以實際的筆、墨和紙呈現，搭配引導策略，提供給操作者更接近真實的情境，並且希望不同程度的操控者能夠藉由輔助模式幫助自身完成特定毛筆字的書寫，進而從輔助過程中學習特定毛筆字的書寫方法。

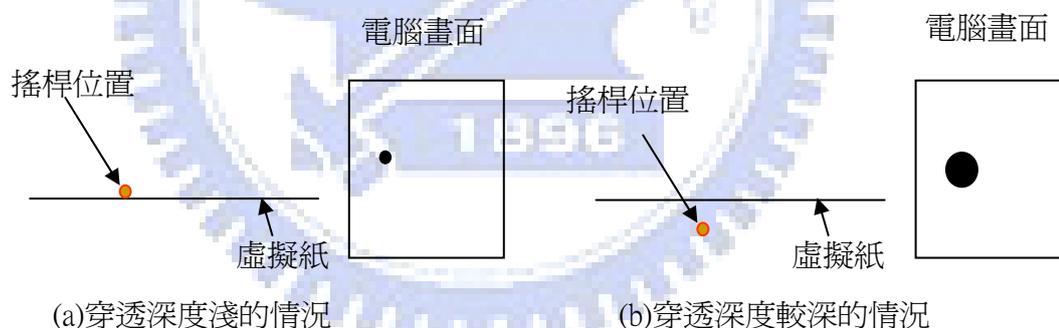


圖 1-5 虛擬紙示意圖

由於毛筆字書寫需要三個自由度的操控，為了實現本系統，我們利用機器臂 Phantom omni 三軸力回饋搖桿作為輸入設備，此搖桿能夠提供三個自由度方向的操作和力感回饋，以及搖桿的操作柄外觀為筆，對於模擬寫字所用的工具來說，很適合作為本系統的控制設備，對於毛筆字書寫方面，寫的好的重點在於操作者的書寫路徑及毛筆深度方面的控制，因此我們請有書法學習經驗的示範者示範特定毛筆字，再以該書寫路徑作為操

控者的操作路徑，並且設計了三種輔助模式適合不同程度的操作者，分別為：(1)路徑限制模式，所適用對象為方向、速度控制上具有不錯的操控能力，但路徑操控能力欠佳的操控者，此模式提供操控者特定書法字的操作路徑；(2)具阻尼路徑限制模式，適用對象為方向、速度控制上具有不錯的操控能力，但在轉彎處的操控能力欠佳的操控者，此模式讓操控者操作在轉彎路徑上時，具有穩定的操控感，以及(3)路徑引導模式，對象為方向、速度和路徑操控能力都欠佳的操控者，此模式能夠引導操控者以適當的速度和正確的方向操控在路徑上，實驗結果部份，我們以操作者與示範者軌跡的誤差值和書寫結果進行討論。本論文將於第二章介紹我們所設計的毛筆字書寫輔助系統，輔助模式的種類，以及各輔助模式下所用到的輔助策略；第三章說明實現毛筆字書寫輔助系統的軟硬體；第四章利用所發展的毛筆字書寫輔助系統進行實驗；最後，第五章提出結論以及未來發展的方向。



第二章

毛筆字書寫輔助系統

本章的內容主要在敘述毛筆字書寫輔助系統如何透過力輔助策略對使用者進行力輔助，在毛筆字書寫輔助系統中，使用者乃是透過力回饋搖桿與虛擬環境進行互動，其中繪圖軟體負責建立虛擬場景給予使用者視覺上的回饋，毛筆字書寫輔助系統同時在虛擬場景中建立虛擬力輔助，藉此給予使用者力覺上的輔助，透過虛擬力輔助，我們得以改善操作者在毛筆字書寫上的成果，我們將毛筆字書寫輔助系統分為虛擬場景建構以及力輔助模式二部份，如圖 2-1 所示，系統首先透過虛擬場景建構，將示範者的操作軌跡交由繪圖軟體繪製，並且顯示在電腦螢幕上，作為操作者與系統之間的人機介面。操作者透過人機介面送出控制指令選擇力輔助模式，接著力輔助模式開始判斷操作者是否在力輔助範圍內，並且決定是否對操作者提供虛擬力輔助，幫助操作者達到軌跡復現的目的。接下來我們分別介紹虛擬場景建構和力輔助模式的流程，在虛擬場景建構部份，說明取得示範者的操作軌跡後，將其調整成為我們所想要的軌跡形式，接著經過字體筆劃切割的步驟，將字體的筆劃切割成適合搭配力輔助策略；在力輔助模式部份，將說明各輔助模式如何對操作者進行輔助，以及各模式所使用的力輔助策略，及其輔助力演算法。

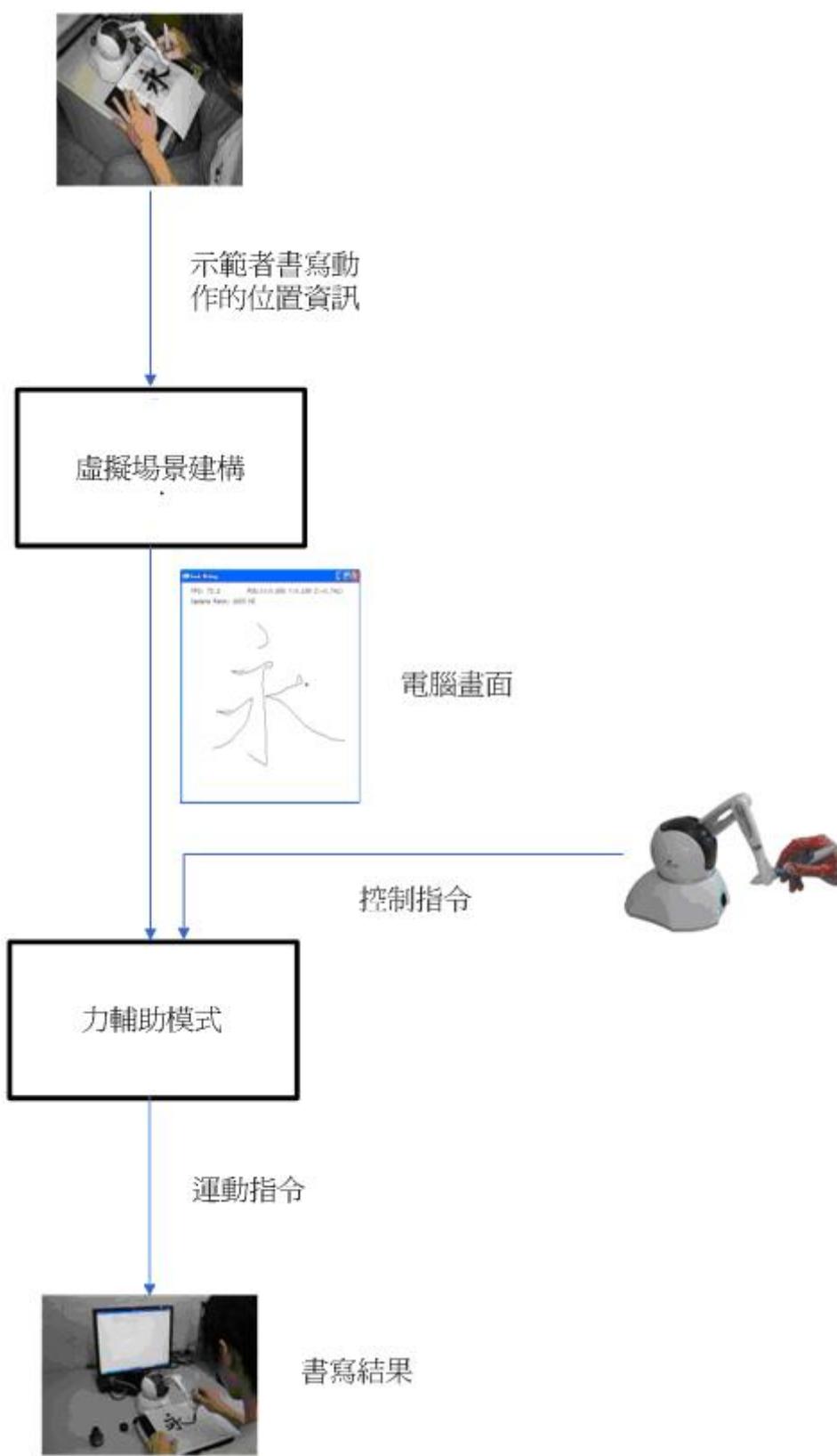


圖 2-1 毛筆字書寫輔助系統流程圖

2.1 虛擬場景建構

在建構虛擬場景前必須先取得示範者的操作軌跡，因此我們經由示範者操作搖桿，以示範書寫毛筆字的方式記錄示範者的操作軌跡，並且調整成為適合的軌跡形式，接著經由字體筆劃切割的步驟後，再以程式設計軟體 visual c++ 以及 OpenGL 繪圖函式庫來建構虛擬場景，流程如圖 2-2 所示。

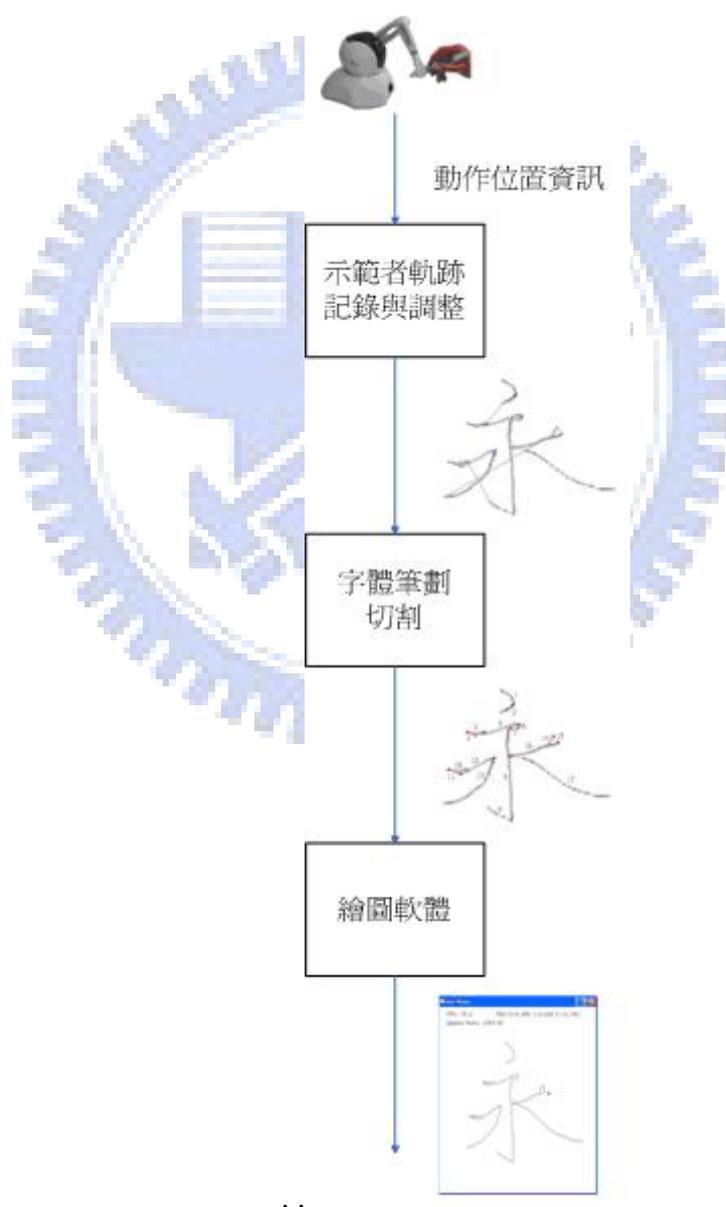
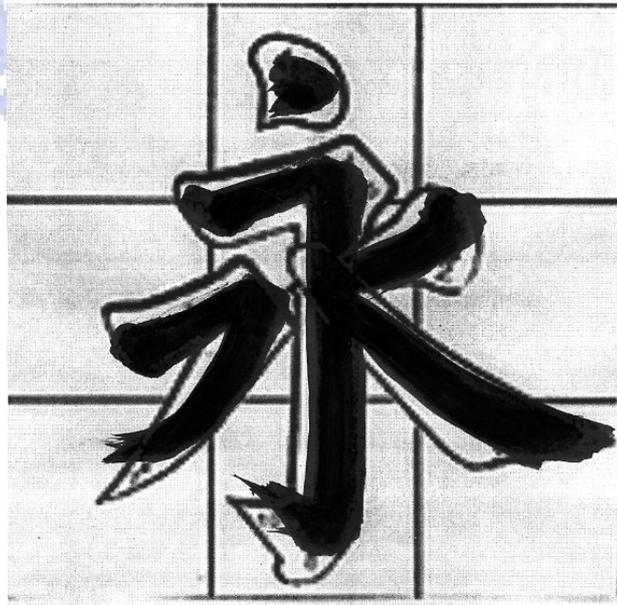


圖 2-2 虛擬場景建構流程圖

2.1.1 示範者軌跡記錄

藉由示範者在無輔助的情況下操作搖桿，在紙上示範書寫所要建構在虛擬場景中的字型，同時記錄搖桿座標，座標點的取樣頻率為 0.1Hz，此選擇有幾個考量，(1)若取樣點數過多，軌跡的調整會相當繁瑣；(2)軌跡是以線段連接各個取樣點的方式建構於虛擬場景中，如果取樣點數增加，會使得各點間距減少、線段數增加，對於軌跡建構沒有實質上的助益。我們請示範者示範了四次相同字型，嘗試以取樣頻率在 0.1Hz 的情況下，取得四次示範者軌跡，如圖 2-3 所示，雖然這四次的書寫以結果來看似無明顯差異，但仔細觀察軌跡卻不盡相同，會造成這樣的原因是由於取樣點的位置是搖桿操作柄的尖端，而示範者書寫的位置卻是連接在操作柄尖端的毛筆頭，當示範者操控搖桿操作柄的角度與紙面呈現傾斜時，那麼取樣點在水平面上的位置將會與毛筆頭的位置不同，為了避免此情況出現，我們請示範者操控操作柄的角度盡量與紙面呈現垂直，如圖 2-4 所示。



(a) 示範者示範結果

圖 2-3 示範者書寫結果(續)

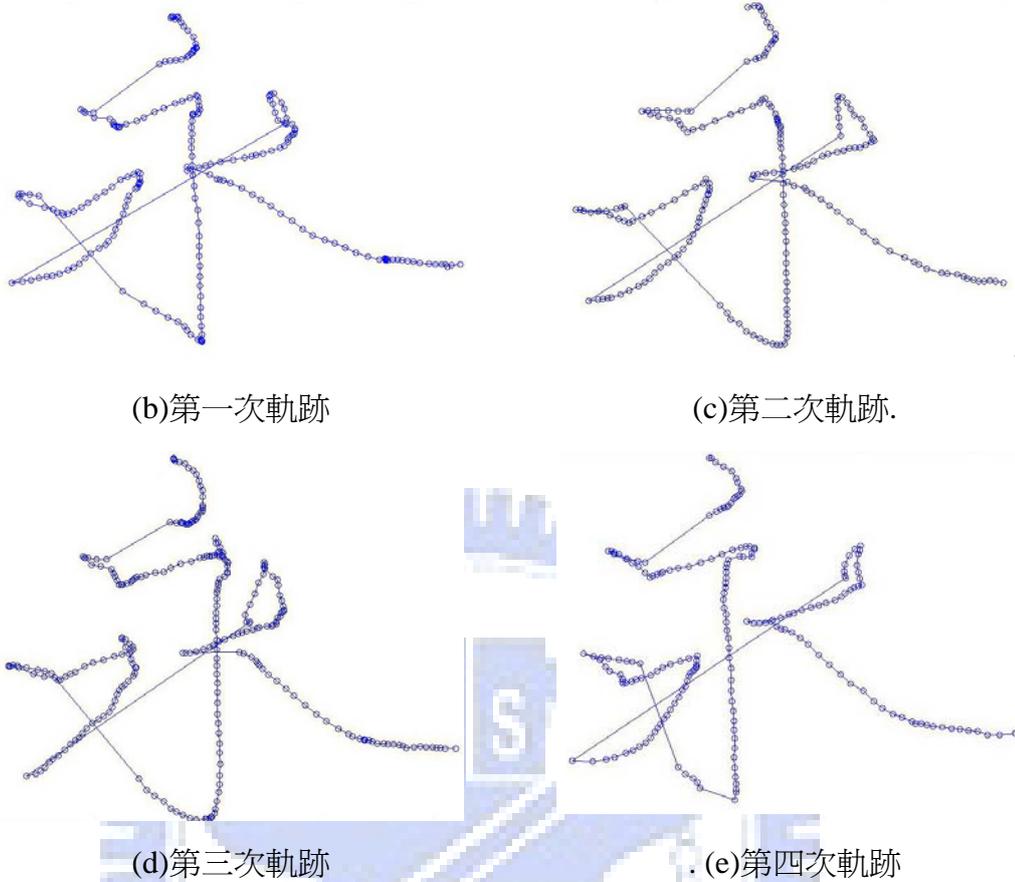


圖 2-3 示範者書寫結果

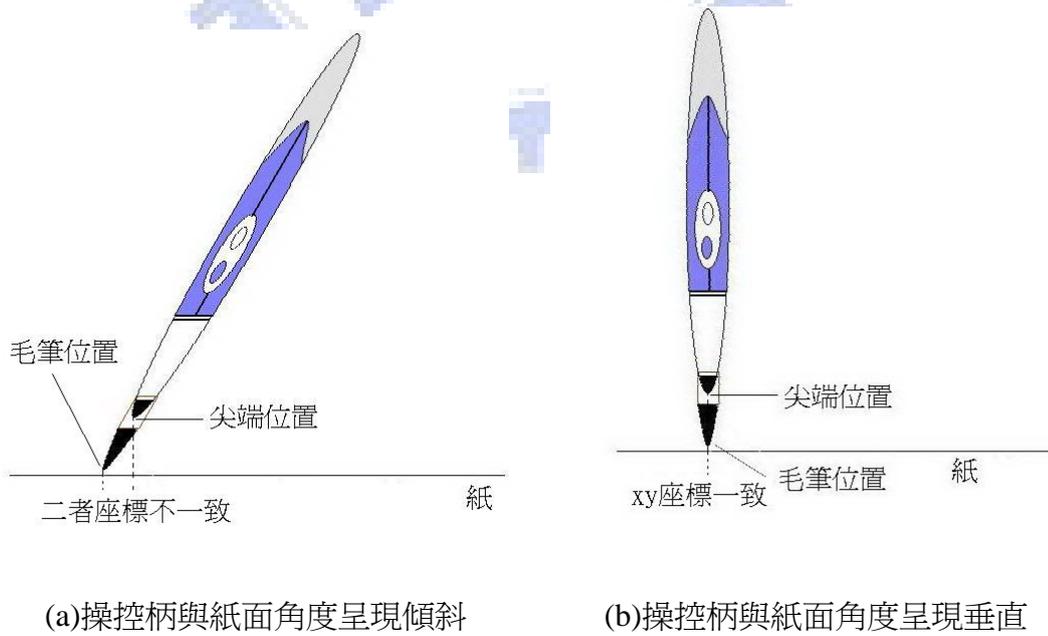


圖 2-4 操作柄與紙面的角度影響取樣結果的情形

我們嘗試以示範者的四次軌跡直接建構在虛擬場景中，作為操作者的學習模型，書寫結果，如圖 2-5 所示，會造成如此不好的結果，除了操作柄的角度與紙面呈現傾斜造成取樣點在水平面上的位置將會與毛筆頭的位置不同，使得學習路徑與示範者實際操作路徑不同，另外的原因是由於示範者非意識的手部細微動作所造成，導致某些取樣點與鄰近取樣點的位置差異太大，讓原本預期的操作軌跡是條平順的曲線，卻在非意識手部動作影響下，造成操作軌跡不如預期，進而影響到書寫結果，如圖 2-6 所示，為了解決以軌跡直接建構在虛擬場景中所會面臨的問題，我們選用示範者所示範的四次軌跡中，相對於其它三個來說較好的軌跡，並且調整此軌跡中因為示範者非意識手部細微動作所造成的取樣點，使操作軌跡都能夠呈現平順的操作路徑，如圖 2-7 所示。

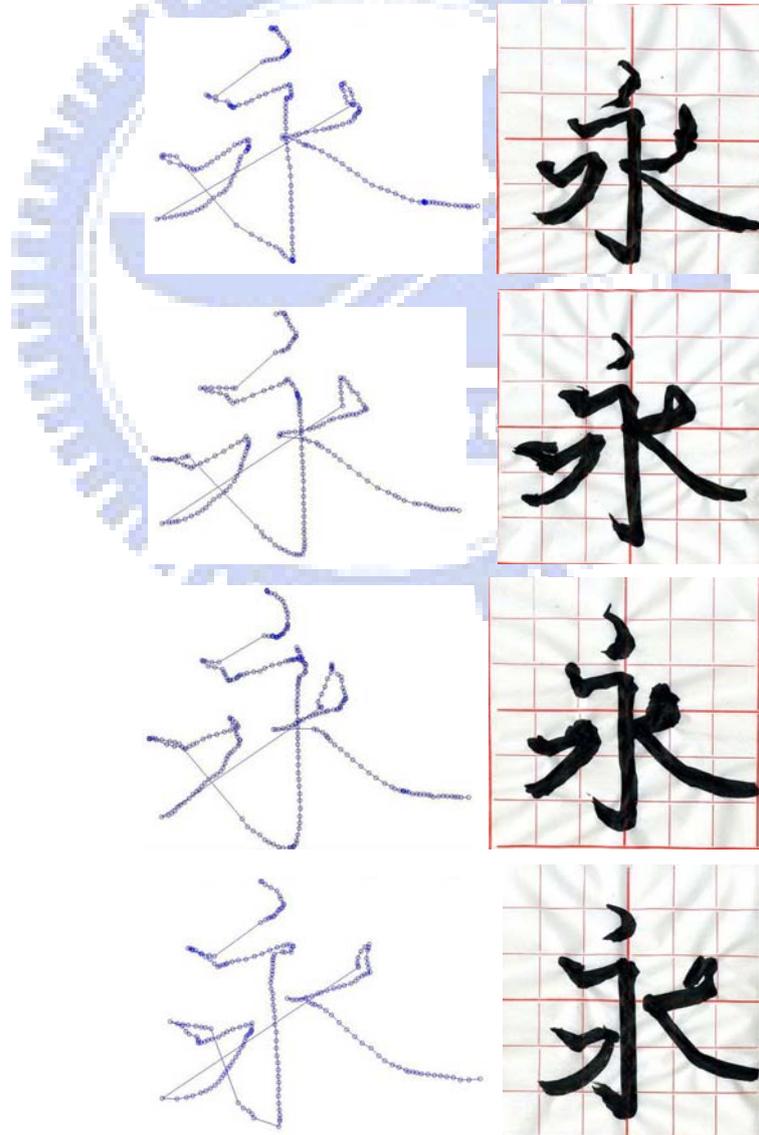


圖 2-5 以示範者的四次軌跡直接建構模型的書寫結果

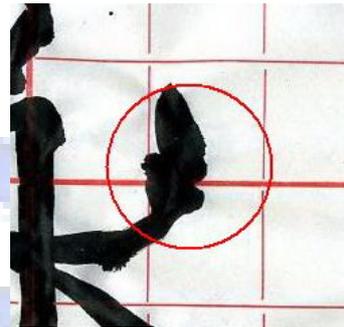
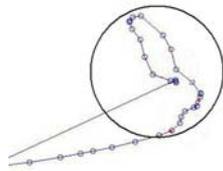
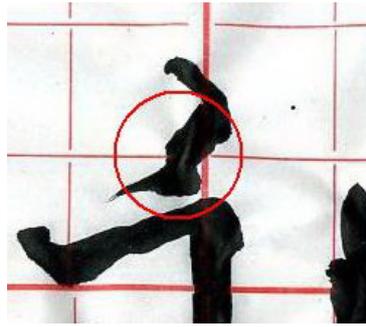
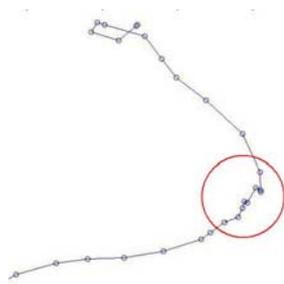


圖 2-6 示範者非意識的手部動作所造成的影響

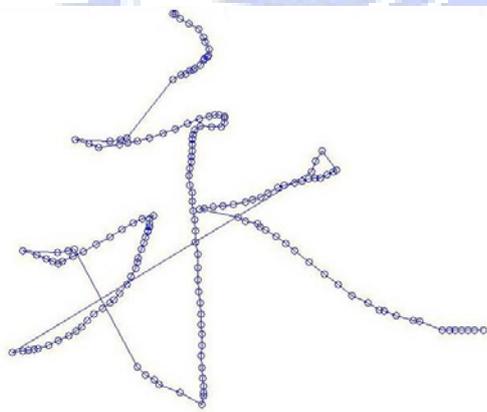


圖 2-7 調整後軌跡與書寫結果

2.1.2 字體筆劃切割

在記錄示範者的軌跡，並且經過調整以達到我們所想要的軌跡形式後，在將軌跡資訊傳遞給繪圖軟體繪製在虛擬場景中之前，需先經過字體筆劃切割的程序，字體筆劃切割的目的是為了使我們所設計的輔助策略容易實現，雖然即使沒有經過此程序也能建構成虛擬場景，但是對於我們所設計的輔助策略，在設計上會遭遇一些問題，稍後會進行討論。字體筆劃切割所針對的地方除了一個字的正規筆劃外，重點在於筆劃轉折處，切割方法為一筆劃中遇到轉折或呈現非直線方向即視為一筆劃，本論文示範者所示範的永字第二劃中有四個轉折點，因此將第二筆劃再細切割為五劃，如圖 2-8 所示。

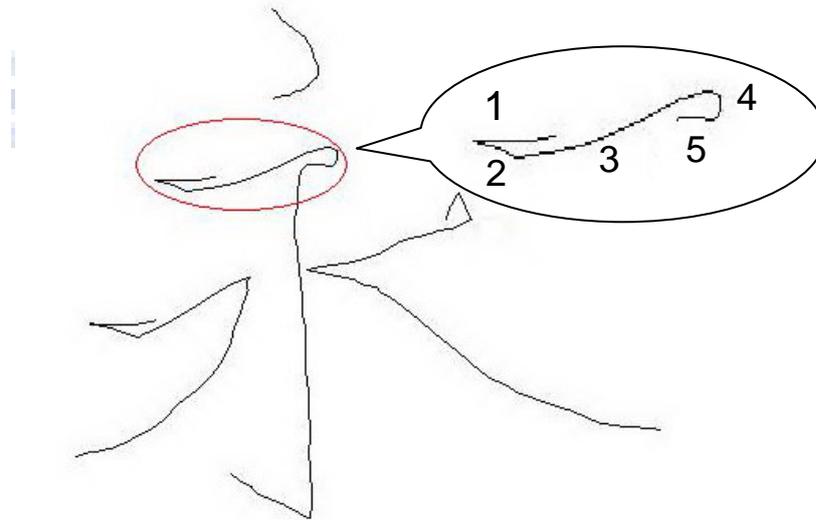


圖 2-8 筆劃切割示意圖

2.2 力輔助模式

操作者能夠經由系統所提供的力輔助模式，及所搭配的輔助策略進行書寫操作，操作者透過人機介面對系統下達控制指令，告知系統操作者所選擇的力輔助模式，選取力輔助模式後，系統會判斷操作者是否在力輔助範圍內，來決定是否對操作者施以力輔助策略；而虛擬實境以視窗的方式顯示在電腦螢幕上傳達給操作者，讓操作者能同時觀察到搖桿在虛擬場景中的位置，同時操作者能觀察在紙上的書寫情況，流程如圖 2-9 所示，之後分別介紹我們所設計的三種力輔助模式(1)路徑限制，(2)具阻尼路徑限制，和(3)路徑引導，以及各自所使用的輔助策略。

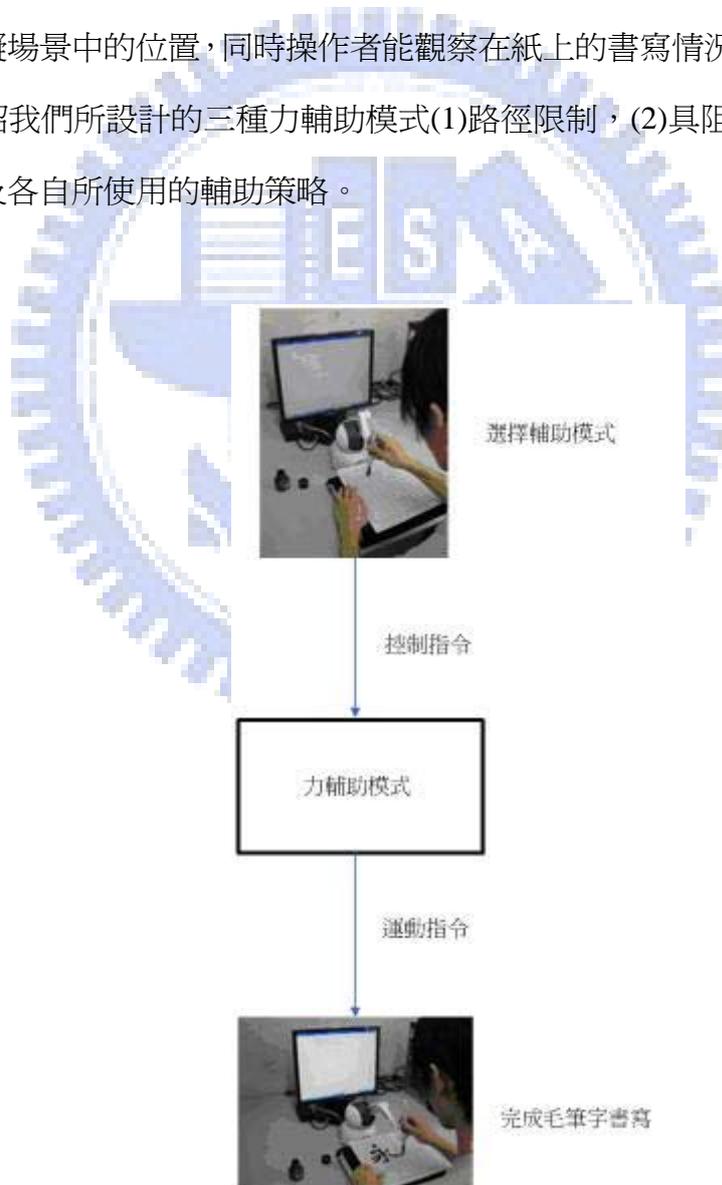


圖 2-9 力輔助模式流程圖

2.2.1 路徑限制模式

路徑限制模式的目的是為了使操作者能夠隨著筆順引導，依序在預設的路徑上進行操作，藉此練習字的結構，各筆劃的間距以及深淺程度。在本模式中，我們不預設操作者在路徑上的操作方向和速度，模式中所用的路徑限制策略目的在於使操作者的操作範圍在預設路徑上，當操作者操作在路徑上時，路徑和搖桿間會產生黏滯力，使搖桿能吸附在操作路徑上，一旦操作者試途離開操作路徑時，路徑限制策略會計算出力向量將操作者拉回操作路徑上；而筆順引導策略的目的在於引導對於該字的筆劃、結構不熟悉的操作者，操作在正確的筆順上，如圖 2-10 所示，本模式所提供的力輔助屬於被動形式。



圖 2-10 路徑限制模式示意圖

在介紹過路徑限制模式的輔助程序後，接著介紹本輔助模式中所採用的兩項輔助策略以及在章節 2.1.2 字體筆劃切割中未談及經過筆劃切割程序與否，對於所採用的輔助策略在設計上有哪些幫助和困難。本模式中路徑限制策略的設計作法為，將示範者的軌跡以線段的方式繪製在虛擬場景中，而這些線段就是操作者在虛擬場景中的操作路徑，當搖桿在虛擬場景中的位置，位於某個操作路徑能對操作者進行該輔助策略的作用範圍內，該輔助策略會產生力向量迫使操作者的操控方向往該路徑移動，而力向量的大小決定於搖桿和操作路徑上與搖桿最近的點之間的距離，關係式為

$$\vec{F} = \begin{cases} k(pos - jpos) & \text{if } jpos < pos + d \\ 0 & \text{if } jpos > pos + d \end{cases} \quad (2.1)$$

其中 k 為常數， pos 為操作路徑上距離搖桿位置最近的點， $jpos$ 為搖桿的位置， d 為該輔助策略所能對操作者進行力輔助的最大距離。當輔助策略產生力向量迫使搖桿的位置停留在操作路徑上時，轉而在搖桿與操作路徑接觸點間產生黏滯力，使搖桿不能輕易的脫離操作路徑，當搖桿的距離大於輔助策略能對操作者進行力輔助作用的範圍時，該輔助策略停止對操作者進行力輔助，如圖 2-11 所示。

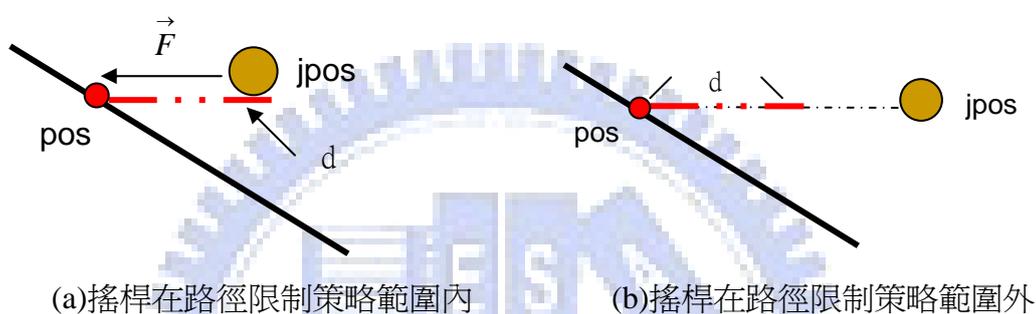


圖 2-11 路徑限制策略示意圖

另外一個策略為筆順引導策略，首先我們先說明筆順引導策略的作用，接著討論經過筆劃切割程序與否，對於筆順引導策略在設計上有哪些幫助和困難。筆順引導策略必須與路徑限制策略搭配使用，路徑限制策略的設計用來限制操作者的操作範圍，而筆順引導策略使操作者操作在正確的筆劃上。在沒有經過筆劃切割程序情況下，一筆劃中有許多轉折時，這些筆劃全部都會限制操作者的操作範圍。當操作者往正確的操作路徑移動時，如果移動途中先接觸到其它操作路徑，那麼操作者的操作範圍將會限制在首先接觸到的路徑上，而非原本操作者的所預計的操作路徑，如圖 2-12(a)說明未經過筆劃切割程序情況下，圖中所示的三條路徑對於限制操作者的操作範圍都為最優先，意指操作者先接觸三條路徑中的任一條，該路徑都會把操作者的操作範圍限制在該路徑上，圖中標示為目標筆劃的路徑，為操作者正確的操作路徑，有可能因為操作者在往目標筆劃的移動途中，先接觸到其它任二條路徑而操作範圍被限制在非原本預計的操作路徑上。在經過筆劃切割程序後，能避免上述情況發生，分割後的筆劃我們給予先後次序之分，筆劃

引導和路徑限制策略就會照著分割後規劃好的筆劃順序，依序限制操作者在各筆劃上的操作範圍，圖 2-12(b)說明經過筆劃切割程序情況下，圖中所示的三條路徑對於限制操作者的操作範圍有先後之分，意指三條路徑會按照先後次序限制操作者的操作範圍，其引導程序為操作者必須先操作在順序為 1 的路徑上之後，順序為 2 的路徑才會限制操作者的操作範圍，在操作次序還未到的路徑則不會限制操作者的操作範圍，在圖中以虛線表示。

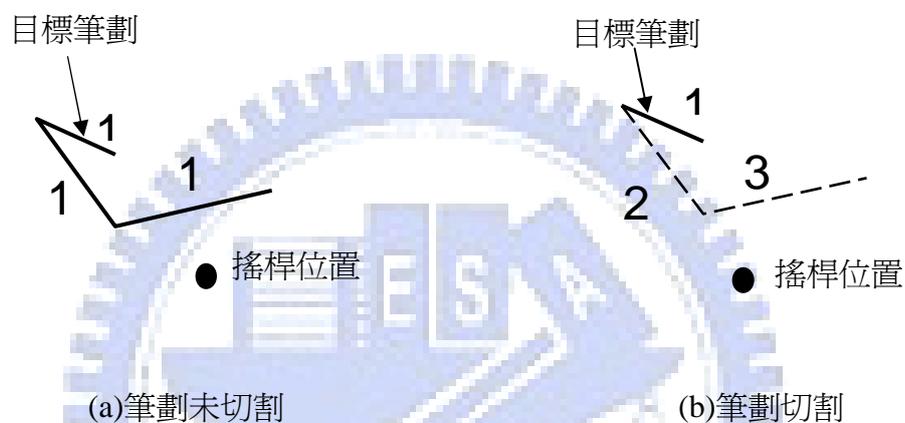


圖 2-12 筆劃切割與否對於筆順引導策略造成的影響

2.2.2 具阻尼路徑限制模式

具阻尼路徑限制模式是路徑限制模式的延伸，模式中採用了路徑限制和筆劃引導策略目的在於使操作者能夠隨著筆順引導，依序在預設的路徑上進行操作，藉此練習字的結構，各筆劃的間距以及深淺程度，並且在筆劃轉彎處加入阻尼策略，藉由此策略提供一個與操作者方向相反的緩衝力，使操作者在轉彎處能夠獲得穩定的操控感，呈現較好的結果，如圖 2-13 所示，在本模式中我們不預設操作者在路徑上的操作方向和速度，所提供的力輔助屬於被動形式。

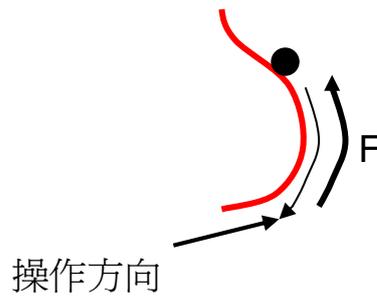


圖 2-13 阻尼策略對操作者提供一緩衝力

在介紹過具阻尼路徑限制模式的輔助程序後，除了之前介紹過的路徑限制和筆劃引導策略，接著介紹本模式中在轉彎路徑所採用的阻尼策略以及經過筆劃切割程序與否，對於所採用的阻尼策略在設計上有哪些幫助和困難。阻尼的作用與黏滯阻力相同，皆為抵消速度，阻尼所發出的力，與兩連接物體之間的相對速度和阻尼常數 k_d 成正比，相對速度和阻尼力(damping force)的關係式如下

$$F_d = k_d (v_0 - v_1) \quad (2.2)$$

其中 F_d 是阻尼常數和兩連接物體上連接點相對速度的函數，阻尼力的公制單位是 N，而速度的單位是 m/s，而 k_d 的單位是 kg/s。接著討論有無筆劃切割對於阻尼策略設計上影響，在沒有經過筆劃切割程序的處理下，無法把一筆劃內部的轉折路徑切割出來，導致在轉折處所設計的阻尼策略無法只針對轉折路徑使用，使得整個筆劃上都有阻尼效果，如圖 2-14(a)實線所示，在經過筆劃切割程序處理後，我們設計的阻尼策略能針對轉折的路徑使用，如圖 2-14(b)實線所示。



圖 2-14 筆劃切割與否對於阻尼策略造成的影響

2.2.3 路徑引導模式

路徑引導模式是路徑限制模式的延伸，目的在於預先規劃了每個筆順的操作速度和方向，使得操作者在操作每個筆順時，能夠照著預先規劃的方向和速度操作，藉此引導操作者書寫時的手部動作，此模式除了使用路徑限制和筆劃引導策略使操作者能夠隨著筆順引導，依序在預設的路徑上進行操作，藉此練習字的結構，各筆劃的間距以及深淺程度外，另外針對每個筆順使用了彈簧策略，使操作者能夠依照預先規劃的方向操作，如圖 2-15 所示，本模式中我們預設操作者在路徑上的操作方向和速度，所提供的力輔助屬於主動形式。

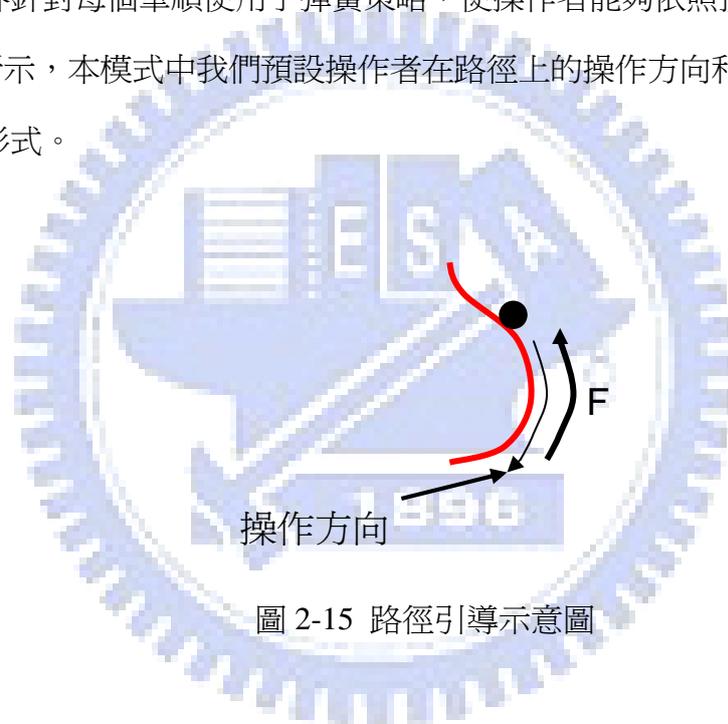


圖 2-15 路徑引導示意圖

在介紹過路徑引導模式的輔助程序後，除了之前介紹過的路徑限制和筆劃引導策略之外，接著介紹本模式中對於每個路徑所採用的彈簧策略以及經過筆劃切割程序與否，對於所採用的彈簧策略在設計上有哪些幫助和困難，彈簧是最常被用來作為描述物體的軟硬度的結構，用來連結二個物質，在本實驗中，此二物質分別為搖桿位置和預先所規劃要引導操作者所到達的目的地位置。而彈簧的長度為搖桿位置和目的地位置間的距離，彈力的計算根據虎克定律，公式為

$$F_s = k_s (p_0 - p_1) \quad (2.3)$$

其中 F_s 是彈力， k_s 是彈力常數，用來指出彈簧收縮的劇烈程度， k_s 值越大，彈簧越硬， p_0 是目的地位置，而 p_1 是搖桿位置。最後討論有無筆劃切割對於彈簧策略在設計上的影響，在筆劃未切割情況下，直接把引導路徑的起始位置以及目的地位置設定成筆劃的起點和終點，有可能出現引導方向錯誤的情況，如圖 2-16(a)所示，經過筆劃切割，就能避免此情況發生，如圖 2-16(b)所示，圖中說明經過筆劃切割後成為有先後次序之分的三個操作路徑，我們即以三個路徑的起點和終點個別規劃引導方向，而每個路徑的終點都作為接續下一路徑的起點，圖中以黑點表示。



圖2-16 筆劃切割與否對於彈簧策略造成的影響

在介紹完三個輔助模式的輔助程序和各模式所採用的輔助策略，以及筆劃切割對於各輔助策略在設計上有那些優勢後，接著我們呈現整個力輔助的流程，此流程說明從操作者經由人機介面選擇輔助模式，到系統判斷操作者是否在力輔助範圍內來決定是否進行輔助的流程，如圖 2-17 所示，當使用者對系統輸入起始命令後，系統判斷使用者所選擇的輔助模式，開始對使用者進行力輔助，系統先根據操作點的位置以及輔助策略的不同設置不同形式的力輔助，接著判斷使用者是否離開力輔助範圍，若使用者已離開力輔助範圍，表示已完成動作或者使用者偏離操作路徑，系統將取消力輔助；反之則依搖桿位置在力輔助範圍中的位置計算出回饋力，並透過搖桿回饋給使用者。

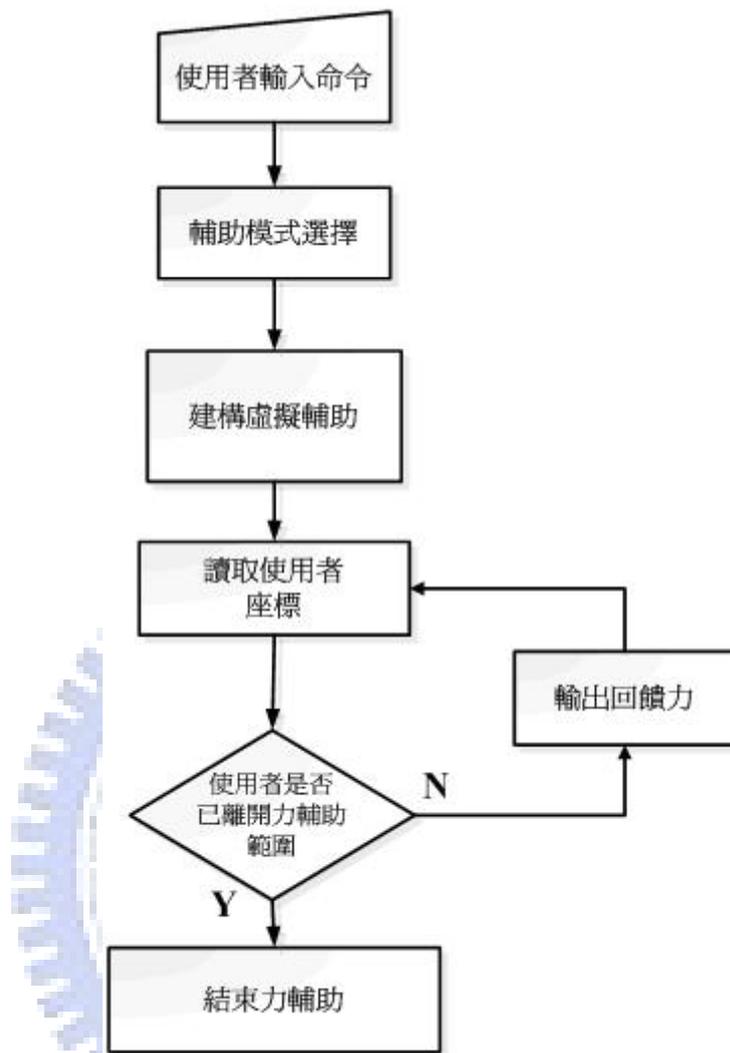


圖 2-17 毛筆書寫輔助系統對使用者進行力輔助流程圖

第三章

系統實現

毛筆字書寫輔助系統由 1 台電腦主機，1 支力回饋搖桿，以及 1 台平面顯示器組成，如圖 3-1 所示，電腦主機負責提供操作者視覺回饋、建立虛擬實境、建構虛擬導引，以及與力回饋搖桿進行溝通，其透過介面卡讀取搖桿資訊，並轉換為位置指令，同時也經由螢幕提供使用者視覺回饋，工作執行中將使用者對搖桿輸入的命令傳送到毛筆字書寫輔助系統，系統在回送力資訊及視覺資訊時，將該力資訊與虛擬導引所計算之回饋力合成，成為操作者實際所感受到的回饋力。

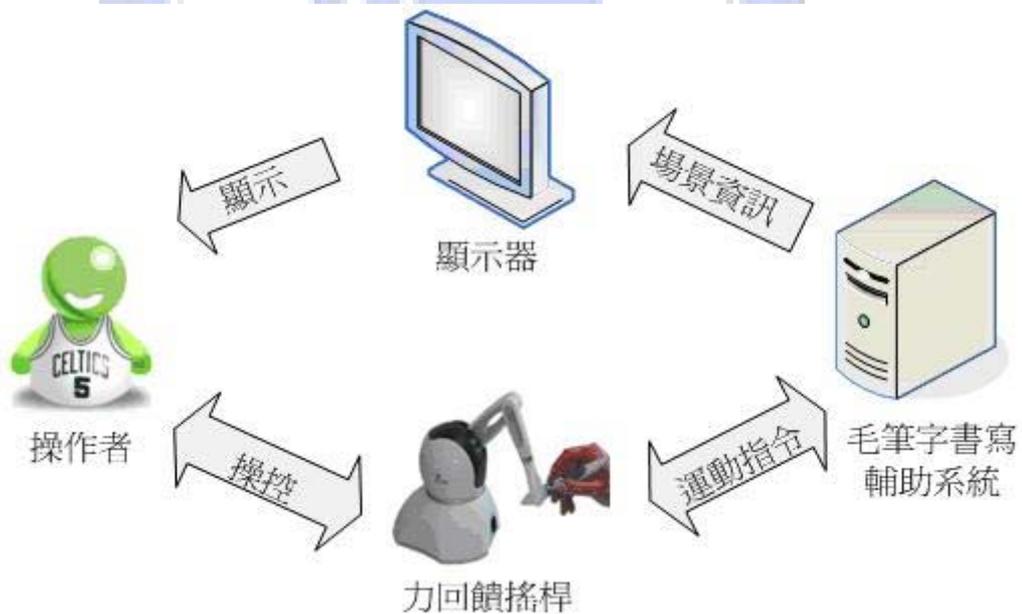


圖 3-1 毛筆字書寫輔助系統架構圖

3.1 軟體實現

本論文運用到的軟體工具主要為 Visual C++ 6.0 和 OpenGL，實現的部分為搖桿的驅動以及建構虛擬環境。由於力回饋裝置的資訊和虛擬實境的更新必須即時處理，因此我們採用多執行緒的方式，我們先介紹虛擬場景的成像過程，接著介紹虛擬實境中 3D 效果的呈現及 3D 物體的觸覺演算法，之後介紹多執行緒如何運用於本系統。

3.1.1 虛擬場景

由於本系統所用搖桿之開發應用程序介面(API)能與OpenGL API相容，能直接透過 Open GL程式設計語法建構虛擬環境，因此我們採用OpenGL繪圖函式庫來建構虛擬場景。如圖3-2所示，為OpenGL成像流程，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，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，將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、顯示卡等，到了這個階段，圖形顯示裝置會把影像顯示到螢幕上。

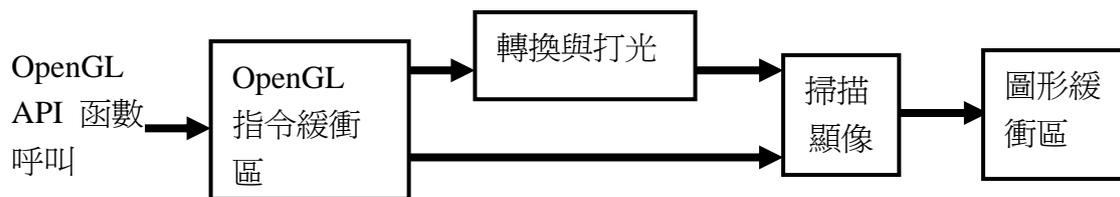


圖 3-2 OpenGL 成像流程

3.1.2 3D 效果呈現

在三維的空間中，我們使用笛卡兒座標(Cartesian Coordinates)系統來表示物體的位置，如果要將三維的笛卡兒座標轉換成可繪製在螢幕上的二維座標，使用的方法是投射(Projections)，投射的概念是把 3D 座標投射到 2D 平面上，就像在一片玻璃後面用奇異筆把某物件的輪廓描出來，當物件移開或玻璃移動時，依然可以看到物體在該角度下的輪廓，如圖 3-3 所示，將背景的一棟房子描繪到玻璃上。投射的方法主要有兩種，第一種是透視投射(Perspective Projections)法[8]，這種投射法會使遠方的物體看起來比近端物體更小，如圖 3-4 所示，靠近觀察區域前方的物體看起來大小與實際大小較相近，但靠近末端的物體投射後縮小許多，這種投射方式能為模擬 3D 的畫面帶來較佳的真實感。

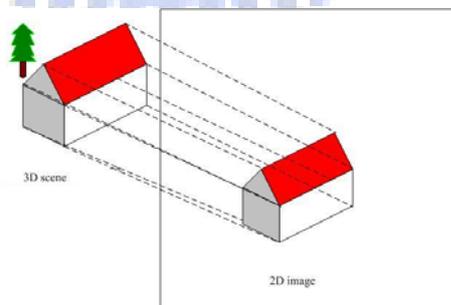


圖 3-3 3D 影像投射到 2D 平面示意圖

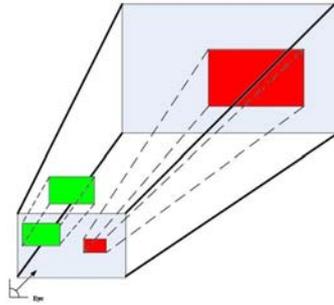


圖 3-4 透視投射示意圖

第二種是正交投射(Orthographic Projections)[8]，此投射法也是我們實驗所使用的方法，因為操作者只需要得知搖桿在虛擬場景中的位置是否與書寫平面上的位置相對應，加上我們建構在虛擬場景中的字型大小必須與操作者書寫的字型大小相近，如果我們使用透視投射法，那麼建構在虛擬場景中的字型大小與操作者書寫出來的字不成比例，在此投射法下所提供的視覺回饋對於操作者來說，不容易抓到搖桿在虛擬場景中和書寫在紙面上的對應位置，反而造成操作者的困惑，雖然正交投射法在視覺上無法提供深度的資訊，但是搖桿卻能夠提供深度的力覺回饋，比起提供深度視覺回饋，深度的力覺回饋更能讓操作者有深刻的體會。此法的概念為指定一個立方或長方體為範圍來進行這種投射，立方或長方體範圍外的任何事物都不會畫出來，所有具有相同維度的物體不管遠近看起來大小都相同，如圖 3-5 所示，這種投射常用在建築設計、CAD(電腦輔助設計)或 2D 圖形上。

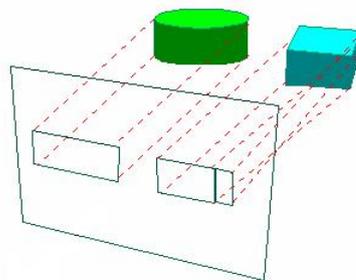


圖 3-5 正交投射示意圖

而 3D 轉 2D 的座標點轉換程序，如圖 3-6 所示，其中 Modelview 矩陣是用來轉換原始座標到眼睛座標，接下來眼睛座標會被乘上投射矩陣，產生裁減座標(clip coordinates)，這會去除不需出現在觀景窗中的資料，如物體的背面，再來經過透視投射的處理並正規化，然後將所得結果給 viewport 以映射到 2D 平面上。

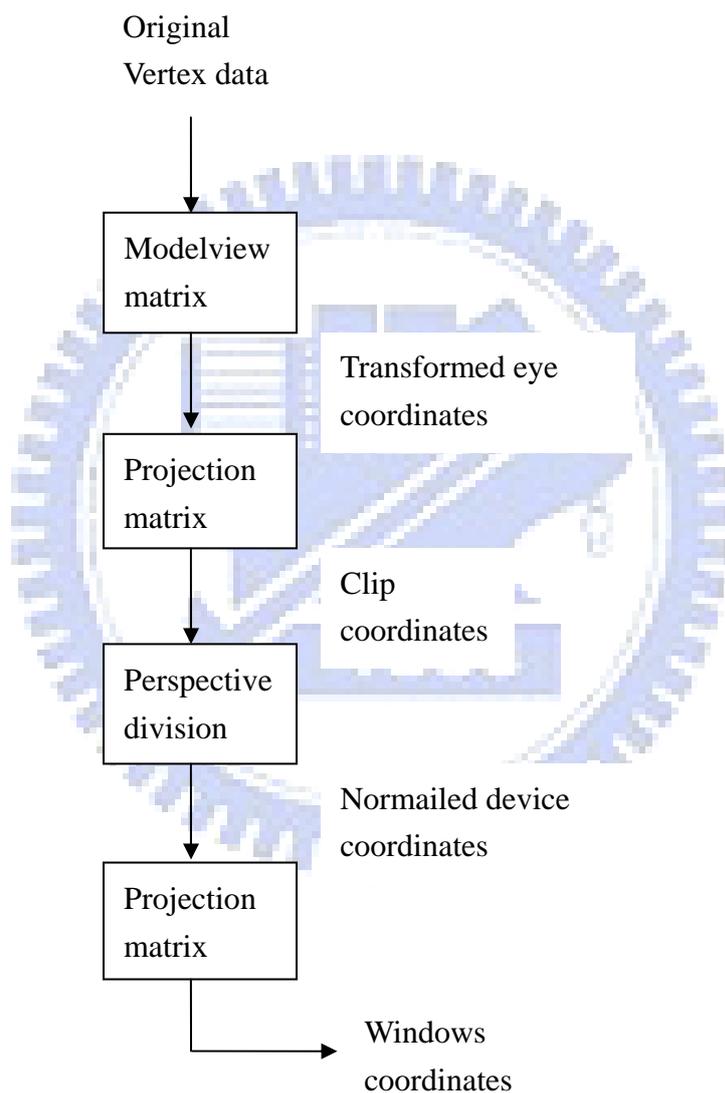


圖 3-6 3D 轉 2D 座標點轉換程序

3.1.3 觸覺演算法

本實驗所用搖桿之開發應用程序介面(API)採用了一個觸覺演算法，God-Object Method [9]，程式設計者能夠以編寫 Open GL 程式碼的方式建構 3D 物體，而 God-Object Method 觸覺演算法使得操作者能夠感受到建構在虛擬環境中的 3D 物體，此法增加一個搖桿虛擬位置變數的方式來獲得所需要的資訊，得以改善了前人提出的 volume methods 觸覺演算法，以下介紹觸覺設備在力輸出上的限制，和以 volume methods 作為觸覺演算法並且搭配觸覺設備進行虛擬物體的擬真時，會衍生出的缺點，以及 God-Object Method 如何改善其缺點。

觸覺設備能夠順應任何虛擬物體，透過力輸出進行軟硬程度上的描述，由於力輸出的最大程度受限於機構設計，因此對於物體堅硬程度方面的擬真不夠真實，此情況使得操作者在碰觸一個預計被設計成高硬度的虛擬物體時，由於觸覺設備的力輸出受限於機構，使得觸覺設備無法提供相當的力輸出回饋給操作者，讓操作者感受不到虛擬物體該有的硬度，因此造成操作者在操控觸覺設備上和觸覺設備在虛擬場景中穿透虛擬物體的程度不成比例。之前的 volume methods 觸覺演算法都是以一對一的映射關係，直接由空間中計算出穿透虛擬物體的深度進而決定力輸出的大小，加上觸覺設備的力輸出最大程度受限於機構，在此觸覺演算法下衍生出四個問題：(1)無法清楚界定搖桿是從物體的哪個表面侵入物體，如圖 3-7 所示；(2)當搖桿橫越子物體的交界時，會產生力的不連續：Massie 和 Salisbury[10]試圖把一個物體的內部以訂定的方式把物體內部和其所屬的表面切成許多子物體，當搖桿侵入子物體其中之一時，根據侵入的深度和彈性係數產生一反作用力在所接處的表面。雖然此法解決了第一個缺點，但卻衍生另一個缺點，就是當搖桿在物體內移動時，橫越不同的子物體在遇到轉角處會短暫的感覺到力的不連續產生，如圖 3-8 所示，(3)搖桿侵入體積過薄的物體會誤判力方向：搖桿侵入物體內部之前

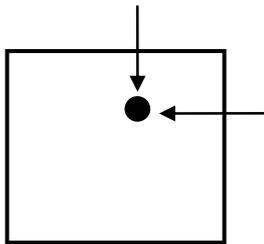


圖 3-7 無法清楚界定搖桿進入位置

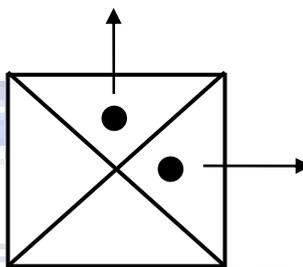


圖 3-8 轉角處會產生不連續的力[10]

必須要產生足夠的力，使操作者感受到物體的硬度。在侵入的物體為細薄的情況下，穿透的深度大過於物體，在機構構造和物理特性上的限制，場向量模型(vector field model)會產生不接近真實的力。當搖桿操作在由兩個子物體構成的物體內部時會產生一個垂直於表面的反作用力，如圖 3-9(a)所示，在搖桿操作在其中一子物體內部且往子物體內部更深處操作，如圖 3-9(b)所示，直到深度跨越到另一子物體時，搖桿會被判斷為是從第二個子物體表面進去，產生一個方向錯誤的反作用力，如圖 3-9(c)所示。一個可行的解決方法為記錄搖桿進入物體表面的位置，再用記錄的位置產生適當的反作用力。不過，此法在記憶體和執行時間上數值的表示方法不容易處理；(4)搖桿在多個物體的交界處會誤判反作用力大小：用簡單的幾何體構成結構較為複雜的物體時，在子物體間的邊界交界處，產生的力不一定正確。交界處為正交時，力的大小為二個直角分量和，這個力的量為正確，如圖 3-10(a)所示，交界處呈頓角時，會產生一個略大於實際情況的力，如圖 3-10(b)所示，交界處呈近似平行時，會產生大於實際情況二倍的力，如圖 3-10(c)所示。

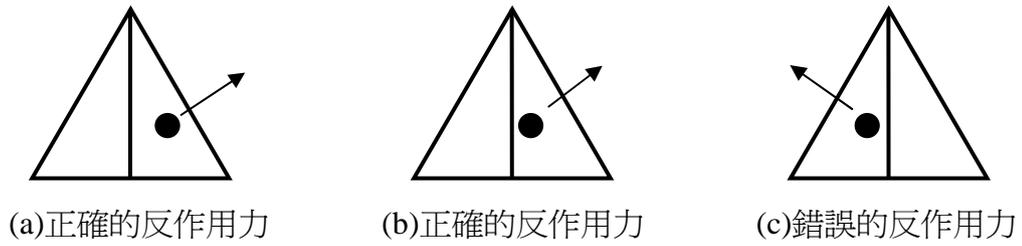


圖 3-9 搖桿穿透細薄物體示意圖

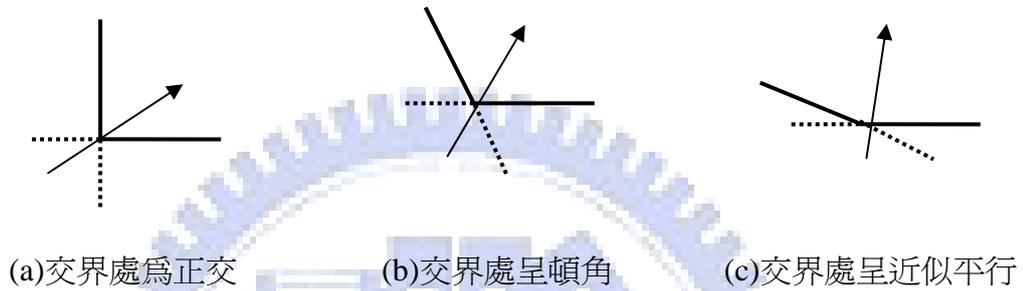


圖 3-10 交界處可能產生錯誤的反作用力

在介紹過 volume methods 的情形後，接著介紹 God-Object Method 如何改善其缺點，此觸覺演算法是利用一個跟隨搖桿的點取得所需要的資訊，此點稱為 god-object，在搖桿進入物體內部時，god-object 會被限制在物體表面，不會隨著搖桿進入物體內部，此時所感受到物體的軟硬度是藉由搖桿和 god-object 中間的彈簧，以及阻尼計算而得，如圖 3-11 所示，在搖桿沒有與物體接觸時，god-object 會持續更新並且跟隨著搖桿移動。

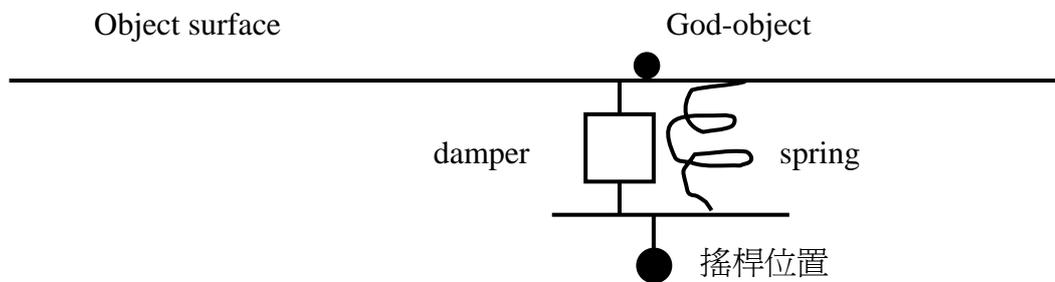


圖 3-11 利用 God-object 和搖桿間的阻尼彈簧模擬物體的軟硬程度

當搖桿所接觸到的面為凸面時，同時間只有一個面會被視為接觸，God-object 在二個面所構成的凸面上的判斷分為二個階段，首先操作在物體表面的 god-object 會停留在第一個接觸到的面，如圖 3-12(a)所示，隨著搖桿移動到另一個面內部，god-object 仍然被限制在前一個平面上，如圖 3-12(b)所示，之後 god-object 會落在搖桿所在物體內部的相對應平面上，如圖 3-12(c)所示，由於處理的時間和掉落的距離都很微小，所以使用者是無法察覺到的，當所接觸到的面為凹面時，多個面能同時被視為接觸。凹面為二個平面交界的情況時，二個面都作為 god-object 不能進入的限制面，而二個平面交界構成的線為 god-object 所能移動的範圍，凹面為三個平面交界的情況時，三個面構成一個限制 god-object 的點，在凹面超過三個時，同時最多只有三個面視為有效接觸面。凹面呈現銳角時，搖桿侵入其中一面內部時，如圖 3-13(a)所示，搖桿在物體內部滑動，當搖桿滑動至第二個表面的延伸面，虛線部份時，如圖 3-13(b)所示，god-object 的位置卻侵入了物體內部，此情況不被允許。因此用了一個反覆的檢查程序來修正這個錯誤。作法為首先找出後面被接觸到的面，把它視為新的限制面，再利用新的 god-object 的位置作為搖桿在物體內部的新的位置，如圖 3-13(c)所示，並再重覆這個程序直到沒有錯誤產生。

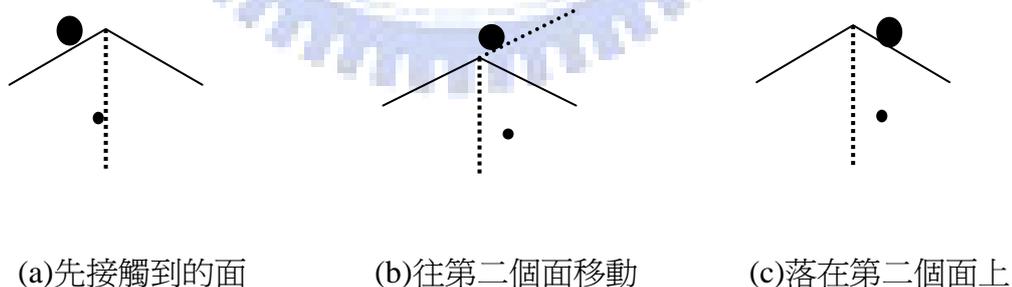


圖 3-12 搖桿接觸物體凸面的程序

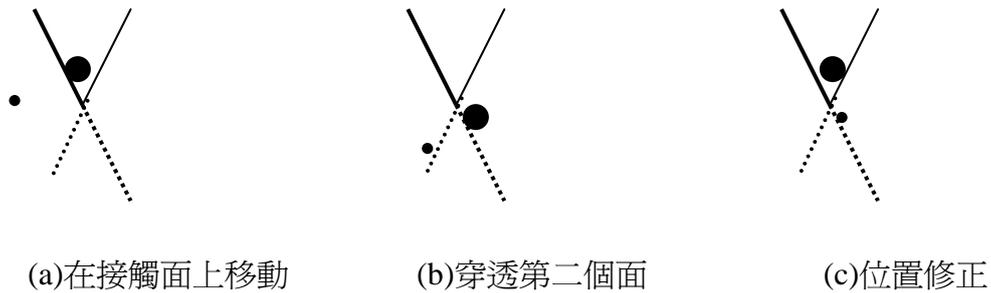


圖 3-13 搖桿接觸物體凹面呈現銳角時的處理程序

3.1.4 多執行緒

由於人類先天對於視覺與力感兩者感知能力的不同，因此虛擬場景對操作者提供的視覺回饋頻率與力感的更新率也不同，為了使這二個更新頻率不同的程序能夠實現於本系統上，我們採用多執行緒的方法，以下介紹人對於視覺和力覺感知程度的不同，以及多執行緒的概念和如何運用於本系統。

一個應用程式的繪圖效能以每秒所能繪的畫面張數(frames per second,fps)衡量，這就是螢幕一秒之內所能更新的畫面數，一般來說，能讓使用者感到互動順暢大多在 15 fps 以上，這個值大約是 3D 電腦動畫最低的標準，但仍然稍嫌不足，電影的流暢程度約為 24fps，電腦動畫要達到流暢程度的門檻為 30fps，就某些情況而言還是不夠，如果畫面顯示的是相當大的區域，那麼要達到流暢程度為 60fps。

在觸覺回饋 (haptic feedback) 裝置方面，要依其應用設定合適的力更新率，例如醫生開刀所需的觸感回饋 (touch feedback) 裝置需要高達 1kHz 才能提供足夠的觸感 [11]；模擬一堵剛性為 10000 N/m，黏性為 1000 N/(m/sec) 的硬牆所需力更新頻率要達 500 Hz 以上，但若操作者的手快速移動則需要更高的更新頻率 [12]；在這兩個例子中，

所要求的力更新頻率遠較影像的更新頻率 30 – 60 Hz 高出很多，顯見在某些情況下，觸覺和影像使用相同的更新頻率並不符合人的輸入要求。如果將力與視覺的更新頻率調為相同，即將視覺更新頻率拉高到與觸覺標準，則會因圖形處理耗去電腦硬體的大量計算，使得電腦執行效能大幅下降；所以不把電腦的圖形處理和觸覺處理在同一個執行緒裡頭運行，而以多執行緒 (multithread) 的方式進行，將兩者分別獨立運作。

多執行緒能夠同時提供多個執行緒 (thread) 的功能，通常執行緒所指的是在一個程式中的副程式，多重執行緒將一個程式分割成幾個小模組，每個模組都在一個特定的執行緒當中執行，這就宛如平行處理一般，但是卻利用了多工的能力，表面上使用者並未感覺到多重管線的運作方式，但實際上程式執行的效率卻提高了，如圖3-15所示，其中圖3-14(a)為一般程式設計時的執行順序和時間的關係，圖3-14(b)則為執行緒程式執行的順序和時間的關係。

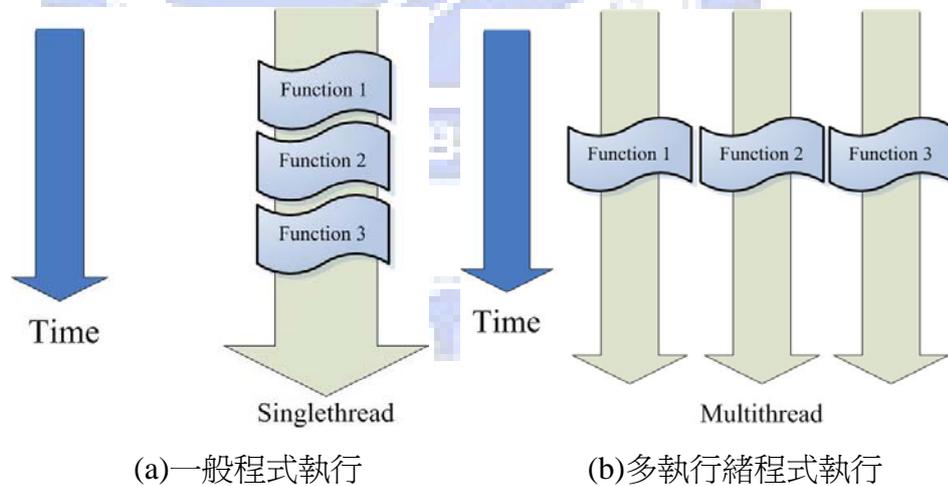


圖3-14 執行順序和時間關係

由於搖桿力覺的更新以及虛擬實境畫面的更新必須同時進行，因此我們將開啓二個多執行序來完成這二項工作，執行序 A 負責搖桿位置以及力覺更新，執行序 B 則存取搖桿位置後將其輸入虛擬場景並更新畫面，並且將虛擬導引計算出的虛擬的回饋力回傳給執行序 A，執行序 A 會將從執行序 B 收到的力資訊，回饋給使用者，圖 3-15 所示為執行

緒 A、B 之間互相通訊的示意圖。

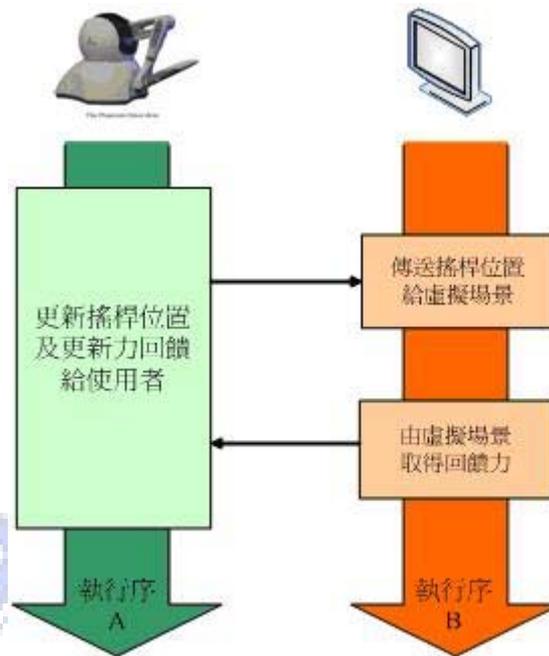


圖 3-15 多執行序執行示意圖

最後我們介紹毛筆字書寫輔助系統的使用介面，如圖 3-16 所示，虛擬場景中提供四個資訊，分別為 FPS(frame per second)，提示目前虛擬場景更新率是否達到流暢值(最低要求為 30 fps)；Update Rate，提示系統中觸覺更新率是否達到流暢值(1000Hz)；Magnitude，提示在路徑引導模式下彈簧策略透過力搖桿對操作者施力的量，單位為牛頓(N)；以及 POS，提示搖桿在虛擬場景中的位置。另外當操作者在路徑引導模式下操作時，在搖桿和彈簧策略預先規劃所要引導到的目的地間會有一條連接線作為提示；而輔助模式的選單則在視窗中以滑鼠右鍵來開啓，選單依序為 Quit，結束，Free Draw，不受任何力輔助操作，Path Constraint，路徑限制，Path Constraint with damping，具阻尼路徑限制，以及 Path Guidance，路徑引導。

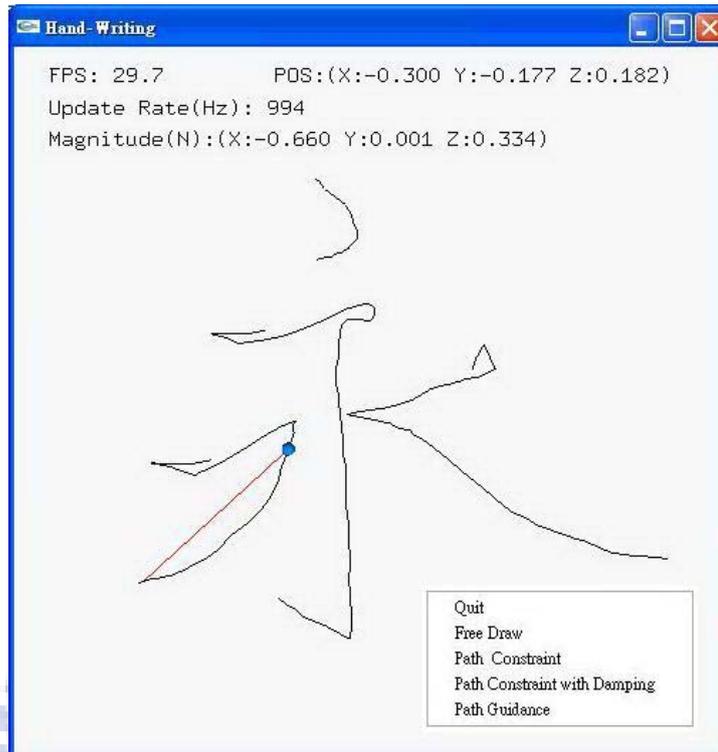


圖 3-16 毛筆字書寫輔助系統使用介面

3.2 硬體實現

在實現毛筆字書寫輔助系統時，所使用到的設備包含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及計算虛擬力回饋的方向和大小，並回饋給使用者；力回饋搖桿部份，我們採用美國Sensable公司出產的 Phantom Omni三軸力回饋搖桿，如圖3-17所示，由於搖桿的操作握柄外觀為筆的設計，因此很合適用來作為本系統的輸入設備，上面有二個按鈕，可以設計作為特定控制功能；此搖桿的工作區域涵蓋了大部份書寫情況的要求；解析度為0.055mm高於許多市售搖桿，對於毛筆字書寫來說，手部細微的操控動作是很講究的，因此搖桿所提供的高解析度很適合用來發展本系統；相較於搖桿的最大輸出力3.3N，本系統的輔助模式所提供給操作者的力感回饋最大為2.46N，因此搖桿能夠提供足夠的力輸出，表3-1為此搖桿規格。



圖 3-17 Phantom Omni 三軸力回饋搖桿

表 3-1 Phantom Omni 三軸力回饋搖桿規格表

Forcefeedback workspace	160 W x 120 H x 70 D mm.
Weight (device only)	15 oz.
Range of motion	Hand movement pivoting at wrist
Nominal position resolution	0.055 mm.
Maximum exertable force at position	3.3 N
Continuous exertable force	0.88 N
Stiffness	X axis 1.26 N / mm. Y axis 2.31 N / mm. Z axis 1.02 N / mm.
Inertia (apparent mass at tip)	45 g
Force feedback	x, y, z
Interface	IEEE-1394 FireWire port: 6-pin to 6-pin
Supported platforms	Intel or AMD-based PCs

第四章

實驗

本實驗目的為檢測操作者透過毛筆字輔助系統輔助與否，對於操作者在毛筆字書寫結果上是否有所影響，受測者為三名年齡 25 歲不擅長毛筆字書寫的男性學生，在實驗開始前都已經有多次使用本系統的經驗。本實驗中虛擬場景所顯示之示範者軌跡與示範者書寫結果為等比例，因此操作者經由毛筆字輔助系統所寫出的字體大小與示範者相近，示範字體為永字，因為永字八法涵蓋了大部份書法字中的筆法，示範字體大小為 12.5x12.5cm²，毛筆字書寫區域大小為 14x14cm²，力回饋搖桿輸出範圍為-3.3N~3.3N，路徑限制策略使力回饋搖桿輸出力大小 \vec{F} 公式如下：

$$\vec{F} = \begin{cases} k(pos - jpos) & \text{if } jpos < pos + d \\ 0 & \text{if } jpos > pos + d \end{cases} \quad (4.1)$$

常數 k 經過實驗後， k 值設為 0.8 為一個較理想的值， $d=1.5(\text{cm})$ 為該輔助策略所能對操作者進行力輔助的最大距離， pos 為操作路徑上距離搖桿位置最近的點， $jpos$ 為搖桿的位置，此策略使得力回饋搖桿對操作者輸出力範圍在 0N~1.2N。阻尼策略使力回饋搖桿輸出力大小 \vec{F}_d 公式如下：

$$\vec{F}_d = k_d (v_{pos} - v_{jpos}) \quad (4.2)$$

阻尼常數經過實驗後， k_d 值設為 1 為一個較理想的值，使得力回饋搖桿輸出力 \vec{F}_d 對操作者提供的緩衝力範圍在 0N~2.46N 之間， v_{pos} 、 v_{jpos} 分別為路徑和搖桿連接點的相對速度。

彈簧策略使力回饋搖桿輸出力大小 \vec{F}_s 公式如下：

$$\vec{F}_s = k_s (p_0 - p_1) \quad (4.3)$$

彈力常數 k_s 經過實驗後， k_s 值設為 0.008~0.009 為一個較理想的值，使得力回饋搖桿輸出彈力 \vec{F}_s 範圍在 0N~0.59N 之間， p_0 是目的地位置，而 p_1 是搖桿位置。

4.1 實驗流程

整個實驗流程如圖 4-1 所示，步驟如下，(1)首先請受測者確認本次實驗所要練習的字，(2)受測者在未輔助情況下，以本身過去書寫毛筆字的筆法書寫三次，(3)在路徑限制輔助模式下書寫三次，(4)在具阻尼路徑限制輔助模式下書寫三次，(5)在路徑引導輔助模式下書寫三次，(6)再次在未輔助情況下，書寫三次，此目的在觀察受測者透過毛筆字書寫輔助系統後，是否與未輔助前對於該字的架構、筆法都有改進，全部流程結束，受測者一共書寫十五個相同的書法字。



圖 4-1 實驗流程圖

4.2 操作實景

本節將介紹毛筆字書寫輔助系統實際運作的情形，圖 4-2 為操作者進行操作的實景，操作者透過螢幕回傳的畫面得自己控制的搖桿在虛擬場景中以及本身書寫的情形，操作者在路徑引導模式時，虛擬場景中對操作者的視覺回饋，比起另外二個輔助模式，會在搖桿與引導目的地間以紅色的連接線作為提示，告知操作者當下所操作筆劃的引導方向。



圖 4-2 操作者進行毛筆字書寫輔助系統的情形

4.3 結果與討論

每位受測者在二種未輔助以及三種具輔助情況下各進行三次的操作，共計有十五組實驗數據，礙於篇幅我們在此只呈現在每種情況下，三位受測者三組數據中其中一組，來與其它情況作比較。我們把實驗數據分成兩部份來討論，第一部份我們討論每位受測者與示範者軌跡的比較，並且統計在五個實驗情況下，受測者與示範者的軌跡誤差值，是否因為透過輔助模式或者經過輔助模式學習後，在與未使用輔助系統前比較，有下降的趨勢；第二部份我們就以書寫結果進行觀察，是否因為透過輔助模式或者經過輔助模

式學習後，在與未使用輔助系統前比較有所進步。在受測者與示範者的軌跡誤差計算方式方面，我們將受測者書寫永字八法中的勒(橫)、努(直)筆法軌跡與示範者對應的筆法軌跡對齊，再分別計算每個筆劃，受測者在起筆位置與收筆位置，與示範者的誤差，最後分別加總起筆誤差與收筆誤差。這樣的計算方式用意在於，在受測者和示範者操作軌跡中心點對齊的情況下，觀察受測者是否因為力輔助或者使用輔助系統後，對於各個筆劃的起收筆情形，比起未使用輔助系統前進步，起收筆誤差值 d_{start} 以及 d_{end} 計算公式如下

$$d_{start} = \sum_{s=1}^n |mpos_s - spos_s| \quad (4.4)$$

$$d_{end} = \sum_{s=1}^n |mpos_s - spos_s| \quad (4.5)$$

其中 s 為筆劃， n 為字體筆劃數，本實驗有 17 劃，因此 $n=17$ ，在 4.4 式中 $mpos_s$ 和 $spos_s$ 分別為示範者和受測者在各筆劃的起筆位置，在 4.5 式中， $mpos_s$ 和 $spos_s$ 分別為示範者和受測者在各筆劃的收筆位置。

在說明如何計算起收筆誤差值的方式後，接著個別討論三位受測者的實驗結果。第一位受測者在使用輔助系統前後，與示範者的操作軌跡進行比較及其書寫結果，如圖 4-3 所示，起收筆誤差值統計表，如表 4-1 所示，將各筆劃起收筆誤差值以折線圖表示，如圖 4-4 所示，我們由受測者一折線圖中的曲線分佈可觀察出，受測者使用系統前後到三個輔助模式所呈現的誤差值大幅下降，此現象指出受測者一在使用過系統後以及透過輔助模式，能夠有效的改善本身的操作軌跡，曲線分佈中受測者操作在三個輔助模式下的誤差值皆很接近，都大大的降低各筆劃起收筆誤差值，指出三個輔助模式皆能夠提供受測者一定呈度的輔助效果，由統計表 4-1 中可觀察出一個比較特別的情況，就是在系統使用後的收筆誤差值與系統使用前相當接近，這是由於受測者在永字八法中的趯(鈞)和掠(長撇)筆法收筆過晚所造成，由書寫結果觀察得知，受測者在使用輔助系統後筆劃的起筆處及轉折處都比在使用前來的稍有進步，這個結果可以預期，因為從受測者在使用系統前的操作軌跡觀察得知，受測者起筆的筆法與示範者差距甚大，導致在筆劃起筆或是轉彎的結果有待加強，而受測者在三個輔助模式下的結果，都比使用輔助系統前後要

來的好，雖然從三個輔助模式輔助後的書寫結果觀察，沒有明顯的優劣之分，但受測者覺得具阻尼路徑限制模式與路徑限制模式比較起來，在筆劃轉彎處加入了阻尼策略，在操控上比較穩定，而路徑引導模式對於操作的流暢度，筆劃深度的落實都能透過此輔助模式得到幫助，說明了輔助模式不止對於軌跡操作有幫助，在書寫結果上也有程度上的助益。

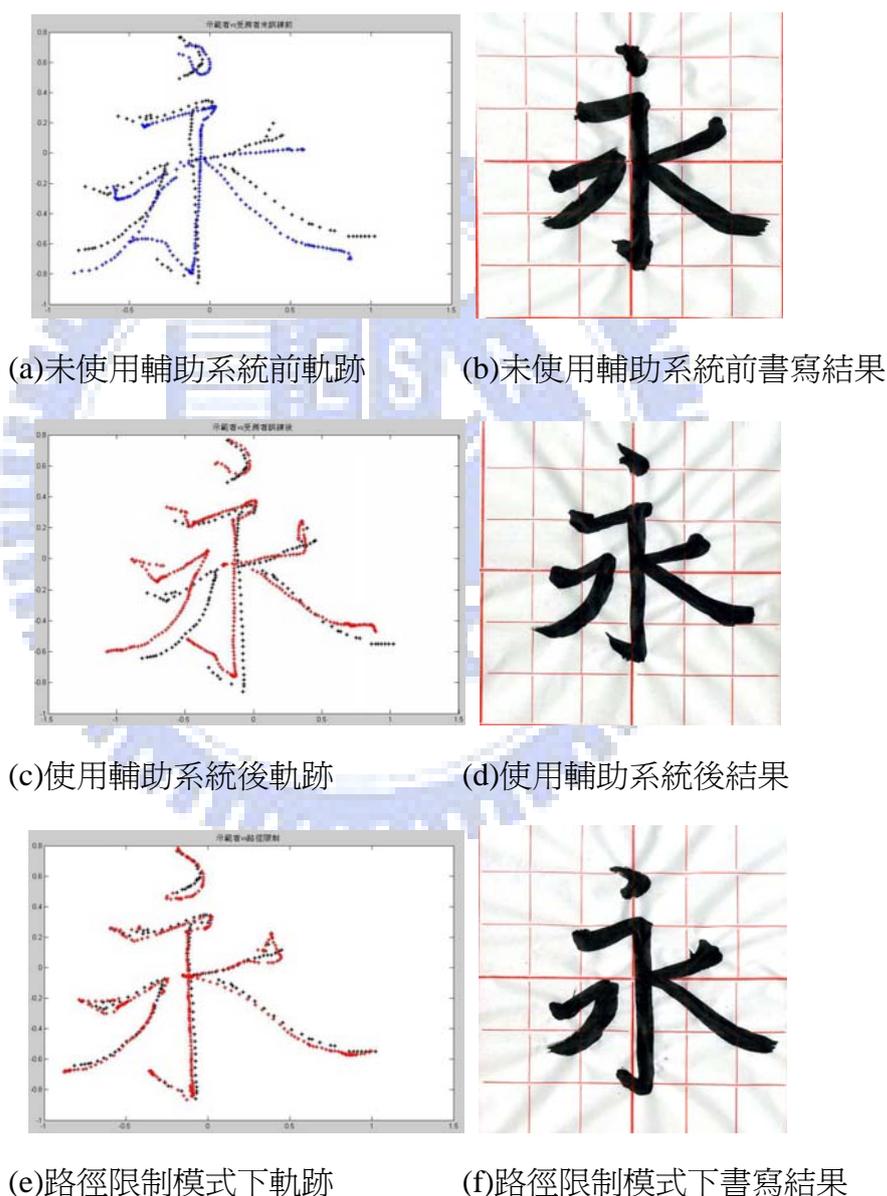
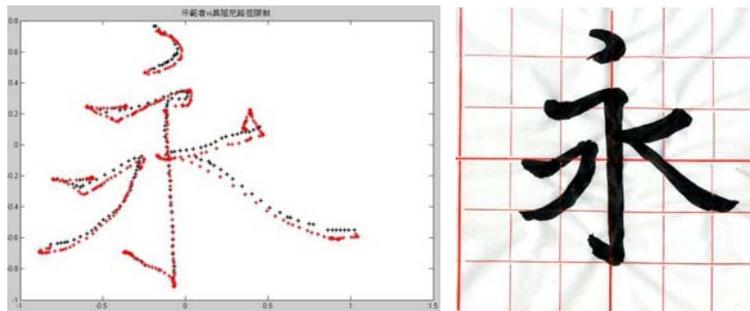
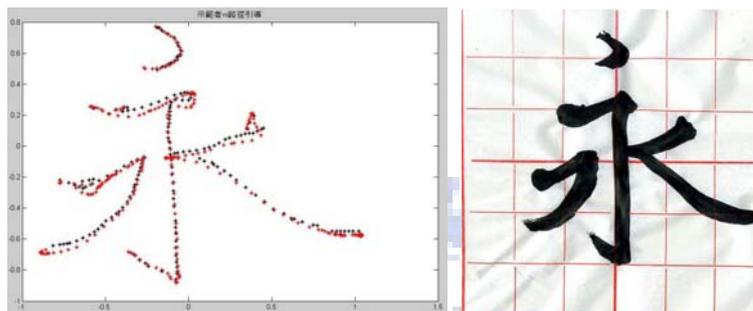


圖 4-3 受測者一在輔助前後與三個輔助模式中的操作軌跡與書寫結果(續)



(g)具阻尼路徑限制模式下軌跡 (h)具阻尼路徑限制模式下書寫結果



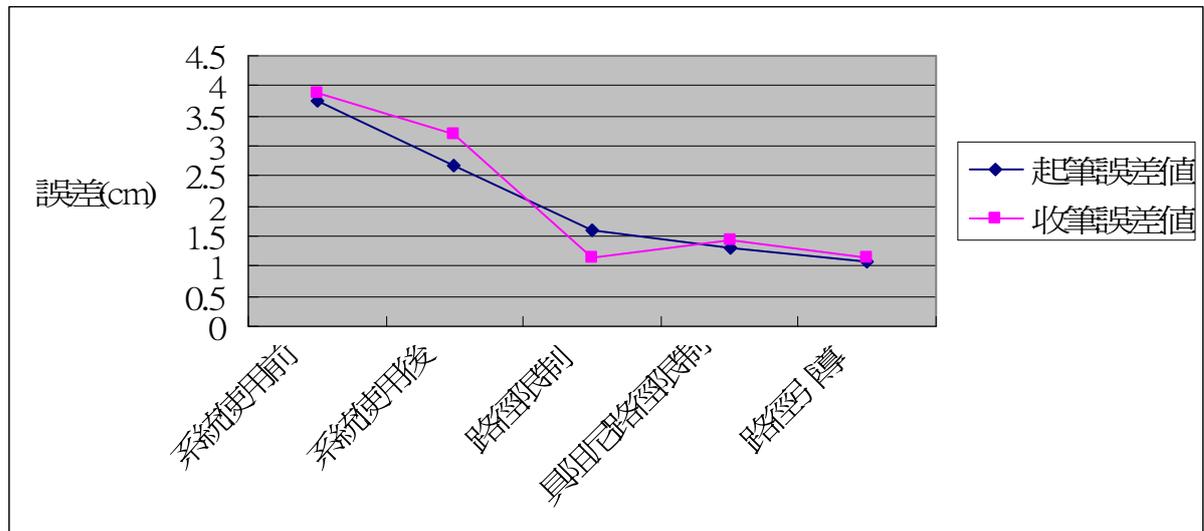
(i)路徑引導模式下軌跡 (j)路徑引導模式下書寫結果

圖 4-3 受測者一在輔助前後與三個輔助模式中的操作軌跡與書寫結果

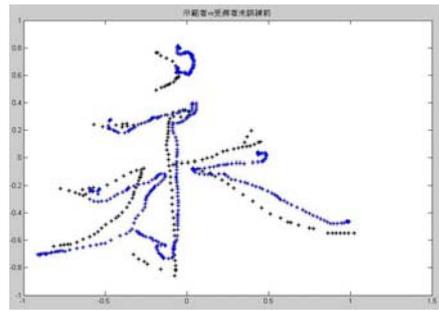
表 4-1 受測者一：起收筆誤差值統計表

受測者一		
	起筆誤差值(cm)	收筆誤差值(cm)
系統使用前	3.7593	3.8722
系統使用後	2.6735	3.1824
路徑限制	1.6001	1.1491
具阻尼路徑限制	1.3032	1.4331
路徑引導	1.0811	1.1519

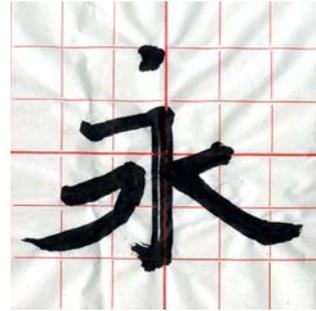
圖 4-4 受測者一：起收筆誤差值折線圖



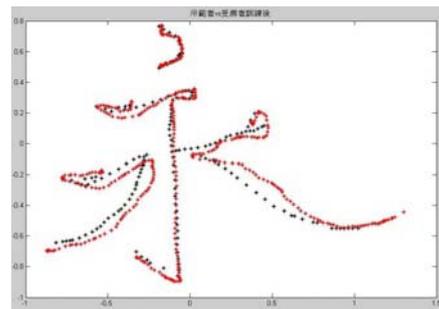
第二受測者在使用輔助系統前後與示範者的操作軌跡進行比較及其書寫結果，如圖 4-5 所示，起收筆誤差值統計表，如表 4-2 所示，將各筆劃起收筆誤差值以折線圖表示，如圖 4-6 所示，第三位受測者的相對結果則分別呈現於圖 4-7、表 4-3 和圖 4-8；我們由兩位受測者的折線圖可觀察出，誤差值都呈現大幅下降，此情況與第一位受測者的結果符合，指出兩位受測者在使用系統後以及在輔助模式下操作，能夠有效改善本身的操作軌跡，曲線分佈中兩位受測者操作在三個輔助模式下的誤差值皆很接近，證明三個輔助模式皆能夠提供受測者一定呈度的輔助效果，由書寫結果觀察得知，兩位受測者在使用輔助系統後筆劃的起筆處及轉折處都比在使用前來的稍有進步，這個結果可以預期，因為從兩位受測者在使用系統前的操作軌跡觀察得知，受測者起收筆的筆法與示範者差距甚大，導致在筆劃起筆或是轉彎的結果有待加強，而兩受測者在三個輔助模式下的結果，都比使用輔助系統前後要來的好，也與受測者一的情況符合。



(a)未使用輔助系統前軌跡



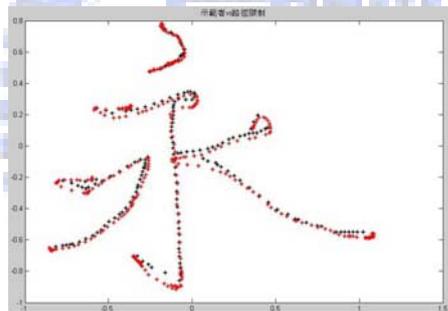
(b)未使用輔助系統前書寫結果



(c)使用輔助系統後軌跡



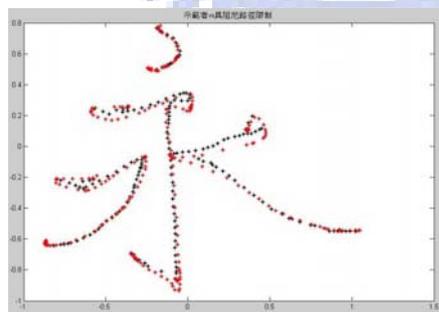
(d)使用輔助系統後結果



(e)路徑限制模式下軌跡



(f)路徑限制模式下書寫結果

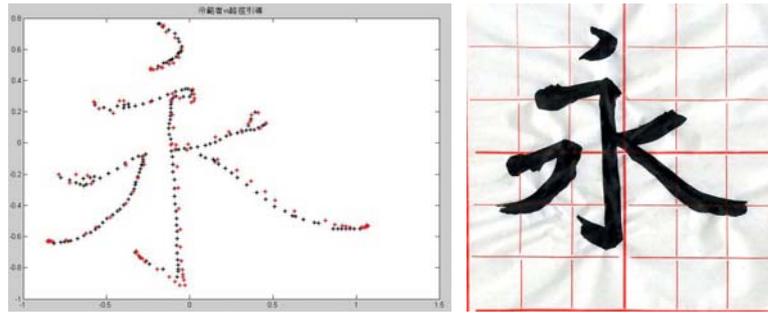


(g)具阻尼路徑限制模式下軌跡



(h)具阻尼路徑限制模式下書寫結果

圖 4-5 受測者二在輔助前後與三個輔助模式中的操作軌跡與書寫結果(續)



(i)路徑引導模式下軌跡

(j)路徑引導模式下書寫結果

圖 4-5 受測者二在輔助前後與三個輔助模式中的操作軌跡與書寫結果

表 4-2 受測者二：起收筆誤差值統計表

受測者二		
	起筆誤差值(cm)	收筆誤差值(cm)
系統使用前	3.4121	3.7126
系統使用後	1.6675	2.1488
路徑限制	0.9929	1.1041
具阻尼路徑限制	1.1338	1.1686
路徑引導	1.1387	1.1176

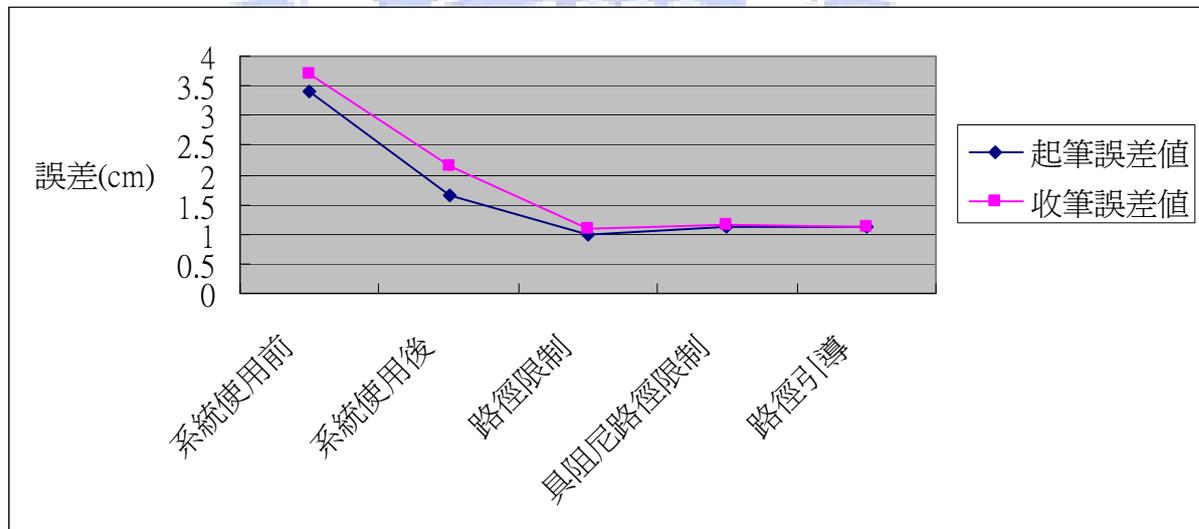
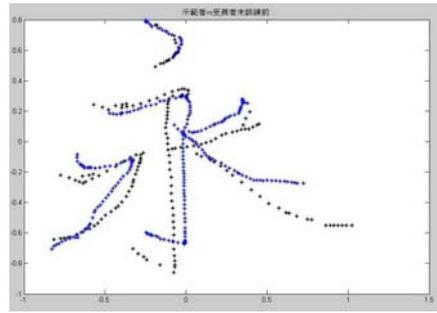
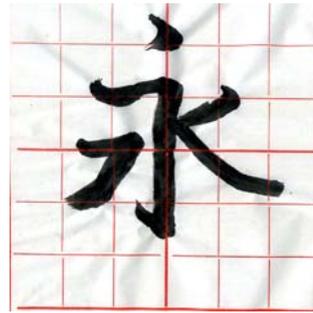


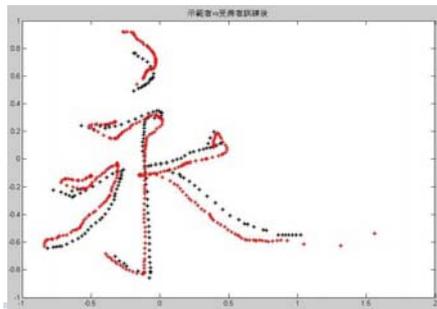
圖 4-6 受測者二：起收筆誤差值折線圖



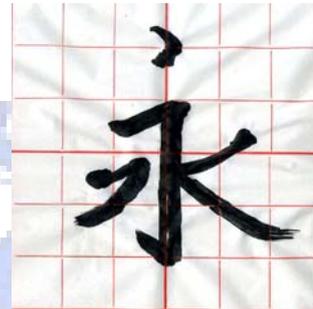
(a)未使用輔助系統前軌跡



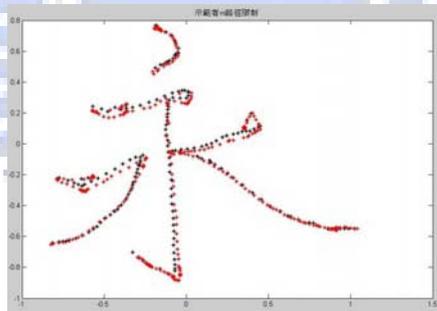
(b)未使用輔助系統前書寫結果



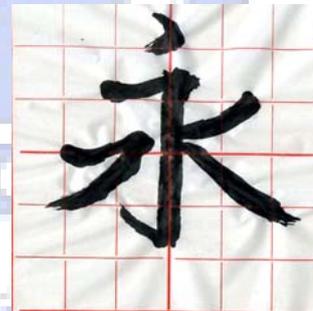
(c)使用輔助系統後軌跡



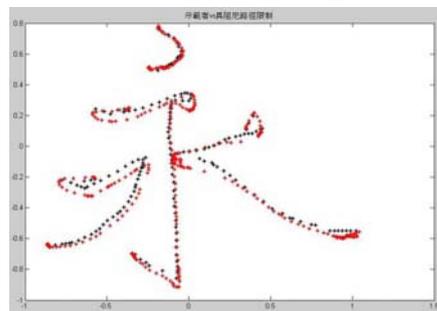
(d)使用輔助系統後結果



(e)路徑限制模式下軌跡



(f)路徑限制模式下書寫結果

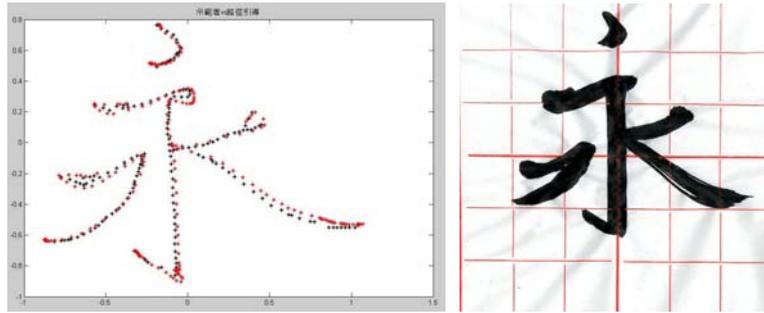


(g)具阻尼路徑限制模式下軌跡



(h)具阻尼路徑限制模式下書寫結果

圖 4-7 受測者三在輔助前後與三個輔助模式中的操作軌跡與書寫結果(續)



(i)路徑引導模式下軌跡 (j)路徑引導模式下書寫結果

圖 4-7 受測者三在輔助前後與三個輔助模式中的操作軌跡與書寫結果

表 4-3 受測者三：起收筆誤差值統計表

受測者三		
	起筆誤差值(cm)	收筆誤差值(cm)
系統使用前	4.644	4.7982
系統使用後	2.6412	2.746
路徑限制	0.8741	0.8892
具阻尼路徑限制	1.2249	1.1371
路徑引導	1.0593	0.852

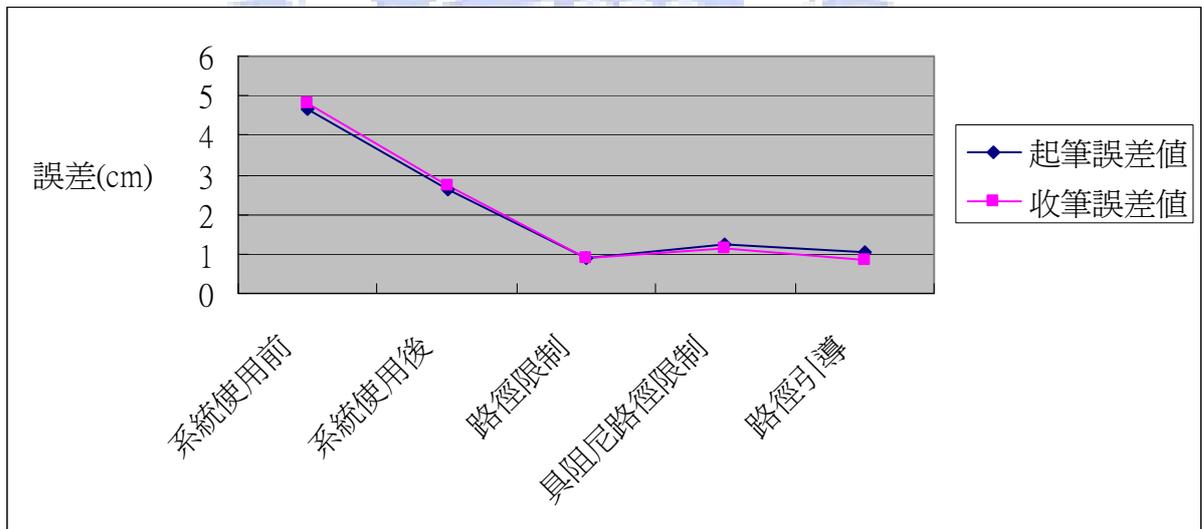


圖 4-8 受測者三：起收筆誤差值折線圖

第五章

結論

在本論文中，我們使用了三軸力回饋搖桿發展了一套毛筆字書寫輔助系統，讓操作者能夠從輔助過程中體會到特定毛筆字的書寫方式，系統中的人機介面提供操作者操控在虛擬場景中的視覺回饋，藉著視覺回饋操作者能夠同時觀察本身操作在虛擬場景中以及書寫在紙上的相對位置，即時修正操作的位置。此輔助系統能夠適用於不同程度的操作者，系統中的路徑限制模式，所適用對象為方向、速度控制上具有不錯的操控能力，但路徑操控能力欠佳的操控者；具阻尼路徑限制模式，適用對象為方向、速度控制上具有不錯的操控能力，但在轉彎處的操控能力欠佳的操控者，以及路徑引導模式，對象為方向、速度和路徑操控能力都欠佳的操控者。從實驗結果來看，透過輔助模式使得受測者與示範者在操作軌跡上的誤差值能夠被大大降低，在書寫結果上，受測者透過三種輔助模式書寫出來的結果都有一定程度的相似性，而且都比使用系統前後要好，因此本輔助系統確實適用於不擅長毛筆字書寫的操作者，能夠對於操作者手部的動作提供輔助，幫助操作者進行書法字的學習。

5.1 未來工作

對於我們所提出的毛筆字書寫輔助系統，在將來有以下可以再作延伸：

1. 力搖桿操控柄與紙面角度不同會造成取樣軌跡失真，我們希望在未來能夠把操控柄的旋轉參數納入考量，決定出實際操作者軌跡的取樣記錄，並且以視覺回饋的方式，提示操作者該操控柄的傾斜角度。
2. 除了以示範者的操作軌跡作為操作者的學習路徑，另外可考慮提供示範者的操控速度對操作者進行力引導。
3. 可設計出具智慧的字體筆劃切割演算法，經由此演算法省去人工判斷的步驟，藉此提昇效率。
4. 建立字體資料庫，使操作者能夠自行決定所要練習的特定字體，並且能夠輸入操作者的軌跡至資料庫中，作為其它操作者學習的樣本。
5. 在重覆使用輔助系統後，操作者對於字的書寫會有一定程度上的操作熟練度，使得操作者的字體優劣無法再從書寫結果來評斷，因此我們希望能夠設計出一套具精確性的評斷法則，作為操作者自我評估的準則。

參考文獻

- [1] N. Law, W. W. Ki, A. L. S. Chung, P. Y. Ko and H. C. Lam, “Children's stroke sequence errors in writing Chinese characters”, *Reading and Writing: An Interdisciplinary Journal*, Vol. 10, No. 3-5, pp. 267-292, 1998.
- [2] H. C. Lam, K. H. Pun, S. T. Leung, S. K. Tse and W. W. Ki, “Computer-assisted-learning for learning Chinese characters”, *An International of Chinese and Oriental Languages Processing Society*, Vol. 3, No. 1, pp. 31-44, 1993.
- [3] C. K. Tan, “An algorithm for online strokes verification of Chinese characters using discrete features”, *Frontiers in Handwriting Recognition*, pp. 339–344, 2002.
- [4] M. A. Eid, M. Mansour, A. H. El Saddik, and R. Iglesias, “A haptic multimedia handwriting learning system”, *Educational multimedia and multimedia education*, pp. 103 - 108, 2007.
- [5] R. Palluel-Germain, F. Bara, A. Hillairet de Boisferon, B. Hennion, P. Gouagout and E. Gentaz, “A visuo-haptic device-telemaque-increases kindergarten children’s handwriting acquisition”, *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 72-77, 2007.
- [6] K. Henmi and T. Yoshikawa, “Virtual lesson and its application to virtual calligraphy system”, *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1275-1280, 1998.
- [7] C. L. Teo, E. Burdet, and H. P. Lim, “A robotic teacher of Chinese handwriting”,^h *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 335–341, 2002.
- [8] K. C. Finney, *3D game programming all in one*, Thomson/Course Technology, Boston, 2004.
- [9] C. B. Zilles, J. K. Salisbury, “A constraint-based god-object method for haptic display”,

- IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 3146, 1995.
- [10] Y. K. Kim, and X. Yang, "Hand-writing rehabilitation in the haptic virtual environment" *Haptic Audio Visual Environments and their Applications*, pp. 161-164, 2006.
- [11] Y. K. Kim and X. Yang, "Real-time performance analysis of handwriting rehabilitation exercise in haptic virtual reality", *Conference on Electrical and Computer Engineering*, pp. 1357-1360, 2007.
- [12] J. Solis, C. A. Avizzano and M. Bergamasco, "Teaching to write Japanese characters using a haptic interface", *Symposium on Haptic Interfaces For Virtual Environment and Teleoperator Systems*, pp. 255-262, 2002.
- [13] J. Mullins, C. Mawson, and S. Nahavandi, "IEEE International Conference on Haptic handwriting aid for training and rehabilitation", *Systems, Man and Cybernetics*, Vol. 3, pp. 2690-2694, 2005.
- [14] T. Yoshikawa and K. Henmi, "Human skill transfer using haptic virtual reality technology", *International Symposium on Experimental Robotics*, pp. 362, 1999.
- [15] W. Daniel, Z. Yuru, and Y. Chong, "Stroke-based modeling and haptic skill display for Chinese calligraphy simulation system", *Virtual Reality*, Vol. 9, pp. 118-132, 2006.
- [16] G. Srimathveeravalli and K. Thenkurussi, "Motor skill training assistance using haptic attributes", *First Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 452- 457, 2005.
- [17] W. Dangxiao, Z. Yuru, and Y. Chong, "Machine-mediated motor skill training method in haptic-enabled Chinese handwriting simulation system", *IEEE International Conference on Intelligent Robots and Systems*, pp. 5644-5649, 2006.
- [18] S. Saga, N. Kawakami, and S. Tachi, "Haptic teaching using opposite force presentation", *World Haptics Conference*, pp. 18-20, 2005.
- [19] K. T. Tang, K. K. Li and H. Leung, "A web-based Chinese handwriting education

system with automatic feedback and analysis”, *International Conference on Web-based Learning*, Vol. 4181, pp.176-188, 2006.

[20] Sensable Co., The phantom omni developer kit, version 2.0, 2005.

