

# 國立交通大學

資訊科學與工程研究所

碩士論文

空中繪圖-利用WiiRemote在空中軌跡的資訊  
當作繪圖的輸入



研究生：陳穎盛

指導教授：李嘉晃 教授

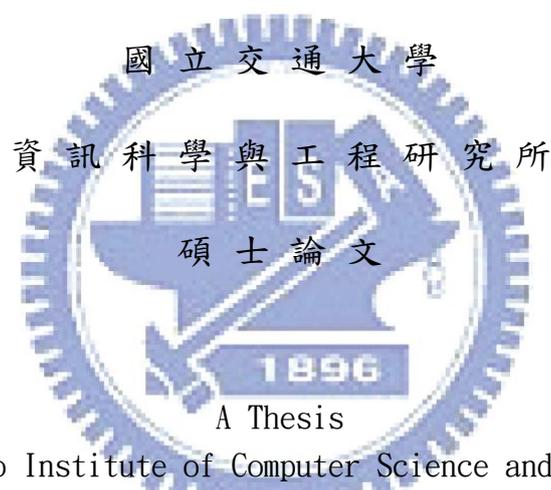
中華民國九十七年六月

空中繪圖-利用WiiRemote在空中軌跡的資訊當作繪圖的輸入

Painting in the air -Using the trajectory of WiiRemote as the  
input of painting device

研究生：陳穎盛            Student：Ying-Sheng Chen

指導教授：李嘉晃        Advisor：Chia-Hoang Lee



A Thesis  
Submitted to Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
In

Computer Science  
Jun 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

# 空中繪圖-利用WiiRemote在空中軌跡的資訊 當作繪圖的輸入

學生：陳穎盛

指導教授：李嘉晃 教授

國立交通大學資訊學院 資訊科學與工程研究所碩士班



這兩年任天堂所開發出的 Wii Remote，其主要是在家用主機上提供了一個新穎的操作方式，讓人們在遊戲平台上能夠透過一種獨特的懸空操作方式，達到使用上的便利效果。本研究主要是將 Wii Remote 的這種特殊的操作方式，應用到一般的桌上型電腦上，利用 Wii Remote 回饋的資訊當做滑鼠的游標，並且分析而得到其軌跡資訊，來辨識使用者所輸入的文字，如此，可以讓使用者更加直覺性的使用 Wii Remote 來繪圖，增加繪圖的便利性和趣味性。

# Painting in the air -Using the trajectory of WiiRemote as the input of painting device

Student : Ying-Sheng Chen

Advisor : Prof. Chia-Hoang Lee

Department of Computer and Information Science  
National Chiao Tung University

## Abstract

The logo of National Chiao Tung University is a circular emblem with a gear-like outer border. Inside the circle, there is a stylized building or structure with the letters 'CIS' and the year '1896' visible. The logo is semi-transparent and serves as a background for the abstract text.

The Wii Remote, developed by Nintendo two years ago, provides a novel controlling method for game consoles. It provides an unique wireless controlling method and it is easy and convenient for users to use it in the game system. This research primarily applies the Wii Remote controlling method to desktop application, which transforms Wii Remote' s feedback data into the cursor information and identifies the words from the feedback data. In essence, users can obtain different experience from the magic of Wii Remote controller.

# 目錄

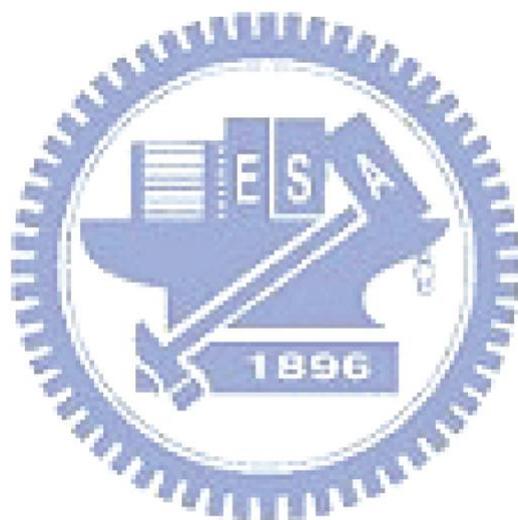
摘要.....	i
目錄.....	iii
表目錄.....	v
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目標.....	2
1.3 論文架構.....	2
第二章 背景知識.....	3
2.1 Wii Remote 的組成.....	3
2.2 軌跡資訊.....	4
2.2.1 手寫辨識.....	4
2.2.2 Free Sketch 辨識.....	5
第三章 系統架構.....	6
3.1 整體架構.....	6
3.2 WiiRemote 輸入.....	7
3.3 軌跡處理.....	8
3.3.1 Noise filter.....	9
3.3.2 Vibration filter.....	9
3.4 文字跟圖形的分類.....	10
3.4.1 中文.....	10
3.4.2 日文.....	11
3.4.3 英文和圖形.....	12
3.5 軌跡方向的判定.....	12
3.6 字庫的建立與比對.....	14
3.6.1 字庫的建立.....	14
3.6.2 字庫的評分和比對.....	16
3.7 圖形評分.....	19
第四章 實驗過程與結果討論.....	22
4.1 實驗說明.....	22
4.1.1 系統的架設.....	23
4.2 實驗過程.....	24
4.3 實驗結果與討論.....	25
第五章 結論.....	26
5.1 研究總結.....	26
5.2 未來工作.....	26
參考文獻.....	27

## 圖目錄

圖 1 WiiRemote 的使用方式 .....	1
圖 2 Wii Remote 內部晶片 .....	3
圖 3 Sensor Bar .....	3
圖 4 Nakai 所利用的 24 種筆劃 .....	4
圖 5 左邊為 Free Sketch 原圖 右圖為找出轉角與直線 .....	5
圖 6 系統流程圖 .....	6
圖 7 Wiin Remote 軟體所繪製出紅外線光點的位置 .....	7
圖 8 系統紀錄抖動的情況 .....	8
圖 9 least square line 示意圖 .....	9
圖 10 中文軌跡的種類 .....	11
圖 11 日文軌跡的種類 .....	11
圖 12 英文和圖形軌跡的種類 .....	12
圖 13 軌跡的分類 vector .....	13
圖 14 軌跡方向編碼流程圖 .....	14
圖 15 中文日文 encode 的方式 .....	15
圖 16 英文 encode 的方式 .....	16
圖 17 系統辨識中文評分後的產生圖 .....	17
圖 18 系統辨識日文評分後的產生圖 .....	17
圖 19 系統辨識英文評分後的產生圖 .....	19
圖 20 三角形手繪辨識圖 .....	20
圖 21 方形手繪辨識圖 .....	21
圖 22 三角形手繪辨識圖 .....	21
圖 23 透過 BlueSoleil_2.3 連接 Wii remote .....	23
圖 24 glove pie communication script .....	23
圖 25 系統環境架構 .....	24
圖 26 左圖為正常的 W 右圖為歪斜的 W .....	25

## 表目錄

表 1 本研究使用的相關硬體設備和軟體支援.....	22
----------------------------	----



# 第一章 緒論

## 1.1 研究動機

近年來任天堂公司開發一套 console game 名叫 Wii，他的操作方式是相當獨特且創新的，在這款主機剛上市的時候，帶給許多玩家相當大的震撼，因為它給予玩家一種身臨其境的感覺。它提供玩家一支搖桿(WiiRemote，如下圖 1)，使玩家的動作對應到遊戲中角色的動作，讓玩家可以使用這支搖桿輕鬆操控遊戲中的角色，更快的進入遊戲情境，且操作也更加的直接。



圖 1 WiiRemote 的使用方式

而在 HCI(Human Computer Interface)的領域裡，一直希望可以提供大眾一種 Augmented Environment，但一般個人電腦的輸入裝置，大都來自於鍵盤或者是滑鼠，而這兩項硬體設備往往讓整個使用的空間增加不少，並且在操作上並不是那麼的直接，倘若輸入文字可以像一般寫字一樣，不僅不需要花費時間在訓練打字上，並且讓電腦的操作變得更加簡單容易上手。換言之，若我們可以利用 Wii Remote 當做電腦的輸入設備，那麼僅需在空中揮舞寫字，就可以達到電腦的輸入功能，這樣一來，使用上將能非常的直接，且不佔空間。

## 1.2 研究目標

本系統主要是結合 WiiRemote 與桌上型電腦，探討利用 WiiRemote 來當作電腦的輸入設備，因此，還需要一套可以和 WiiRemote 溝通的系統，此系統要能分析利用 WiiRemote 蒐集到的軌跡資訊，來當作文字辨識上的基礎。對這些軌跡提供一個模組來方便我們系統上的辨識，並且對於不同的文字給予不同的系統架構，然後利用這些架構，開發出一套繪圖的軟體，此軟體提供使用者基本的繪圖和文字輸入功能。

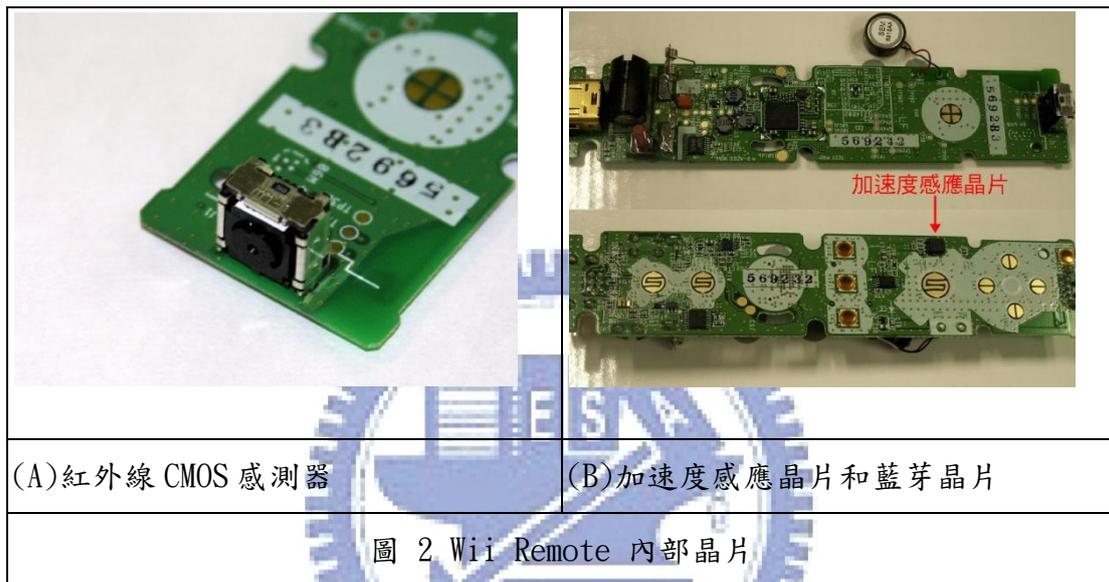
## 1.3 論文架構

第一章為緒論，主要是介紹做此研究的動機並且簡單的敘述整個系統的架構。第二章背景知識會針對個人電腦和用 WiiRemote 溝通的方式做一個介紹，以及一些 Free Sketch Recognition 的方法。第三章系統架構將解說如何把蒐集到的軌跡資訊做一個處理，並對不同的文字有不同的評分方式。第四章實驗結果將包括測試系統的正確率與了解一般使用者的使用經驗。第五章結論會去討論此系統的優缺點，並且了解此系統還需要改善的地方。

## 第二章 背景知識

### 2.1 Wii Remote 的組成

Wii Remote 主要是由紅外線 CMOS 感測器、藍芽控制晶片和一個加速度感測晶片所組成(如下圖 2 所示)[1]。



利用 Wii Remote 前端的 CMOS 感測器，搭配與 Wii 本機連結的 Sensor Bar (Sensor Bar 是一支紅外線發射器(圖 3)，在左右兩端各分別為三顆紅外線 LED 燈)，藉由 Wii 內建的藍芽晶片[2]，把感測到紅外線的座標位址傳送給 Wii 主機本身，便可以得到 Wii Remote 的座標。



圖 3 Sensor Bar

## 2.2 軌跡資訊

軌跡資訊是完整的紀錄游標曾經經過的座標，而在這麼雜亂的座標資訊中，要如何把有用的資訊提領出來，並且經由這些資訊，幫助我們來辨識使用者想要輸入什麼文字，或者他想要畫什麼圖形，已經有許多論文對此有相當研究，而其主要分為兩個部份，一為手寫辨識，二為 free sketch 辨識，前者主要是分析字體為主，後者主要是用來分析圖形。

### 2.2.1 手寫辨識

手寫辨識近年來快速竄起，大部份的方法主要分為兩種，一種是利用固定式的 camera 拍攝使用者所寫好的字跡，與資料庫去比對已經建立好的字體形狀 [3]，找出最接近的筆跡，進而判斷是誰的字跡；另一種則是去分析每一筆劃的方向，先確認筆畫的方向後，再整合所有筆畫的資訊，從已經建立好的字典裡面尋找相同的筆畫資訊，即為所辨識的文字。其中 Nakai, M[4] 等人把中文字分析成 24 種筆畫，並利用 HMM 的方式，把比較不可能的筆畫更改成比較可能的筆畫，然後得到最有可能的字(圖 4)；陳[5]的研究則把 24 種不同的筆畫簡化成 5 種筆畫，而僅僅只利用這五種筆畫，把所有中文字建立成字典資料，最後則把筆畫的資訊與字典資料配對，歸納出最接近的配對資訊，此方法在中文的辨識率頗高，因此本研究主要是利用此方法的架構，進而應用到中文、英文、還有日文的領域上。

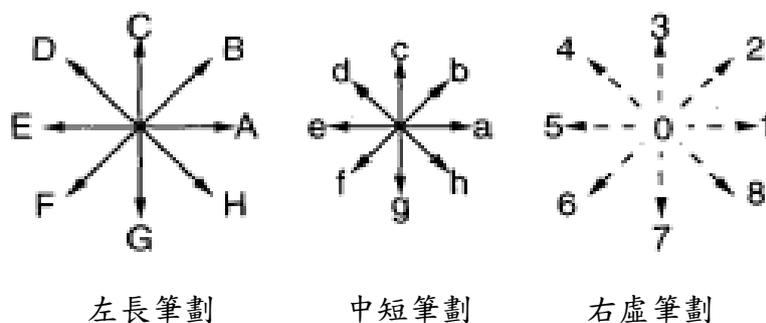


圖 4 Nakai 所利用的 24 種筆劃

## 2.2.2 Free Sketch 辨識

Free Sketch 辨識主要是要從我們隨手畫的圖形，把它分類成許多小部分，一般來說圖形大致上可以分為三種不同的種類，分別為直線、弧形和轉角，用這三種分類可以囊括所有圖形，舉例來說，三角形就是由三個轉角和三條直線所構成，圓形就是由許多弧形的所構成，Tevfik 等人[6]用時間跟曲率來判斷是否為轉折點，進而再將這些轉折點前後一個範圍，取歐式距離  $L1$  和實際弧形的距離  $L2$ ，當  $L2/L1$  的比值接近一時，我們可以判斷他為一個直角，比值大於一時，則可以判斷它為一個弧形而非轉角。經由這個方法我們可以把一般手繪的圖形，變成比較有稜有角(圖 5)。

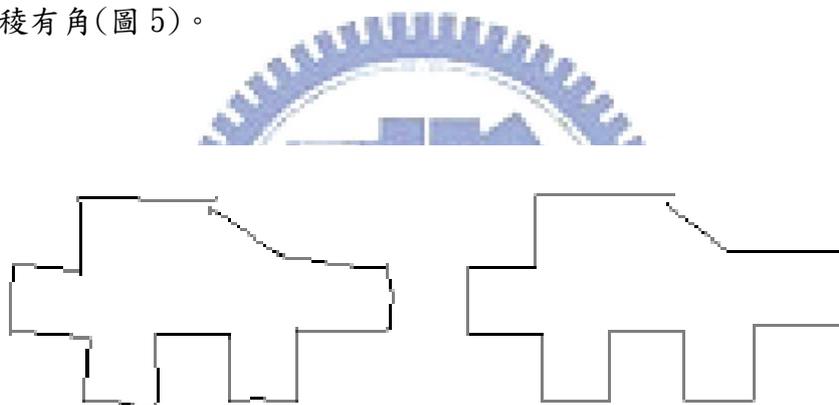


圖 5 左邊為 Free Sketch 原圖 右圖為找出轉角與直線

# 第三章 系統架構

## 3.1 整體架構

本研究主要是分析 Wii remote 移動的軌跡，把得到的軌跡資訊，做了一個去雜訊的處理，然後提供之後的辨識；而在筆劃分類的方式，則因應不同的文字的特性，採取不一樣的分類，本系統提供了三種語言的辨識和三種圖形的辨識，大致上會分為三種不同的類別，下圖為系統整體架構圖；

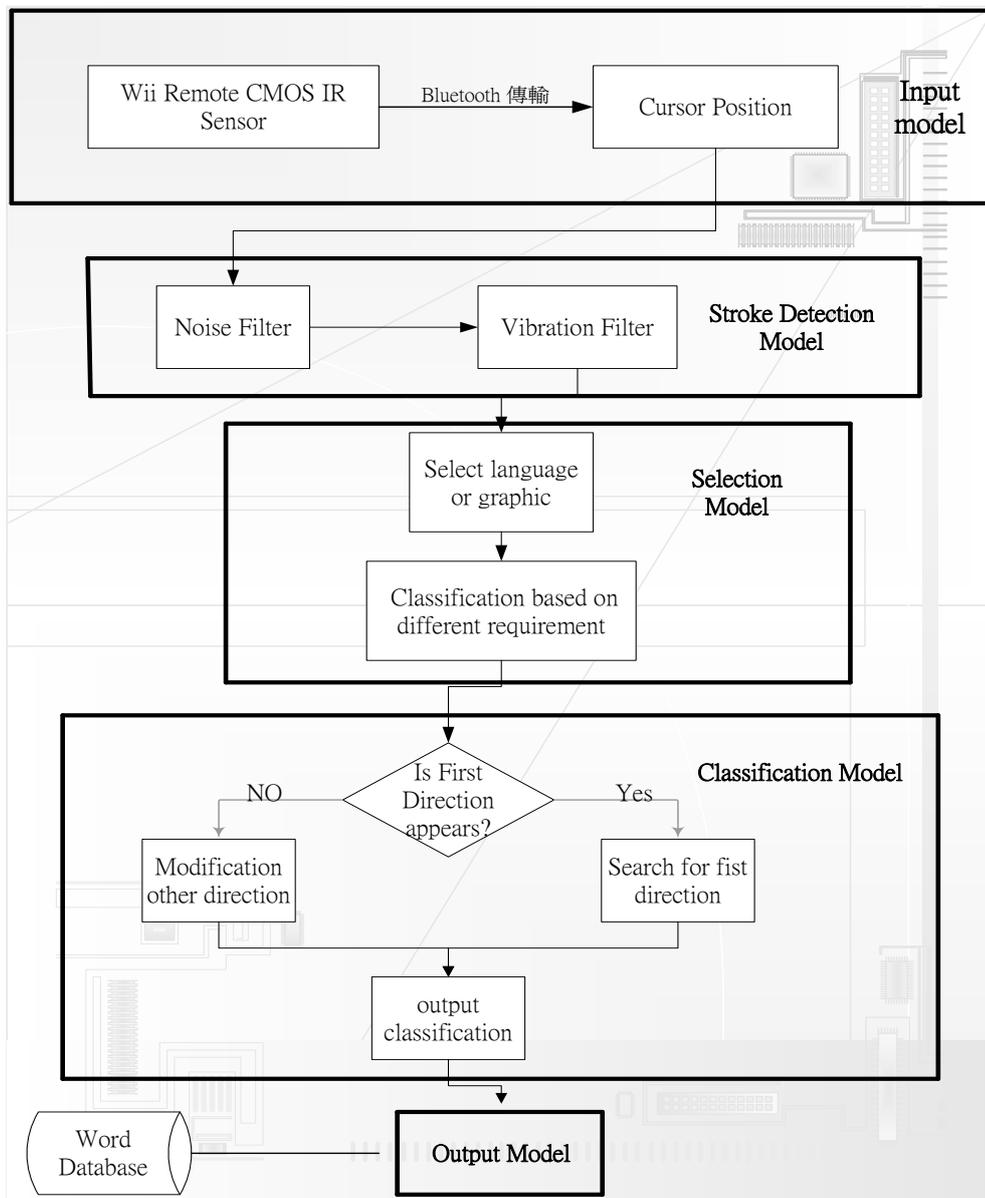


圖 6 系統流程圖

從架構圖我們可以看到系統主要可以分為四個部份，第一部分 Input model 是把 Wii Remote 內建 CMOS 紅外線感測到兩個光點經由藍芽傳入電腦，第二部份為 Stroke model、Selection model，這部分主要是把傳進來的軌跡做一個雜訊的弭除，第三部分為 Classification model 把處理過後的軌跡做一個分類，第四部分則是利用分類完的軌跡，去字庫裡面做一個配對的動作，不同類別的文字有不同的配對和評分方式。

### 3.2 WiiRemote 輸入

這部分的章節主要是介紹如何把 CMOS 感測到的光點，對應成滑鼠的游標；系統一開始利用 Wii Remote 內建的 CMOS 紅外線感測器來感測 IR Sensor 所發出的紅外線光源，我們可以把感測器視為一個紅外線的 Camera，把找到紅外線光點的位置經由藍芽傳回個人 PC 中，取兩個光源的中間點，可視為滑鼠現在的位置，圖 7 為利用日本公司所開發的一套軟體 Wiin Remote[7]，它可以顯示 CMOS 目前紅外線光點的位置，圖中下方可見白色光點 1 和白色光點 2，此兩點分別為兩個紅外線的位置，而中間黃色光點即為兩紅外線的中心點。

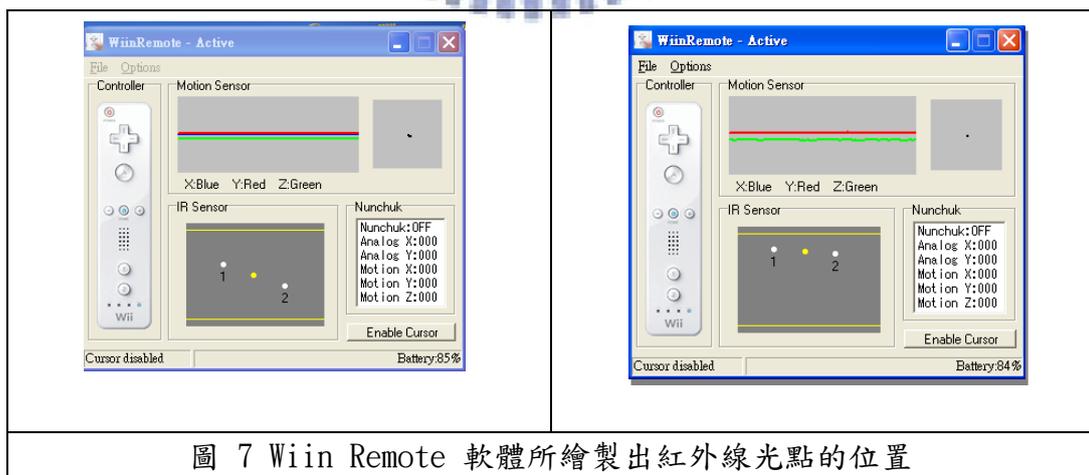


圖 7 Wiin Remote 軟體所繪製出紅外線光點的位置

在第一次接收到藍芽所傳入的資訊，滑鼠游標目前停留的位址就為起始位址，而當第二次在接收藍芽所傳入的資訊時，我們利用 Glove pie [8]這套軟體，可

以得到每次 Wii Remote 這兩次時間點內的位移量，並且根據螢幕的解析度，系統做一個正規化的動作，把所得到的位移量對應成滑鼠游標在螢幕中的位移量；如此，系統便可以開始記錄游標所有的移動的軌跡。

### 3.3 軌跡處理

由 3.2 知道系統可以收集到每次移動方向的資訊，但是這些最原始的資訊也產生了許多的問題。一般而言我們再寫字時，因為手腕有個依靠且有個平台可以讓我們在書寫時有使力點，比較不會有手部抖動的產生，而使用 Wii Remote 在空中書寫時，因為沒有一個明確的使力點，使系統得到的軌跡有許多不預期的抖動如圖 8 所示)，而這些資訊並不是我們想要，系統採用兩個方法來解決此一問題。

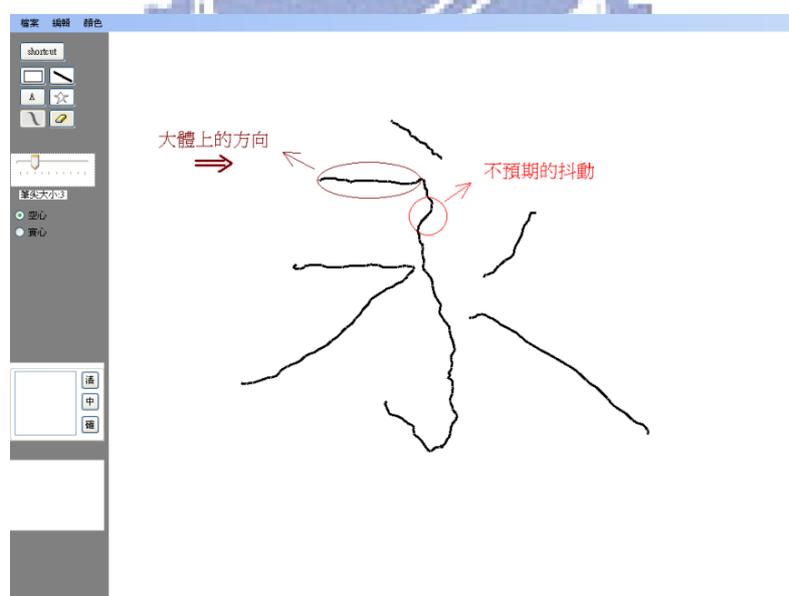


圖 8 系統紀錄抖動的情況

### 3.3.1 Noise filter

Noise filter 主要想解決問題為使用者在空中沒有使力點時會無意識的抖動，而抖動的軌跡並不是系統所想要的資訊，記錄這些資訊會導致分類的困難，因此系統給予一個 threshold ，當這一次與上一次的位移量的值大於 k，系統才會把這個點的資訊記錄下來，否則，系統只能把它當作無意義的抖動，透過這個方式，可以先基本的消除抖動所產生的雜訊。

### 3.3.2 Vibration filter

雖然是不斷的抖動，但因為根據使用者想寫的文字，這些抖動的軌跡，必定會朝向某一個特定的方向，可能是一橫、一撇或者一豎，而要怎麼讓系統知道這些軌跡是我們想要的資訊而不是雜訊，系統採取的是利用最小平方值線的方法 Least square line(圖 9)來求出使用者最有可能輸入的一個方向。

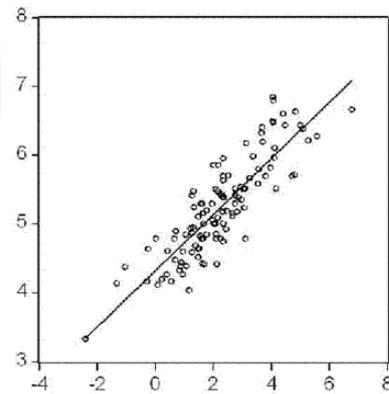


圖 9 least square line 示意圖

Least square line 主要是收集座落在座標空間的點，來找一條最逼近直線，利用這個概念，我們可以找出使用者可能想要畫的一個方向，當有抖動產生時，可以利用這條求出的直線來做一個導正，不會讓方向突然有劇烈的改變。

Least square line 直線的求法如下

$L = ax + b$   $L$  代表所求的最逼近直線

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

系統每當有新的點進來時，使用這次進來的前三個點來求出  $L$ ，把新的點對  $L$  這條直線做投影，取投影在這條直線上的座標和原始的座標的點，取一個權重來當作下一個新點的位置，系統讓原始座標的權重佔三分之二，新進來點的座標佔三分之一，這樣就不會因為抖動的關係導致系統的方向隨時都有巨大的改變，也不能讓這條直線有矯枉過正的情形，因為有可能是真的方向的改變而不是抖動所造成的，所以保留原本的移動方向和較多真實接收到的點，讓方向的變化不會因為一兩次的雜訊，而喪失原本的方向。

### 3.4 文字跟圖形的分類

本系統可以同時支援中文、英文、日文和圖形的辨識，但是每種文字結構的特性都不太一樣，如果都採用同一種軌跡分類，在辨識的準確率上可能有很大的問題，因此，系統針對每一種不同的語言提供不同的軌跡分類。

#### 3.4.1 中文

中文在結構上主要可以分為五個種類，而其中四個種類如下圖，只要在該 Type 的範圍內，系統會把這些方向視為是相同的一種比劃，當有偵測到超過一個範圍的方向時，就把它歸類到 Type5。

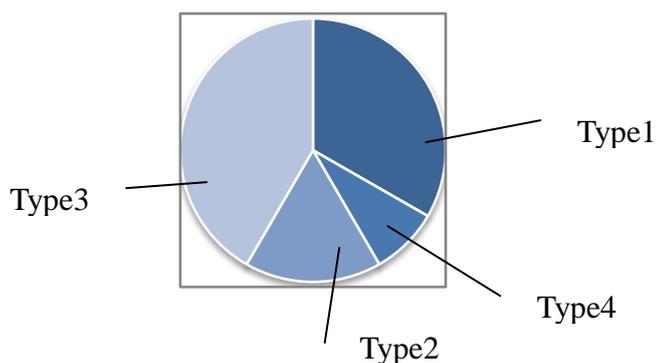


圖 10 中文軌跡的種類

會這樣分的原因是中文比劃幾乎沒有右上跟左上的筆劃，所以可把橫歸類成 type1，豎歸類在 type2，彎為 type3，捺為 type4，再加上一個混合的比劃 type5，便可以把所有的中文字做一個分類。

### 3.4.2 日文

日文方面，系統採用的是平假名的辨識，因為平假名的結構跟中文字其實很像，因此僅僅只加上原本中文沒有的方向，以日文シ si 這個字來說，這個字的第三個比劃為左下到右上的筆劃，於是我們在右上的地方多了一個分類叫做 type5，而混合比劃的部分就當作是 type6，新的分類圖如下。

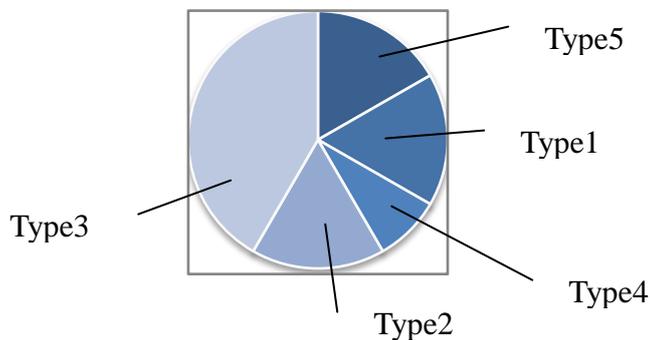


圖 11 日文軌跡的種類

### 3.4.3 英文和圖形

英文與圖形跟中、日文最大的不同是，英文和圖形在結構上都有許多弧形的地方，以英文字母 o 為例，它就包含了所有的方向，而圖形也沒有所謂的特定的比劃與方向，也就是說，每個方向的資訊對系統來說都很重要，因此最簡單想法就是把一個圓形等份的分成八塊，每一塊就是一個分類，但是這麼多的分類對英文而言，只是增加編碼跟辨識的困難度，於是我們把轉折點部份合成一種分類，把 type3~type5 合成 type3，而 type6 和 type7 合成 type7，圖形便不再額外的分類。

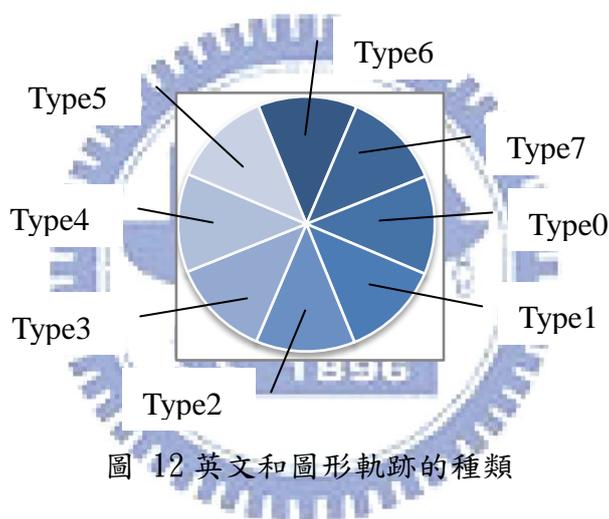


圖 12 英文和圖形軌跡的種類

### 3.5 軌跡方向的判定

從上一節系統針對不同的結構，做了一個適合的分類方式，而要怎麼把這些軌跡的資訊讓系統了解，分類到我們所規定的方向，並且自動的產生方向的編碼，可以提供最後與字庫的比對和評分是這一節的重點。

首先，我們已經知道空中輸入的軌跡是比較不穩定的，但經過先前的處理，我們利用 Least square line 已經可以得到較穩定的方向，但只是這樣還是不夠的，如果光憑得到的方向就認定為該種類的 vector，系統會收到太多錯誤的編碼；於是在這個部分，我們給予每種 Type 一個 vector，以中文來說，type1 的

vector 就是(1, 0)而 type2 的 vector 為(0, -1)如下圖。

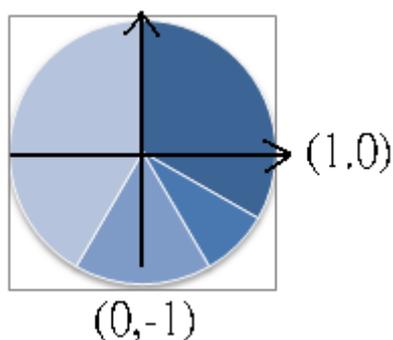


圖 13 軌跡的分類 vector

所以中文除了混合比劃就定義了4種vector，日文和英文分別為5種vector和8種vector，用這些vector再加上一個threshold  $t$ ，把每次已經處理過後的連續兩個點(上一次和這一次的兩點)分別投影到這些vector上，則每經過一次的輸入，所有的vector都會有所改變，主要看兩個向量的夾角，兩個vector的夾角可以以 $90^\circ$ 來做一個分隔，當夾角 $<90^\circ$ 對投影的vector是一個正向的作用，而 $>90^\circ$ 已經是一個副面的作用，如此看來，正方向的vector會很快的累積數值並且超過定義好的 $t$ 值，則這個方向最有可能就是使用者想要輸入的方向，系統就可以輸出該勝出vector的編碼，並且把所有vector再設為初始值，再繼續重複的做加種的動作，一直到輸入結束為止；中文和日文方面，如果輸出的編碼超過一個，就代表它跨越了兩個領域，它就會屬於混合比劃的Type，於是編碼就分別為五(中文)跟六(中文)，英文和圖形的編碼就是原本輸出的編碼，不在做額外的處理，圖14為整個流程圖。

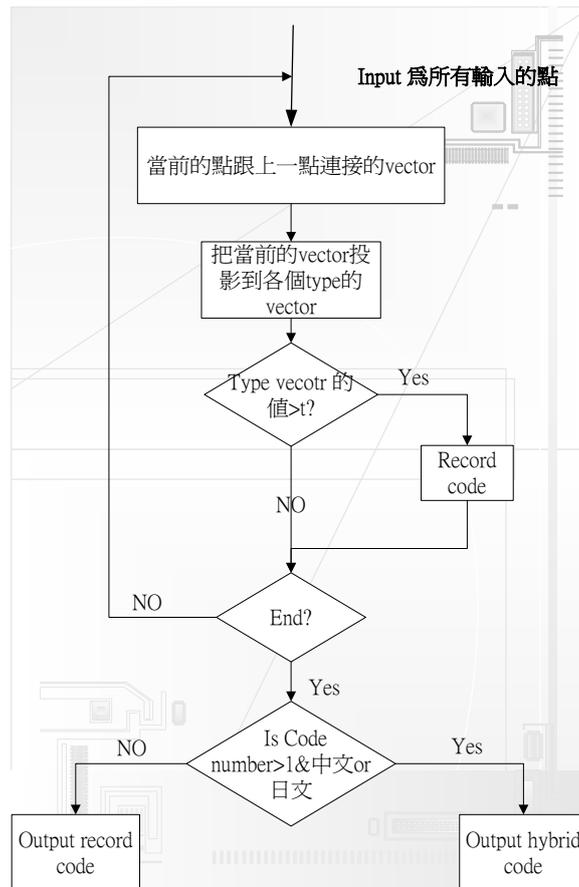


圖 14 軌跡方向編碼流程圖

### 3.6 字庫的建立與比對

經過前面系統介紹系統如何處理軌跡的方式之後，這一節主要介紹的是如何選出最接近使用者所輸入的字，而我們先介紹字庫的建立，再來講解文字的評分標準。

#### 3.6.1 字庫的建立

在說明系統怎麼評分之前，先介紹一下字庫裡面的內容，所有字庫的編碼都是按照先前所講的分類方式來編碼，以下圖為例。

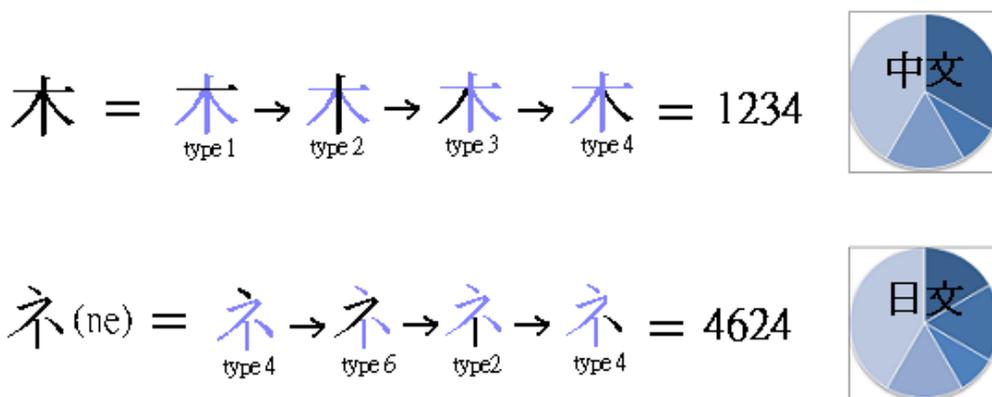


圖 15 中文日文 encode 的方式

木的筆劃第一筆為一橫，這一橫的筆劃座落在 type 1 所以字庫把它 encode 成 1，第二筆為一豎 encode 為 2，第三筆一撇和第四筆一捺就分別 encode 成 3 和 4，所以木這個字最後的編碼就是 1234；日文的編碼跟中文大同小異，以ネ ne 這個日文來看，第一筆為一捺 encode 成 4，第二筆包含了兩個比劃分別為 type 1 和 type 3，這種就是混合比劃，所以 encode 成 6，剩下兩筆同理可知，所以日文ネ ne 的編碼就是 4626，但是因為日文的混合筆劃相當的多，如果都只用 type 6 來編碼這樣準確率會大為下降，於是在這個部分，我們把 type 6 細分為 4 種，端看它結尾的筆劃，結尾如果為 type 1 則 encode 成 7，type 5 則 encode 成 8，type 4 則 encode 成 9 剩下的都不變 encode 成 6，這樣在辨識上的效果可以大大提升；上述這兩種語言都是以分開筆劃為基礎，所以在編碼上面還蠻直觀的，一個筆劃就是一個方向，不在這個範疇裡面的就是混合筆劃，而英文字母因為轉折的部分太多，如果都把它視為混合筆劃，這樣難以取得其特徵值，所以我們把英文的圓弧地帶視為是一種混合筆劃，而圓弧就像是 a、b 這種有圓圈的地方，於是我們把 type 3~type 5 都編碼為 3，type 6~type 7 都編碼為 7 所以英文的編碼就是分別把圓弧和直線的轉折點都記錄下來，以圖 16 為例。

b = ↓b → b → b → b = 2703

t = t → ↓t → t = 020



圖 16 英文 encode 的方式

使用先前所介紹的方法，我們可以建立出中、英、日的字庫，這些字庫即是作為系統比對評分的依據，而圖形因為沒有所謂的筆劃先後的順序，所以並沒有辦法來建立字庫。

### 3.6.2 字庫的評分和比對

在評分方面，系統也是給予不同的語言有不同的評分方式，中、日文的結構相近，所以採用類似的評分方式，英文和圖形的評分則額外的獨立出來。

我們先介紹中、日文的評分，從 3.5 節知道，系統在每一次的輸入之後，都可以輸出一個編碼，這代表每一次的輸入，都可以提供我們兩個資訊，第一個是你輸入了幾次，第二個是輸入的方向，而輸入了幾次就代表這個文字的筆劃數，輸入的方向則可以去字庫裡面一一比對，於是，系統的評分方式就是先去審查你的筆劃個數是否一樣，畢竟一個文字的筆劃個數不一樣，就不可能會是同一個文字；先把不可能的刪去之後，剩下的就由輸出的軌跡和 encode 的編碼去比對，相同個數多的，當然分數就會比較高；舉例來說，以“我”這個字而言，原本的編碼是 3121434，輸入近來的軌跡辨識的 code 3151535，然後把配對到個數再除以編碼的長度，就是系統最後所得的分數，以上例而言，配對到的個數  $K=4$ ，總長度  $L=7$ ，那最後的  $score=4/7$ ，下圖為系統實際情況，因為“的”的編碼是 32511354，在這次評分所得的分數為  $5/7$ ，所以它的排序變成在第一名的位置；



圖 17 系統辨識中文評分後的產生圖

相對於中文而言，日文的評分方式也可以套用中文的方法，但有一個嚴重的問題就是，日文的編碼通常都很短，最長的日文字也才 4 個筆劃就結束了，這樣如果只是單純的使用中文評分方式，會有許多文字產生一樣的分數，使得辨識上造成模糊的情況，所以除了在混合筆劃更加細分之外，評分上也加了一些程序，主要是以方向角度的差異，有不同的標準，舉例而言ト to 本來 encode 為 24，滿分的話當然是辨識出來的 code 為 24，但如果抓取到的結果是 22 or 21 因為 type 4 跟 type 2 和 type 1 是在相鄰的 type 所以會再給予少數的分數，當作是誤判但還有可能是，那如果辨識的結果是 25 or 23 就不在給予分數，因為已經不是相鄰的 type，差異有點遠，不能僅僅只是當作誤判，用這樣的方法，在評分上提供了很大的幫助，下圖為日文的辨識結果。



圖 18 系統辨識日文評分後的產生圖

英文在評分上採取跟前兩種語言截然不同的方式，主要是轉折點的特徵值很容易辨識錯誤，抓錯跟多抓的情形會大幅提升，且英文幾乎都是一筆就結束了整個文字的輸入，所以少了很重要的筆劃資訊，只能單純的使用辨識出來的資訊當作評分的依據。系統在這塊評分的方式利用 LCS(longest common subsequence) [7]的概念來做比對，LCS 主要是用來找尋兩個字串最長的相同子字串，公式如下。

Given the sequences  $X_{1...m}$  and  $Y_{1...n}$

$$LCS(X_{1...m}, Y_{1...n}) = \begin{cases} \emptyset & \text{if } m = 0 \text{ or } n = 0 \\ LCS(X_{1...m-1}, Y_{1...n-1}) + X_m & \text{if } X_m = Y_n \\ \max(LCS(X_{1...m}, Y_{1...n-1}), LCS(X_{1...m-1}, Y_{1...n})) & \text{else} \end{cases}$$

但只利用 LCS 還不太足夠，因為只抓最長的子字串會產生一個問題，系統如果漏抓了特徵值的情況會導致相同的子字串非常的短，可是其它所辨識的方向是對的，相對來說這個字應該分數要較高但卻給了較低的分數，而導致辨識的錯誤，舉例來說：b 的 encode 為 3814，系統漏抓了 8 造成輸出的編碼為 314，則這樣取出來的 LCS 只有 14 這個字串，所得到的分數將會非常的低，為了補償這種情況的發生，我們再引進一個萬用字元的觀念； $Y^*$ 所代表的是一個萬用字元，它可以補償一次漏抓的失誤，並給予一個次一級的分數，不會讓 $Y^*$ 的分數大於原本就 encode 正確的分數，以上例而言，輸出的編碼為 314 在最後比對的時候會變成 3  $Y^*14$ ，這樣 LCS 取出來的子字串就是 3  $Y^*14$ ，但是因為它包含了萬用字元，所以會給予較次一級的分數，而系統只能補償一次的漏抓，如果給予兩個萬用字元，很容易就把原本應該更為近似的文字排除；經過這樣補償的作用，使得容錯性大為提升，便是成功率也相當的高，下圖為英文辨識的結果。



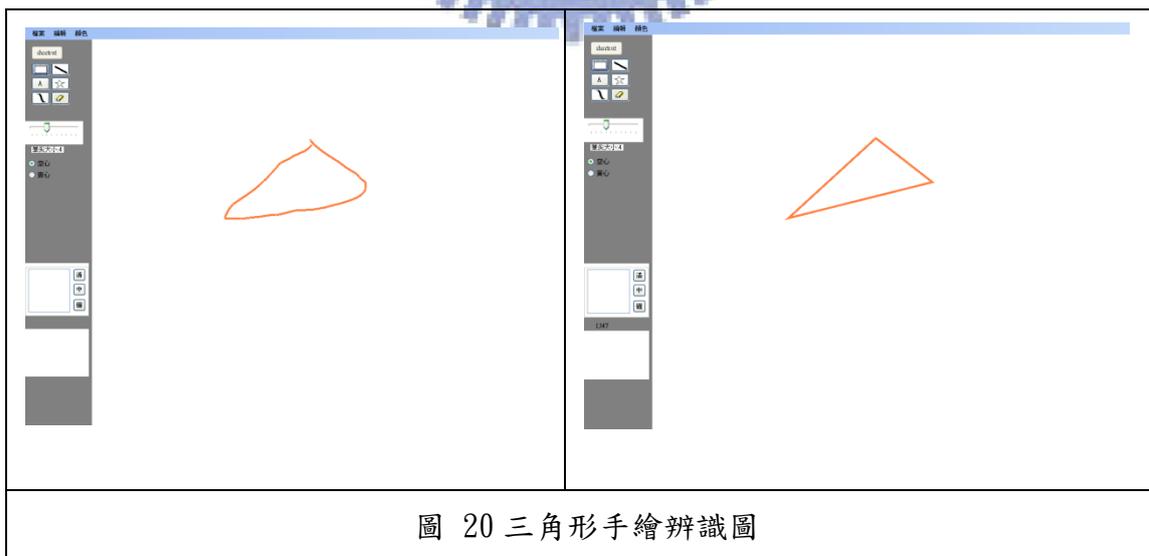
圖 19 系統辨識英文評分後的產生圖

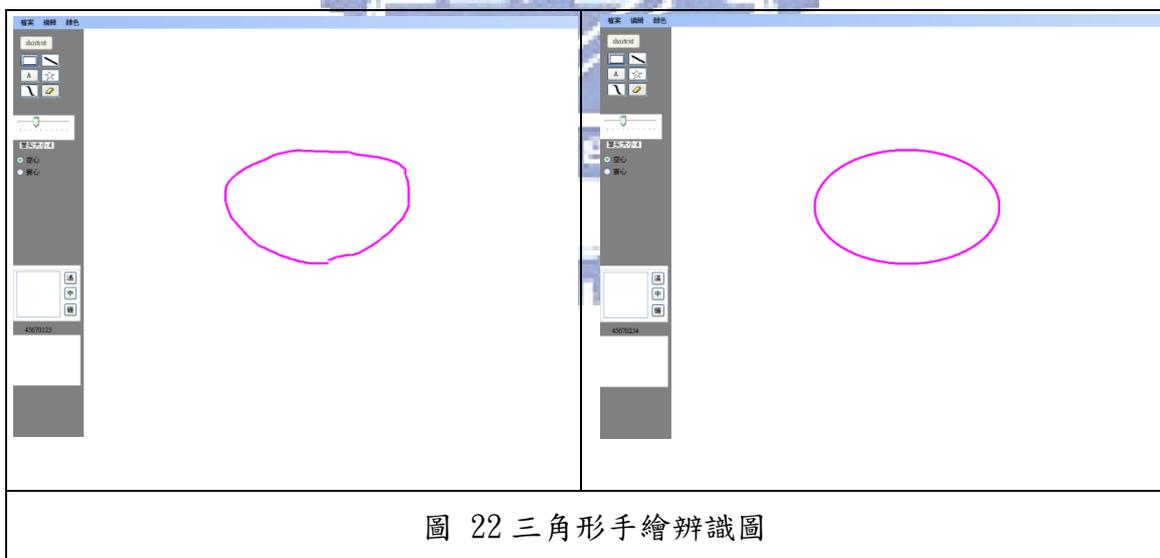
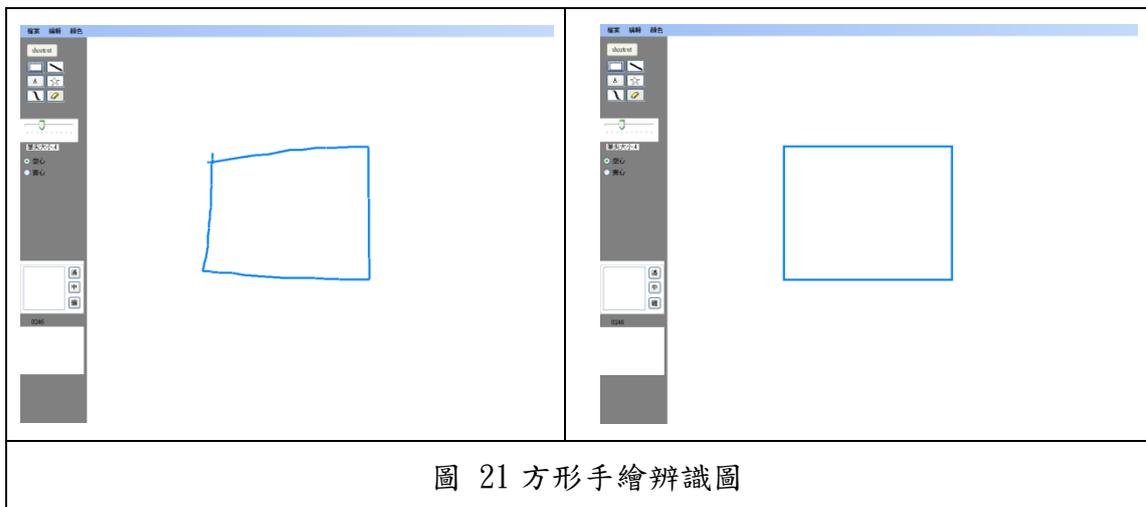
### 3.7 圖形評分

最後要介紹的是圖形的評分，圖形在辨識上系統提供了三種辨識的圖形，分別為三角形、正方形和圓形，因為圖形沒有固定軌跡的先後順序，也沒有統一的一種畫法，所以唯一能得到的只有軌跡的變化，也就是轉折次數的取得。

利用轉折次數來判斷圖形其實還蠻直觀的，舉例來說，三角形就只有兩個轉折，正方形的話有三個轉折，然後圓形就是超過三個轉折以上，如果很正確的接收沒有雜訊的軌跡辨識， $N$  個轉折點就代表  $N+1$  個輸出的軌跡編碼，但通常在空中繪圖軌跡的辨識常常會有錯誤產生，尤其是在方向的變化，如果轉折點的角度不明顯的話，根據系統的特性，原本用來修正的直線會把轉折的角度變小，舉例來說，輸入一個三角形，它原本輸入的軌跡是 317，但很容易辨識成 3217，因為轉折的角度可能介於  $45^\circ-90^\circ$ ，使得 vibration filter 在這地方的修正讓變化的角度變得不明顯，而誤判了 2 這個方向的產生，這種情形，根據原本的系統定義的方式下，變成抓到了 3 個轉折點，因此系統並不會把它辨識成三角形；為了解決這種問題，系統在這邊也容許一次的錯誤，也就是說，當它擷取到四個軌跡的時候，會再加入額外的判斷式，來判斷是否為方形，如果不是的話則認定它是三角形，而截取到五個軌跡的時候同理可知，有可能是方形但多抓了一個軌跡，所以也加入判斷式來判斷是否為圓形，不是的話則視為正方形，判斷式主要是來

檢查是否有直角的產生，因為正方形每次的轉折都是一個直角，以系統的觀點來看，輸入的 sequence 為  $X_{1...m}$ ，檢查  $X_{i+1} - X_i \quad \forall i < m$  是否為 2，是的話就是一個直角，正方形的三個轉折點應該都是直角，所以加入這個判斷式，來判斷是否為正方形，並且給予一次抓錯的機會。而圖形辨識在這邊只做完了一半的工作，接下來的工作是要把圖形繪出，在這邊系統不使用 nakai[4]等人所提出的方法，因為轉折點無法利用時間跟方向的變化來取得，主要是空中的繪圖，常常因為使用者的關係，轉折點不一定是會比較慢，畫直線也不一定會比較快，而方向的變化由先前的探討也可以得知是很不穩定的，在這兩個資訊都不穩定的情況下，取得的轉折點正確率自然就很低；系統這邊使用的方法是採取圖形曾經畫過的範圍，取可以容納這個範圍最大的方形或者是圓形，而三角形因為只有三個點，所以系統先固定好兩個點，再來利用曾經畫過的最遠距離來當作第三個點，因為第三個點一定是曾經畫過的最遠距離，利用這些方法，圖形的辨識與繪圖可以得到一個穩定的效果，圖 20 為手繪三角形和經由系統產生辨識後的三角形的情形，圖 21 和圖 22 分別為方形和圓形。





## 第四章 實驗過程與結果討論

### 4.1 實驗說明

本系統首先利用 wii remote 來得到軌跡資訊，接著對軌跡進行處理變成系統可使用的編碼，使用這些編碼來進行辨識且搭配繪圖的功能，來實做一個空中繪圖系統。實驗的主要目標是探討系統的穩定度與一般使用者的使用心得，整個架構如下表。

表 1 本研究使用的相關硬體設備和軟體支援

遙控器(Input)	Wii remote
藍芽接收器	billionton class 1 bluetooth
中央處理器	Intel Core 2.13 GHz
記憶體容量	1-GB Ram
藍芽驅動程式	BlueSoleil_2.3
紅外線光源發射器	Wii 提供的 sensor bar
Wii communication software	GlovePie

本次實驗邀請了八名受測員，包含了七名男性和一位女性，其中男性都有曾經使用過 Wii remote 的經驗，而女性則沒有這方面的經驗，而實驗方式主要是請使用者分別使用繪圖的功能和辨識的功能，辨識方面包含全部的英文跟圖形還有部分的中、日文。

### 4.1.1 系統的架設

首先，系統要先讓 Wii remote 連接上電腦的藍芽裝置(圖 23)，

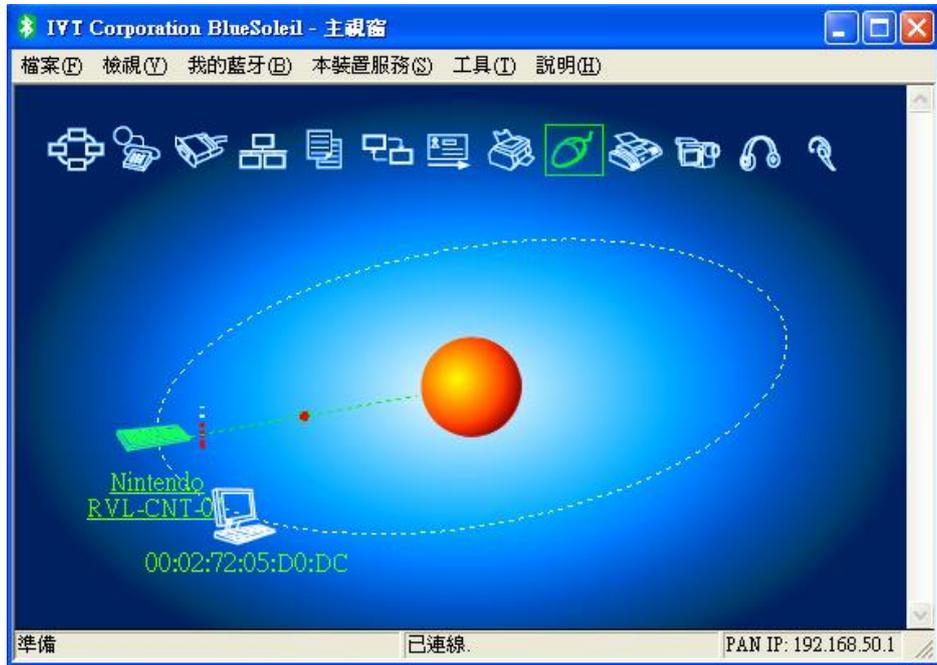


圖 23 透過 BlueSoleil\_2.3 連接 Wii remote

連接上 Wii Remote 之後再透過 Glove Pie 來做一個軌跡 mapping 到滑鼠游標溝通的橋梁(圖 24)，再搭配 sensor bar，整個實驗環境為圖 25，便可以開始進行系統的實驗。

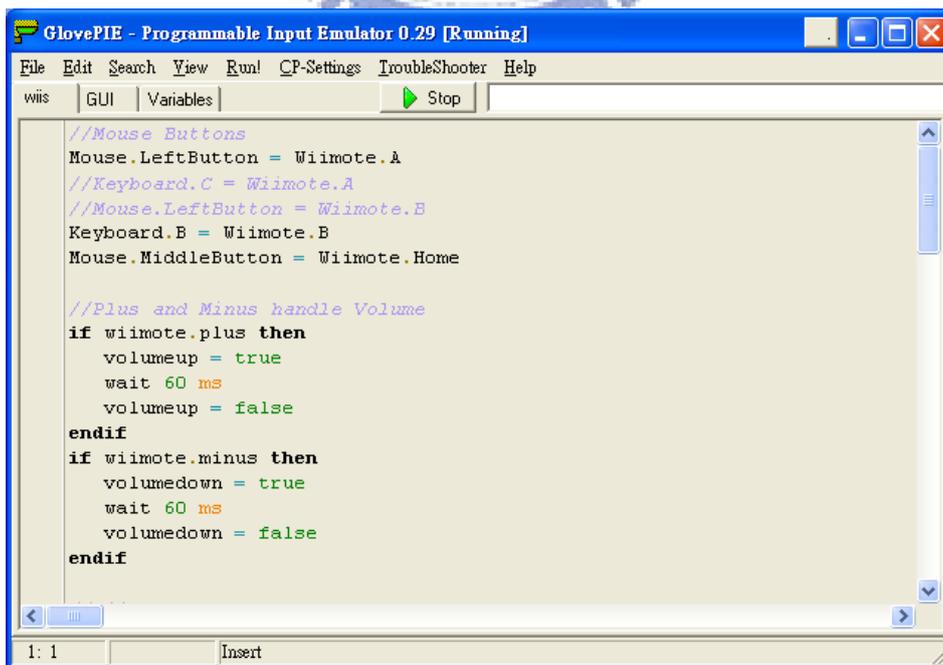


圖 24 glove pie communication script



圖 25 系統環境架構

## 4.2 實驗過程

實驗一開始，受測員對於使用 wii remote 來當作操控器感到相當的不習慣，但曾經使用過 wii remote 的受測員在大概在 1 分鐘左右的時間就可以逐漸上手，並且可以較容易的點選和使用系統裡面的辨識和繪圖，但從未使用過 wii remote 的受測員，經過了大約 10 分鐘左右才熟悉這種輸入方式，而所有的受測員在使用 10 分鐘之後抖動的情形會變得比較嚴重，在點選 icon 方面也變得比較吃力，而整個實驗大概在 10 分鐘左右結束。

### 4.3 實驗結果與討論

受測員統一對這種新穎的輸入方式感到有趣，且是可以接受的，但在系統功能鍵的選取上感到困難，因為當按下 Wii remote 的按鍵來進行選取時，手會不自覺往按下的方向移動，游標因此就會產生些許的飄移，沒有辦法來做精準的動作。

不過在中、英和日文的辨識上成效不錯，正常的輸入情況下，幾乎都在前五名，但對於整體的歪斜，則會導致辨識結果不佳(圖 26 )。

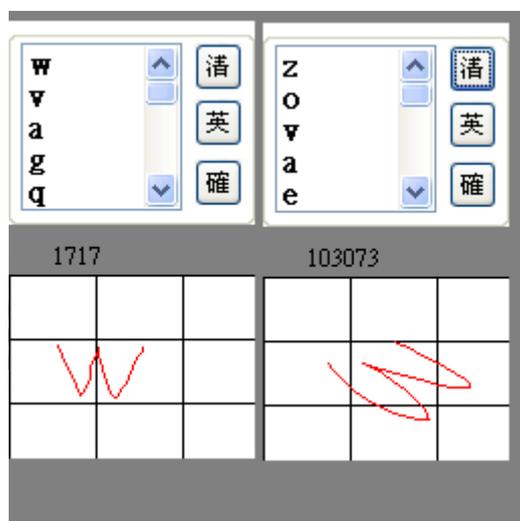


圖 26 左圖為正常的 W 右圖為歪斜的 W

圖形辨識上，成效比文字辨識稍差一些，準確率約莫 7 成，主要原因應該還是沒有一個定義好的字庫予以辨識，且輸入的方向多元化、轉折點的難以選取，都是造成圖形辨識不高的成因。

## 第五章 結論

### 5.1 研究總結

在本研究中，系統主要提出了兩個特點。第一、利用 Wii remote 來當作一個遠距離的輸入方式，第二、手寫文字與圖形的辨識。

系統結合了辨識系統和遠端的輸入，實做了一個可以在空中繪圖的方法，有別於其他類似 touchpad 或者傳統的鍵盤輸入，沒有了嚴謹的距離限制，提供較寬廣的使用環境，且在準確率上也相當的高，提供了一種特殊的輸入，使得使用者在輸入上是輕鬆且直覺的。

### 5.2 未來工作

本研究主要的方向是在於分析軌跡的方向，對於文字整體的旋轉和扭曲，並沒有額外做一個處理，導致這部分辨識率會下降，如果要提升辨識率，還需把旋轉後的軌跡做一個導正，這一部分，可以再更加深入的研究，取得旋轉的資訊，加強這一方面的辨識。



## 參考文獻

- [1] Wii remote 組成 <http://wii.gamer.com.tw/asksee.php>
- [2] 藍芽控制晶片 <http://www.broadcom.com/products/Bluetooth/Bluetooth-RF-Silicon-and-Software-Solutions/BCM2042>
- [3] United State Patent - US 6,947,029 B2 “Handwritten Data Input Device and Method, and Authentication Device and Method”
- [4] M. Nakai, N. Akira, H. Shimodaira, S. Sagayama. “Substroke Approach to HMM-based On-line Kanji Handwriting Recognition,” Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on 10-13 Sept. 2001 Page(s):491 - 495.
- [5] 陳奕安,「空中寫字-以攝影機在空中的移動軌跡作為輸入中文的裝置」, 國立交通大學, 碩士論文, 2007。
- [6] S. Tefvik Metin, S. Thomas, D. Randall “Sketch Based Interfaces: Early Processing for Sketch Understanding,” International Conference on Computer Graphics and Interactive Techniques archive ACM SIGGRAPH 2006 courses.
- [7] WiinRemote <http://onakasuita.org/wii/index-e.html>
- [8] GlovePie [http://carl.kenner.googlepages.com/glovepie\\_download](http://carl.kenner.googlepages.com/glovepie_download)