

逢甲大學

自動控制工程學系畢業專題報告

眨眼輸入系統

Blink Importing System

指導教授：林宸生

學 生：曾一峻

顏偵翔

賴效瑄

第一章 前言

1.1 研究背景及動機

近幾年來,隨著電腦效能與速度的提升,使得電腦的應用範圍不斷的擴大,從最早單純的執行計算功能到複雜化網路多媒體的呈現,常常有許多意想不到的技術上的突破,各種新的人機介面(Human-machine Interface)解決了許多以往操作上的不方便也解決了人們在生活上的許多問題,並且在控制領域上扮演了一個極重要的角色.在日常生活中,我們常常因為某些因素而覺得雙手不夠使用而需要一種新的技術來克服這些問題,因此吸引了一些研究人員的重視,進而想到利用人們本身具有的視覺或聲音來代替傳統的雙手控制,進而解決以上所產生的問題,這些研究的最終目的均希望開發出操作簡便,實用性高的人機介面(Human-machine Interface),使我們能以視覺的變化來下達指令,如此一來不但能減少雙手操作的負擔,更能大大的提高工作效率改善生活品質

在追瞳系統實際的應用方面,可用來幫助四肢癱瘓或半身不遂的殘障人士,藉由追瞳系統來電腦下達指令;或用於急診室內,利用特殊的字母板,配合電腦記錄病患注視的字母並送至螢幕或利用語音系統加以發音,幫助無法言語的重傷患者表達識及與醫生進行溝通;在影像處理方面則是利用追瞳系統不斷追蹤眼球位置並配合其他的位置感測器,計算目前使用者眼前應看到的

何種畫面

由本系主導研發的[光學式瞳位追蹤器],屬於非接觸型的追瞳系統,只需對由 CCD 擷取到的眼睛影像作影像樣本分析,即能達到很高的瞳位辨識率,且對眼睛無任何接觸性的傷害及不適感;在一次指導老師的帶領下,實際見到了患者操作追瞳器的經驗,患者對此套系統贊譽有加也認為如果能加強它的附屬應用功能並簡化系統操作程序,對於患者將有很大的助益,老師也說這套系統的語音輸入法方面,如果能在輸入速度上加以改善並能使患者發出更多的詞彙,而不是僅是面板上的字母,相信對於病患的溝通方面將能有更大的助益因此老師談到病友曾秋路的案例,所謂的運動神經元疾病患者發病到了末期,四肢都無法動彈連說話的能力也被剝奪了,只剩下眼睛的活動還能控制而已,在此曾先生的家人曾經利用所謂的注音符號字母盤也是利用病友眨眼的次數來表達出病人想要的意思,但這項工作照顧的人要聚精會神的注視著病人的眼部動作可說是十分的費神也太不方便了,因此,我們便想到利用電腦來完成這項步驟,這樣不但會比較方便看護的人也會比較輕鬆

因此為了改良追瞳器在發音系統的不方便而製作的,利用完整的注音拼盤(如圖一)節省患者在使用先前系統須換頁尋找想要的注音符號的時間,另外,對於某部分的患者對於眼球的控制並不十分的靈活,因此利用眨眼的動作

來進行發聲的判別,而不再是單調的從面版上選擇接近自己需求的字母,使患者能更方便的發出自己的需求或想法,表達出更豐富的情感出來

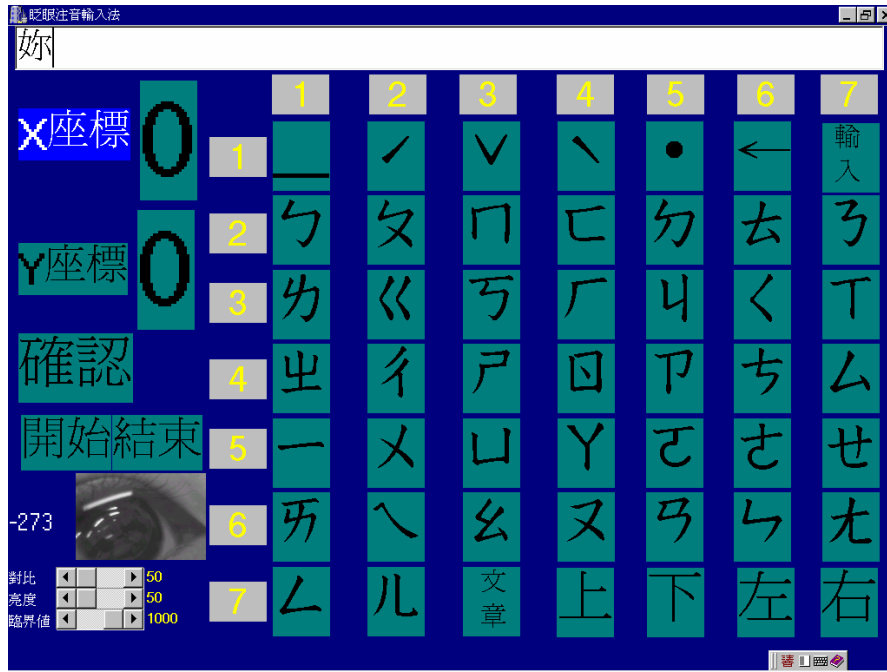


圖 1.1 注音面盤

1.2 研究方法:

本專題報告所要研究的方向共分為兩部分:

1. 注音盤的建立
2. 眨眼系統的確立
3. 語音輔助系統的建立

在第一項的研究中,我們採用了座標軸的系統,利用座標平面上每個座標訊號,皆代表每一個獨立位置的特性,附與每個座標獨立的代表的拼音字母,再由使用者輸入的座標訊號,尋找其所代表的字母,經由反覆的輸入便能進行拼音的工作,雖然要不斷的進行座標訊號輸入的動作,但其所能發出的詞彙卻將不只限於固定的幾句而已,更能將使用者所想要表達的想法說出而不是只是要操作者在眾多句子中找出與自己內心想法相近的句子而已

在第二項的研究中,我們則利用了張眼與閉眼時,所呈現的亮暗程度的不同來判斷是否操作者有進行眨眼的動作,我們在 CCD 所擷取的影像(如圖二)螢幕上建立以瞳孔為準軸上下延伸的一條基準帶,當機準帶上亮點的平均總數小於一固定值時,就由系統自行判斷為眨眼的動作,同時為了避免為無意識的眨眼動作,則利用數個 Timer 來進行判斷是否為有意識眨眼或無意識的眨眼動作;此判斷是必要的,有時受測者會有無意識的眨眼動作,這乃是人體為了避免眼睛過度疲勞的一種自然生理現象,所以我們必須根據眨眼的特性,把有意識與無意識的動作區分開來,因為無意識眨眼對系統沒有幫助,反而是一種負擔,因為系統需要的是有意識眨眼的動作,為了判別有意識眨眼和無意識眨眼,分辨無意識和有意識眨眼的方法,我們則利用兩者閉眼的時間差來進行判斷。

圖 1.2 CCD 擷取的眼部特寫

在第三項研究中,我們利用電腦內建的發音系統來發出利用眨眼輸入完成的字母或句子,另外在輔助操作方面,在計數眨眼的動作方面,利用電腦不斷的將患者眨眼的計算次數利用聲霸卡發音出來,告知其已眨眼的正確次數,那麼患者只要在一開始確定住正確的座標後便不用一直盯著螢幕看了,如此也可以避免造成眼睛過度疲勞

而當操作者經過數次的操作後如果已經將注音拼盤上各各座標所對應的拼音字母牢記後,則可以利用進階的模式來進行輸入的動作,操作者只要在聽到想要的 X 軸座標或 Y 軸座標時張眼就可拼出想要的音符,如此不但能更快速的進行輸入的動作同時對於眼睛更不會造成疲勞

第二章 理論基礎

2.1 二值化基礎理論

大家都知道，我們眼睛所看的顏色都是由紅 (Red)、綠(Green)、藍(Blue)

三原色所組成，例如：

	Red	Green	Blue
紅色	255	0	0
綠色	0	255	0
藍色	0	0	255
白色	255	255	255
黑色	0	0	0

表 2.1

本系統採用 FOCUS Enhancements 公司之 InVideo/PCI 影像擷取卡。當擷取卡驅動 CCD 攝影機擷取到眼球運動的影像之後，便將影像資料傳輸至電腦並儲存在固定的位址。因為所擷取的眼球影像為 8bit 之影像，所以影像中的每一個圖素(pixel)皆由三組亮度分佈在 0~255 之間的紅(Red)、綠(Green)、藍(Blue)影像所組成。

2.2 影像二值化

當我們在進行影像處理時，若被處理的物體和不需要的背景能加以分離的話，將使我們方便許多，並更快處理速度。一張影像是由像素點所構成的二圍矩陣，每一個像素的光強度函數（ Light-Intensity Function ）以 $f(x, y)$ 表示之，俗稱為灰度值（ Gray Level ）或明亮度（ Brightness ），其中 (x, y) 為影像中的座標。一般灰度值的範圍如下：

$$0 \leq f(x, y) \leq 255$$

其中 0 表示最暗或黑色，而 255 表示最亮或白色。如圖 2.4 所示的灰度分佈統計圖，為影像中的像素對應於 $f(x, y)$ 所產生的圖形，整幅影像就由亮的背景和暗的物體所組成，在兩個模式間存在一個臨界點 T （ 閾值 ），若我們想從背景中提出一個物體，即可以 T 為判斷值，如果 $f(x, y) < T$ 則稱為物體點，否則為背景點。在本次專題實驗中，我們將臨界的 T 值設為可調，以調整出最佳的分離狀況。

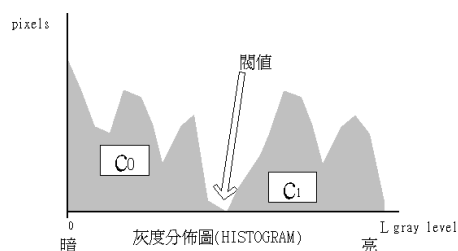


圖 2.1 灰度分佈統計圖

一個二值化的影像 $g(x, y)$ 定義為

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & f(x, y) \leq T \end{cases}$$

影像中的像素若被視為 1 時即被視為背景，標記為 0 則被視為物體，因此在我們的實驗中，瞳孔的影像經二值化處理後就會清楚的與背景分離，以便演算法則判斷。

2.3 人工光源選擇

為了使 CCD 攝影機能清楚的抓取眼睛影像，良好的光源選擇非常的重要。要成為良好的光源需符合下列幾項特點：

1. 獲得明顯的對比（即反差，Contrast），以凸顯其輪廓。
2. 顯露出影像的細節。
3. 強化（Enhance）物體的特徵（Feature）。
4. 增加訊噪比（ S/N ，Signal Noise Ratio），以確保影像資料的真實性。

我們所使用的光源若符合以上的特點越多，則得到的效果越好。而在選擇光源的另一重點，則是需瞭解光源的特性，針對不同環境採用不同的光源，以得到較佳性能及降低對使用者的傷害。以下列舉幾項光源的相關特性：

1. 光的亮度。
2. 亮度的穩定性。
3. 光的頻譜。
4. 是否有螢光。
5. 是否有紫外線。
6. 是否有紅外線。
7. 使用壽命。
8. 是否易於安裝與固定。
9. 色溫（Color Temperature）。
10. 演色性（Color Rendering）

光源體接受電能後，除轉換一部份可見光之外，其餘電能則轉變為紅外線、紫外線、螢光及熱能等等，這些因素不僅會影響到光源本身的壽命，並對與光源有直接接觸的使用者造成若干程度的傷害。以下列舉數種常見的人工光源，列表匯總與比較。

表 2.2 各類人工光源匯總與比較。

	普通燈	鹵素燈	螢光燈	水銀燈	氙氣燈
紅外線	極多	極多	無	少	多
紫外線	無	無	許多	少許	少許
螢光	無	無	少許	少許	少許
發光效率 (lm/W)	22	27	50~104	28~63	30
平均壽命 (小時)	1000	2000	5000	12000	—
機器視覺適用性	佳	佳	佳	劣	佳

由以上總表可看出各類光源的特性，本實驗除了機械視覺的適用性需為佳，也就是可使得 CCD 能更清楚的攝取影像之外，尚需考慮對使用者眼睛的傷害程度，譬如氙氣燈的發光頻率包含紫外線的範圍，直視過量會造成眼睛的傷害；而有些螢光燈（日光燈）的頻譜分佈並不符合 CCD 攝影機取像頻率的要求，所以也不適合採用。而一般常見的普通燈泡其輻射紅外光強，亮度穩定性高，不會有閃爍現象，且具有連續光譜、價格便宜、容易取得、不受環境溫度影響等優點，故我們採用普通燈泡為本實驗的人工光源，經實驗證實，確實有不錯的效果。

第三章 系統架構與流程

本次專題研究報告，共分系統架構（其包含『軟體—程式』與『硬體—CCD攝影機』）與系統流程兩部分，我們將在後面分別討論，並最後將其整合。

3.1 系統架構

3.1.1 實驗設備：

1. 影像處理用個人電腦：

PentiumII 350 電腦，64MB RAM，S3 Trio 64V+

2. 音效卡：

16bit 天音卡

3. 影像擷取卡：

ComputerEyes/PCI Video Capture

4. 影像擷取攝影機：

CCD 攝影機一台

5. 語音設備：

喇叭兩個

6. 影像處理使用的軟體：

Borland C++ Builder 4

3.1.2 軟體程式:

(1) 注音盤面板部分:

黃線框起部分為注音盤面板



圖 3.1 注音盤面板部分

注音盤程式部分:

A. 主程式部分:

```
void keyyd(Byte vk,Byte hk)
{
    keybd_event(vk,hk,NULL,0); //vk--虛擬鍵值碼
} // hk--硬體掃描碼

void keyyu(Byte vk,Byte hk)
{
    keybd_event(vk,hk,KEYEVENTF_KEYUP,0);
}

void keyy(Byte vk,Byte hk)
```

```

{
    keyyd(vk,hk);
    keyyu(vk,hk);
}

```

B.注音 BUTTON 部分:

例:”ㄅ”Button:

```

void __fastcall TKeyForm::Label1Click(TObject *Sender)
{
    za='1';//鍵盤上按鍵
    zb=0x02;//鍵值碼
    keyy(za,zb);
}

```

剩下的按鍵就只改變按鍵與其鍵值碼部份，便可類推之後的所有 Button 程式碼寫法。

(2) 座標計數部分:黃線框起為座標部分

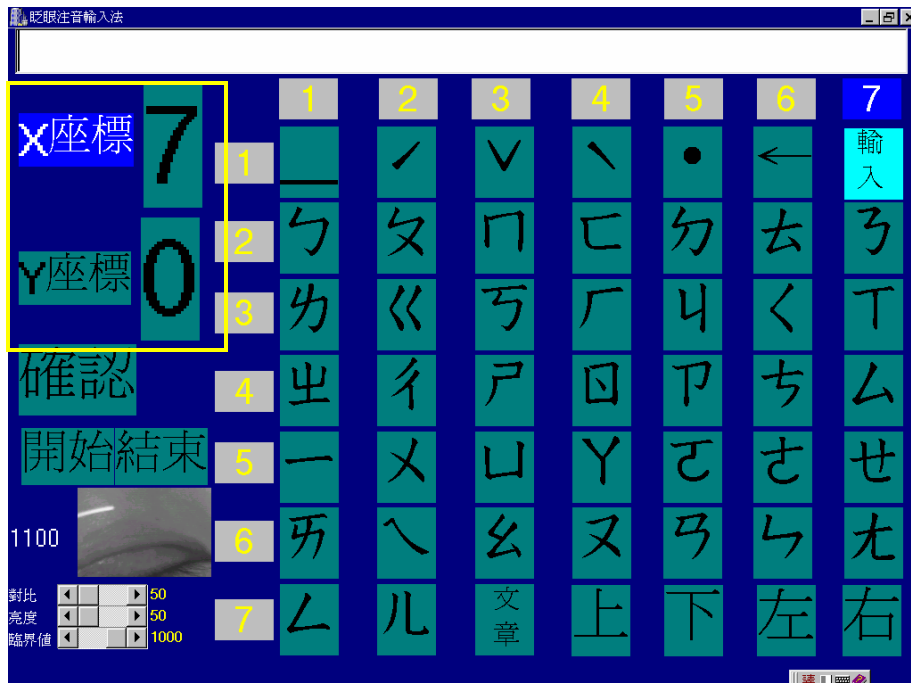


圖 3.2 黃線框起為座標

A.X 座標程式部分:

```

void __fastcall TForm2::Label45Click(TObject *Sender)

```

```

{
x=x++;//每按一次加 1
if(x>7)//假如大於 7 則跳回 1 再循環
{x=x-7;}
Label47->Caption=x;//顯示 X 座標
if(x==1)
{
sndPlaySound("1.wav",SND_SYNC);//加上語音部分
}
if(x==2)
{
sndPlaySound("2.wav",SND_SYNC);
}
if(x==3)
{
sndPlaySound("3.wav",SND_SYNC);
}
if(x==4)
{
sndPlaySound("4.wav",SND_SYNC);
}
if(x==5)
{
sndPlaySound("5.wav",SND_SYNC);
}
if(x==6)
{
sndPlaySound("6.wav",SND_SYNC);
}
if(x==7)
{
sndPlaySound("7.wav",SND_SYNC);
}
}

```

B.Y 座標程式部分:

```
void __fastcall TForm2::Label46Click(TObject *Sender)
```

```

{
y=y++;//每按一次加 1
if(y>7) //假如大於 7 則跳回 1 再循環
{y=y-7;}
Label48->Caption=y; //顯示 Y 座標
if(y==1)
{
sndPlaySound("1.wav",SND_SYNC);//加上語音部分
}
if(y==2)
{
sndPlaySound("2.wav",SND_SYNC);
}
if(y==3)
{
sndPlaySound("3.wav",SND_SYNC);
}
if(y==4)
{
sndPlaySound("4.wav",SND_SYNC);
}
if(y==5)
{
sndPlaySound("5.wav",SND_SYNC);
}
if(y==6)
{
sndPlaySound("6.wav",SND_SYNC);
}
if(y==7)
{
sndPlaySound("7.wav",SND_SYNC);
}
}
}

```

(3) 確認鈕 Label: 黃線框起為確認鈕

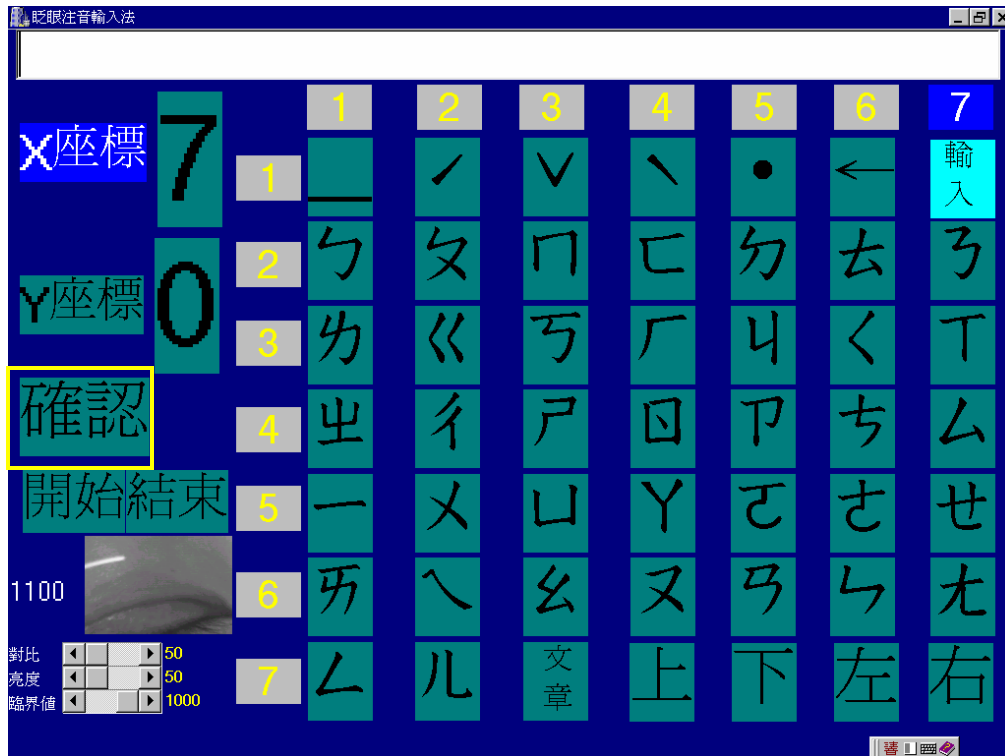


圖 3.3 黃線框起為確認鈕

```
void __fastcall TForm2::Label49Click(TObject *Sender)
```

```
{  
    int a;  
    a=7*(y-1)+x;//計算注音盤面版之座標  
    x=0;y=0;  
    switch(a){//執行面板內 Label 之程式  
    case 1: Label1Click(Sender);  
            break;  
    case 2: Label2Click(Sender);  
            break;  
    case 3: Label3Click(Sender);  
            break;  
    case 4: Label4Click(Sender);  
            break;  
    case 5: Label5Click(Sender);  
            break;  
    case 6: Label6Click(Sender);
```



```
        break;
case 7: Label7Click(Sender);
        break;
case 8: Label8Click(Sender);
        break;
case 9: Label9Click(Sender);
        break;
case 10: Label10Click(Sender);
        break;
case 11: Label11Click(Sender);
        break;
case 12: Label12Click(Sender);
        break;
case 13: Label13Click(Sender);
        break;
case 14: Label14Click(Sender);
        break;
case 15: Label15Click(Sender);
        break;
case 16: Label16Click(Sender);
        break;
case 17: Label17Click(Sender);
        break;
case 18: Label18Click(Sender);
        break;
case 19: Label19Click(Sender);
        break;
case 20: Label20Click(Sender);
        break;
case 21: Label21Click(Sender);
        break;
case 22: Label22Click(Sender);
        break;
case 23: Label23Click(Sender);
        break;
case 24: Label24Click(Sender);
        break;
case 25: Label25Click(Sender);
```

```
        break;
case 26: Label26Click(Sender);
        break;
case 27: Label27Click(Sender);
        break;
case 28: Label28Click(Sender);
        break;
case 29: Label29Click(Sender);
        break;
case 30: Label30Click(Sender);
        break;
case 31: Label31Click(Sender);
        break;
case 32: Label32Click(Sender);
        break;
case 33: Label33Click(Sender);
        break;
case 34: Label34Click(Sender);
        break;
case 35: Label35Click(Sender);
        break;
case 36: Label36Click(Sender);
        break;
case 37: Label37Click(Sender);
        break;
case 38: Label38Click(Sender);
        break;
case 39: Label39Click(Sender);
        break;
case 40: Label40Click(Sender);
        break;
case 41: Label41Click(Sender);
        break;
case 42: Label42Click(Sender);
        break;
case 43: Label43Click(Sender);
        break;
case 44: Label44Click(Sender);
```

```

        break;
case 45: Label69Click(Sender);
        break;
case 46: Label70Click(Sender);
        break;
case 47: Label71Click(Sender);
        break;
case 48: Label72Click(Sender);
        break;
case 49: Label73Click(Sender);
        break;
    }
}

```

(4) 影像二值化及計算範圍

```

void __fastcall TForm2::Timer4Timer(TObject *Sender)
{
    long a;
    int i,j,k,m;
    for(i=74;i<75;i++){//計算 Ceyes 框內中間線之總點數
    for(j=0;j<100;j++){
    TCColor c=Ceyes1->Canvas->Pixels[i][j];//顯示 Pixels 之色素值
    k=(int)c.Red+(int)c.Green+(int)c.Blue;//將色素值轉換成整數
    a=a+k;
    }}
    m=a/100;//將總 Pixels 色素值做平均
    if(g==1)
    {z=abs(m);}//將平均值取絕對值丟到 z
    Label50->Caption=m;
    m=0;
}

```

(5) 眨眼判斷方法

1. Timer1 啟動，0.3 秒後執行

```

void __fastcall TForm2::Timer1Timer(TObject *Sender)

```

```

{
if(z>t-20&& z<t+40)//當 z 值落在某一範圍，為閉眼
{Timer2->Enabled=true;}//Timer2 啓動
if (z<t-30) //當 z 值小於某一範圍，為開眼
{Label50Click(this);}//按下 Label50，將 Timer1 關掉
}
}
//-----

```

2.Timer2 啓動，0.3 秒後執行

```
void __fastcall TForm2::Timer2Timer(TObject *Sender)
```

```

{
Timer1->Enabled=false;
if(Form1->p==1){//進階模式
if(w==0){//x 座標
if(z>t-20&& z<t+40)//當 z 值落在某一範圍
{Label45Click(this);} //按下 x 座標計數
if(w==1){//y 座標
if(z>t-20&& z<t+40) //當 z 值落在某一範圍
{Label46Click(this);} } //按下 y 座標計數
if(Form1->p==0){//初階模式
if(w==0){ //x 座標
if(z>t-20&& z<t+40) //當 z 值落在某一範圍
{Timer3->Enabled=true;}//Timer 3 啓動
if(z<t-30)//當 z 值小於某一範圍
{Label45Click(this);} // 按下 x 座標計數
if(w==1){ //y 座標
if(z>t-20&& z<t+40) //當 z 值落在某一範圍
{if(y>0)//當 y 座標大於 0
{Timer5->Enabled=true;} } //Timer 5 啓動
if(z<t-30) //當 z 值小於某一範圍
{Label46Click(this);} } //按下 y 座標計數
}
}
}
}
//-----

```

3.在初階模式，Timer 3 啓動，2.4 秒後執行

```
void __fastcall TFrm2::Timer3Timer(TObject *Sender)
```

```

{
Timer1->Enabled=false;
Timer2->Enabled=false;
if(w==0){//初階模式
if(x>0){//當 x 座標大於 0
if (z>t-20&&z<t+40)//當 z 值落在某一範圍
{w=1;//y 座標
sndPlaySound("Ding.wav",SND_SYNC);//到 y 座標有 Ding 聲
Label45->Color=clTeal;
Label45->Font->Color=clBlack;
Label46->Color=clBlue;
Label46->Font->Color=clWhite;}
if (z<t-30) //當 z 值小於某一範圍
{Label50Click(this); }
}
}
}

```

4. 當影像擷取時，所執行的程式

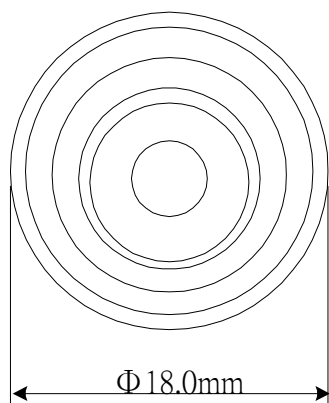
```

void __fastcall TForm2::Ceyes1ReceiveImage(TObject *Sender)
{
if(Form1->p==1)//進階模式
{ if(z>t-20&&z<t+40)//判斷 z 值在這範圍
{Timer1->Enabled=true;//Timer1 啓動
}
if(x>0)//當 x 座標大於 0
{
if(z<t-30)//當 z 值小於某一範圍
{
w=1;
Label45->Color=clTeal;
Label45->Font->Color=clBlack;
Label46->Color=clBlue;
Label46->Font->Color=clWhite;
}
}
if(y>0)//當 y 座標大於 0

```

```
{
    if(z<t-30)
    {
        Label49Click(this);
    }
}
}
if(Form1->p==0)//初階模式
{
if(z>t-20&& z<t+40) //當 z 值落在某一範圍
{Timer1->Enabled=true;}}//Timer1 啓動
}
```

3.1.3 硬體 CCD 攝影機架構



Model G-3.8

圖3.4 CCD正視圖

	Lens
Specification of lens	G-3.8
Type	Glass
Focal length	3.8mm
F Point	F 2.0
Angular view (H*V) ^o	51*40 ^o
Resolution (Horizational)	Over 380 TV lines (center)
Minimum Illumination	0.8 Lux

表3.1

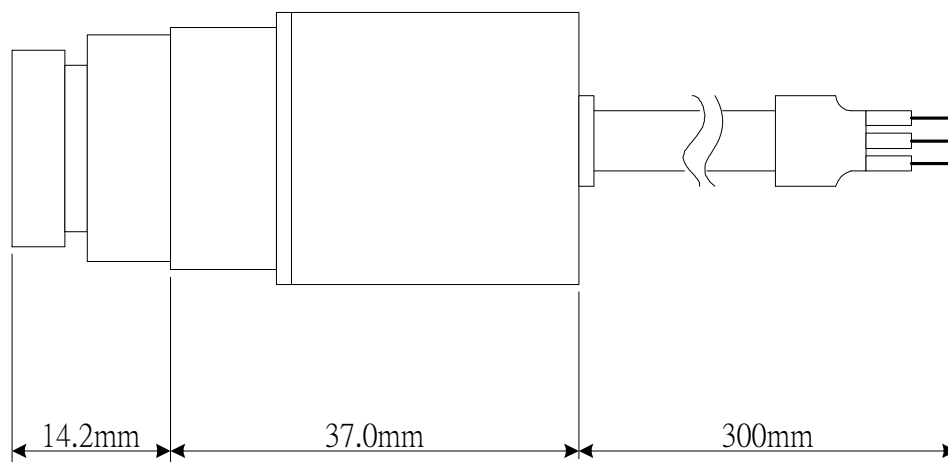


圖3.5 CCD側視圖

3.1.4 CCD 配戴模式

(1) 小型 CCD

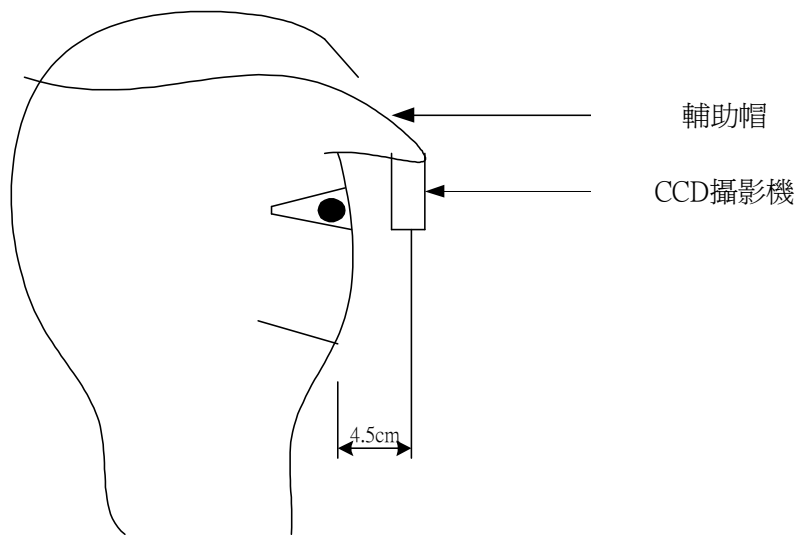


圖 3.6

(2) 線型 CCD

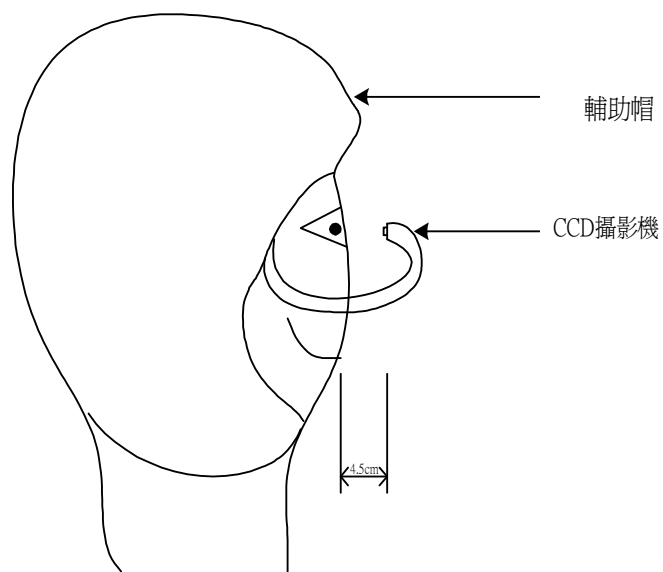


圖 3.7

3.1.5 CCD 攝影機優點

CCD 攝影機與其他傳統取像裝置相比，的確有其優異之處，以下簡述其優點：

1. 因為沒有真空管，所以耐衝擊及抗震性強。
2. 體積小，重量輕。
3. 產品壽命長、不易老化、耗電低、穩定性佳。
4. 因屬 IC 製造技術，其幾何精密度好，影像不會扭曲變形。
5. 因為沒有偏向磁場，所以不受電磁干擾。
6. 每一次取像終了，電荷一定移出，故沒有殘像的問題。

如此的考量，在未來將整套追瞳系統模組化時的效能評估及誤差率的影響都是相當的重要的。

3.2 系統流程

3.2.1 程式判斷流程

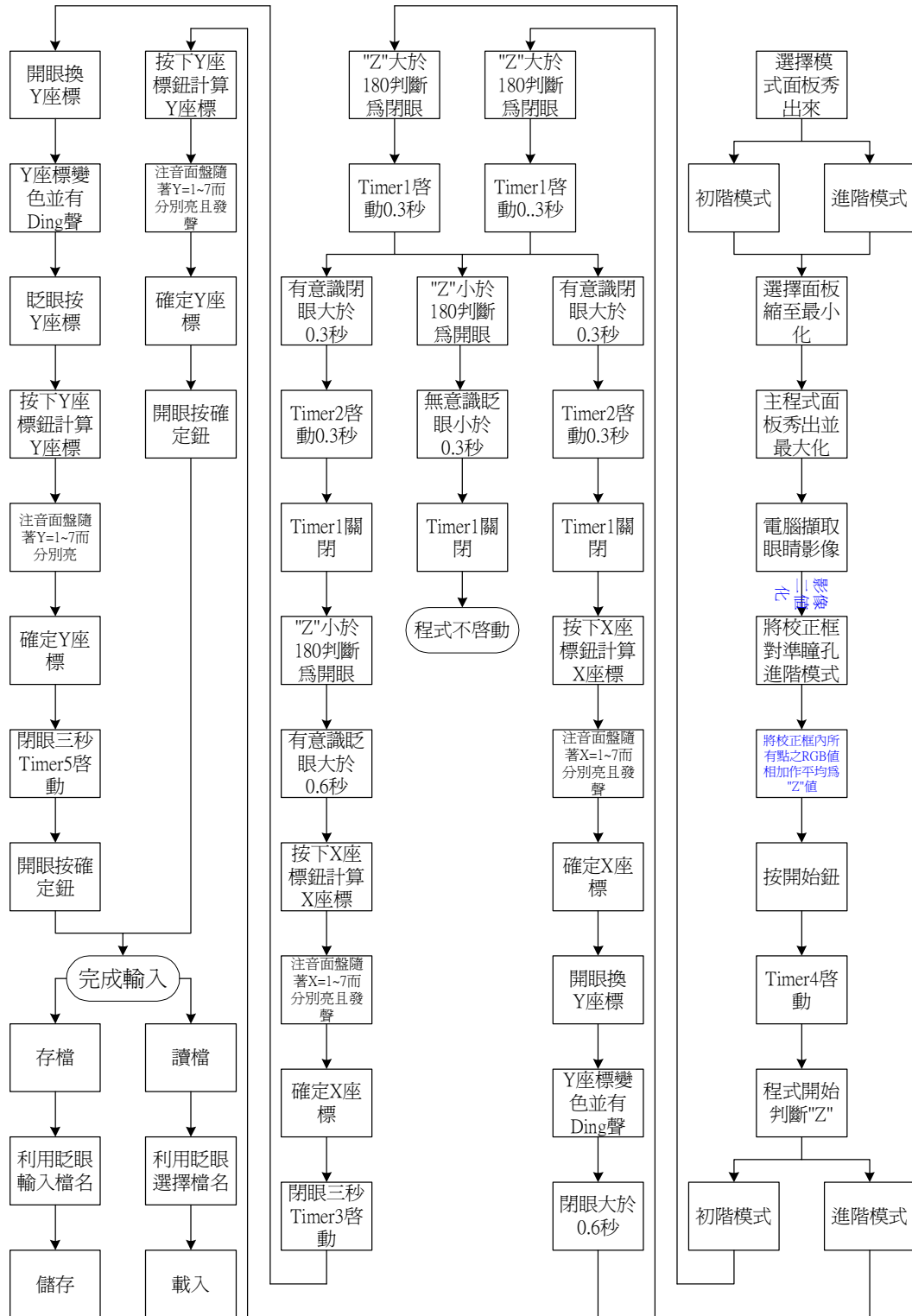


圖 3.8

3.2.2 程式執行流程

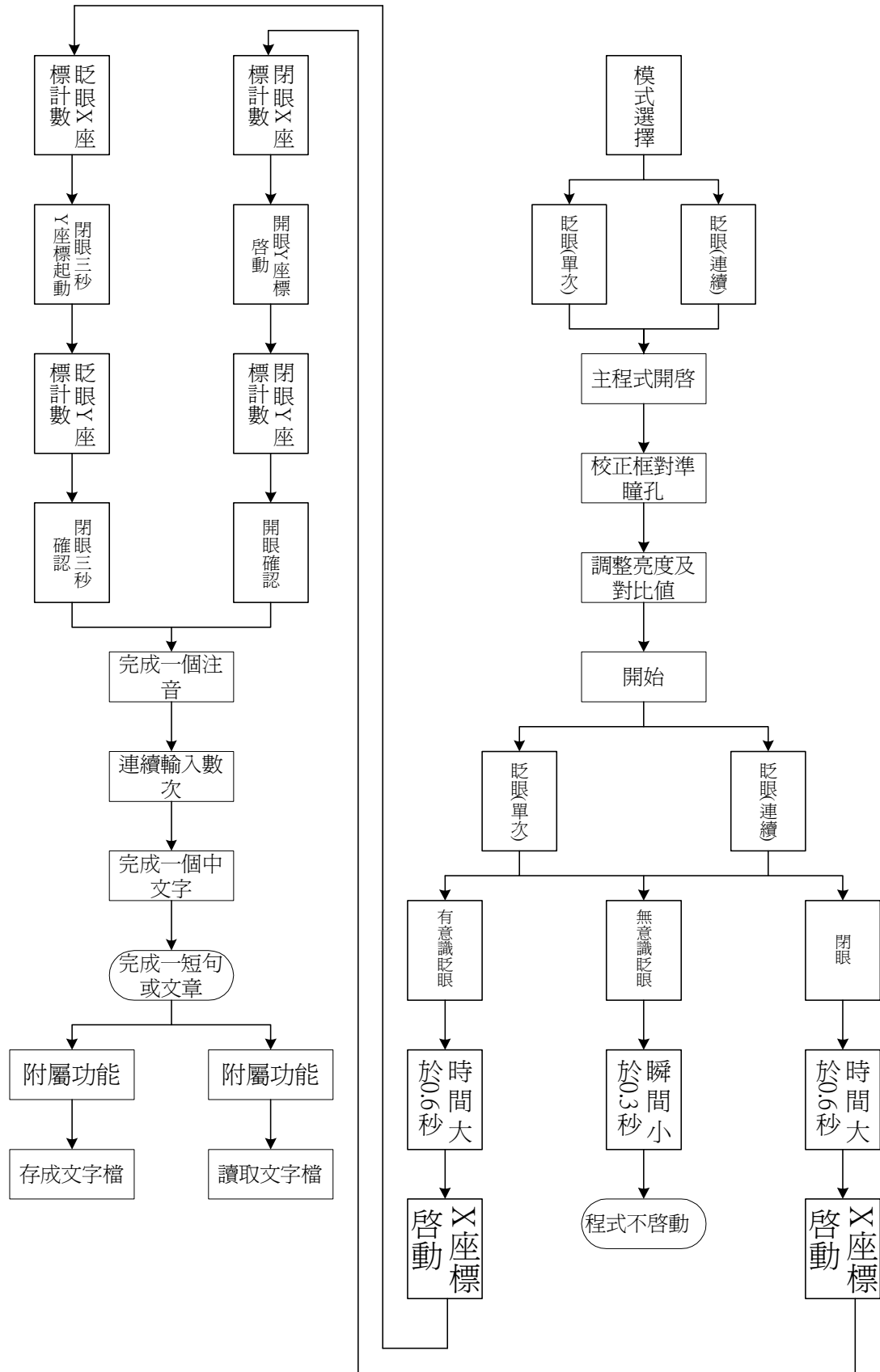


圖 3.9

第四章 輔具使用說明

1. 輔具模式選擇：

- 一. 初階(圖 4.1)模式:此為第一種眨眼法，主要是練習使用輔具及讓病友(漸凍人)可訓練眼部肌肉之用。



圖 4.1 初階模式

- 二. 進階(圖 4.2)模式:此為第二種眨眼法，主要是讓習慣使用輔具之病友，可免除連續眨眼之不便，讓輸入可以更為順暢及快速。



圖 4.2 進階模式

2. 進階輸入模式:以輸入”妳”作範例

一.”ㄋ”X座標為7，先持續閉眼，此時將聽到電腦發出座標聲，聽到7時開

眼(圖 4.3)。

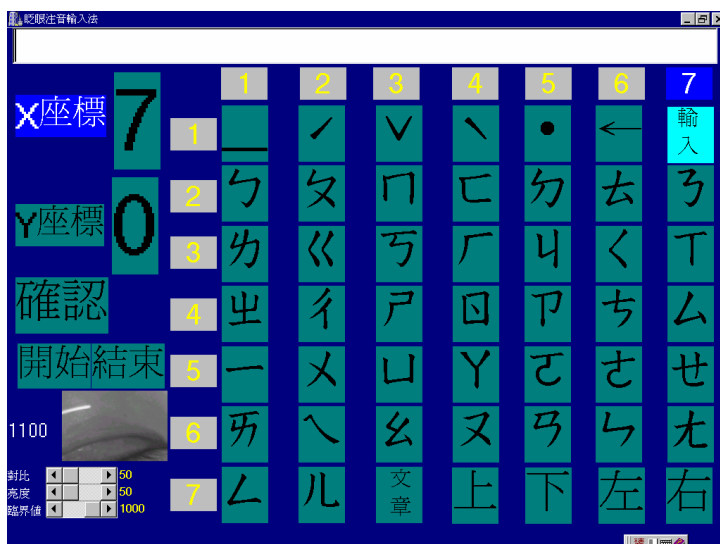


圖 4.3 ”ㄋ”X座標為7

二.”ㄋ”Y座標為2，先持續閉眼，此時將聽到電腦發出座標聲，聽到2時開

眼(圖 4.4)。



圖 4.4 ”ㄋ”Y座標為2

三.”一”X座標為1，先持續閉眼，此時將聽到電腦發出座標聲，聽到1時開眼(圖 4.5)。

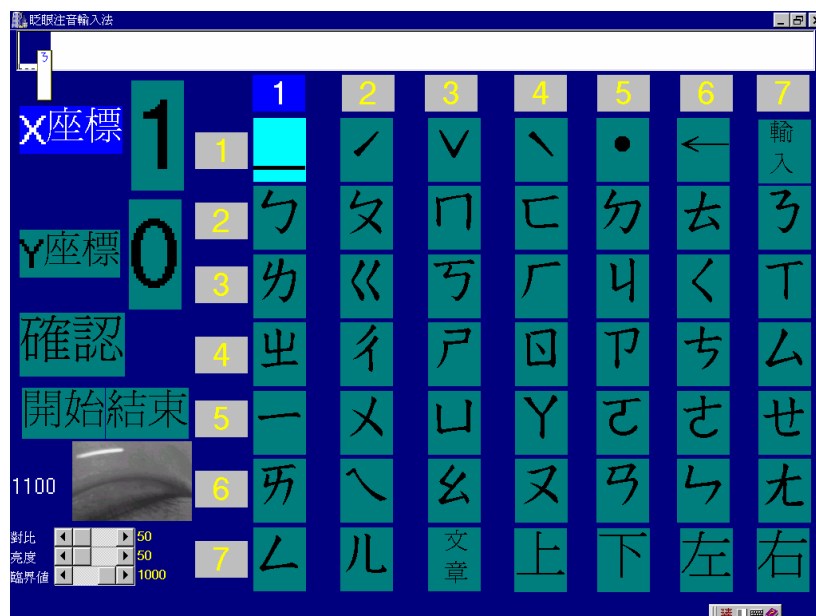


圖 4.5 ”一”Y座標為1

四.”一”Y座標為5，先持續閉眼，此時將聽到電腦發出座標聲，聽到5時開眼(圖 4.6)。

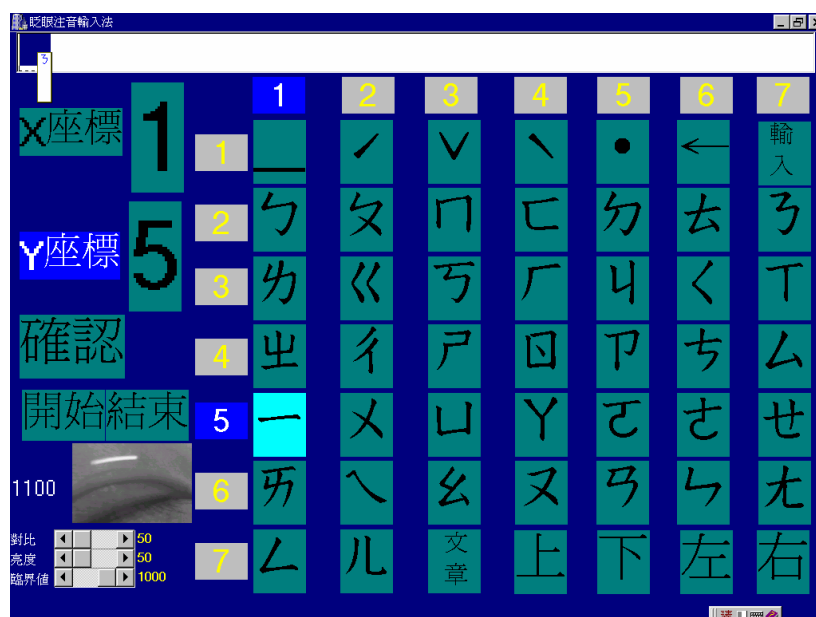


圖 4.6 ”一”Y座標為5

五.”v”X座標為3，先持續閉眼，此時將聽到電腦發出座標聲，聽到1時開

眼(圖 4.7)。



圖 4.7 ”v”X座標為3

六.”v”Y座標為1，先持續閉眼，此時將聽到電腦發出座標聲，聽到1時開

眼(圖 4.8)。

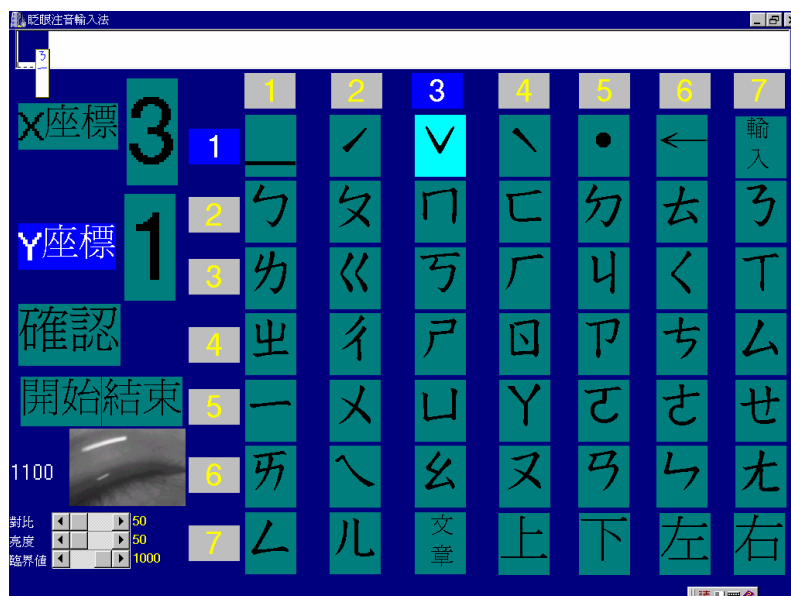


圖 4.8 ”v”Y座標為1

七. 完成”妳”注音輸入，但還未完成還需按下”輸入(ENTER)””，讓字完成文字編

輯(圖 4.9)。



圖 4.9 完成”妳”注音輸入

八.”輸入(ENTER)”X 座標為 7，先持續閉眼，此時將聽到電腦發出座標聲，聽

到 7 時開眼(圖 4.10)。

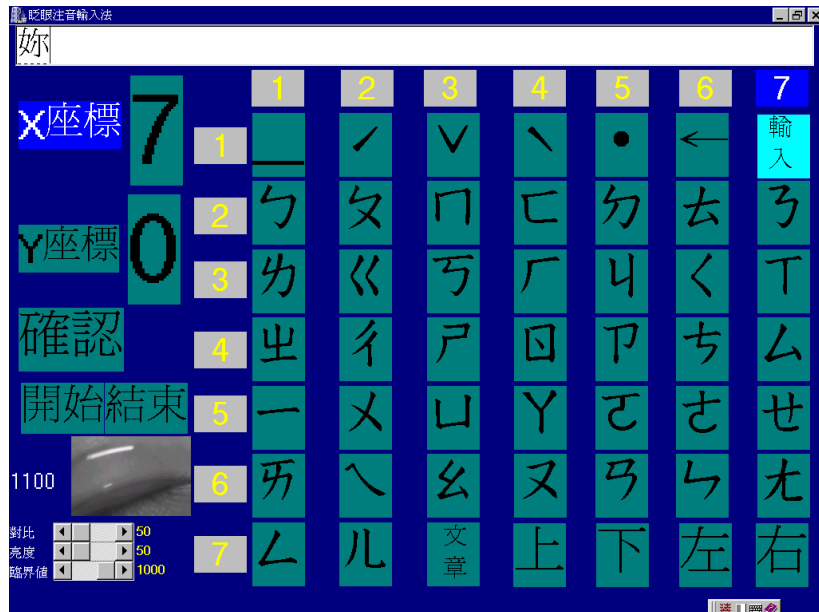


圖 4.10”輸入(ENTER)”X 座標為 7

九.”輸入(ENTER)”Y座標為1，先持續閉眼，此時將聽到電腦發出座標聲，聽到1時開眼(圖4.11)。



圖 4.11 ”輸入(ENTER)”Y座標為1

十. 完成”妳”注音輸入及文字編輯圖4.12。

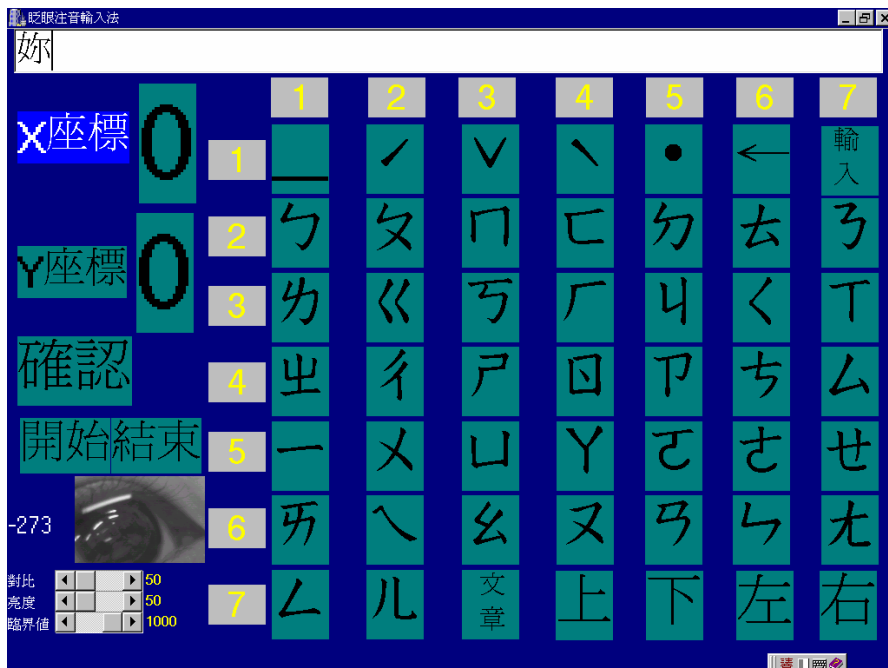


圖 4.12 完成”妳”注音輸入及文字編輯

3. 完成文章編輯後之存讀檔模式：經之前熟悉運用輔具編輯文字後，病友可以把想打的文章編輯後存檔。

一. ”文章”座標(3, 7)，將文字編輯成文章後，病友要存讀檔，可選擇文章模式(圖 4.13)。



圖 4.13 ”文章”座標(3, 7)

二. 選擇”文章”模式後，會出現”請輸入文章檔名:”，及”存檔”、”讀檔”，兩個選項(圖 4.14)。



圖 4.14 選擇”文章”模式後之情形

三. 在”請輸入文章檔名:”需輸入存檔或讀檔文章的檔名，在此用”一”字做檔

名，用之前的輸入法，所以”一”座標(1,5)。(圖 4.15)

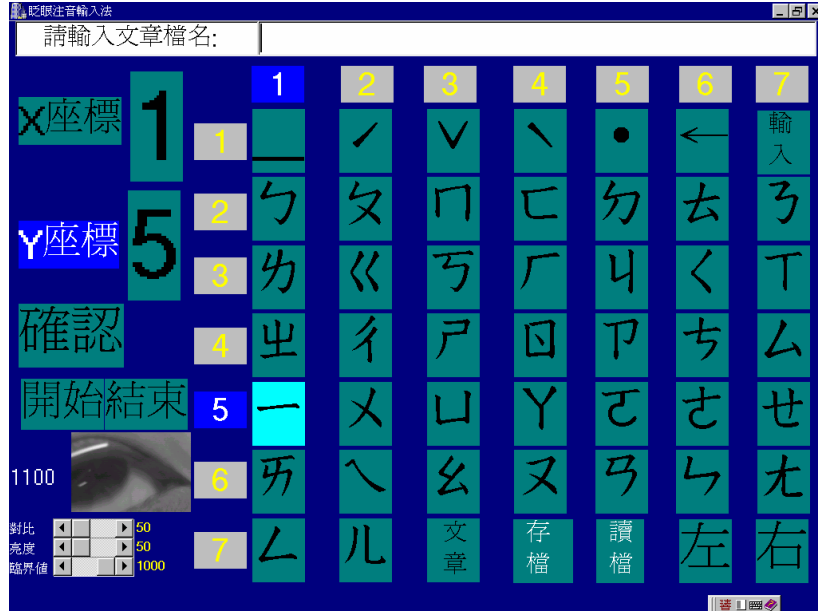


圖 4.15”請輸入文章檔名:”輸入”一”字

四. 在”請輸入文章檔名:”需輸入存檔或讀檔文章的檔名，在此用”一”字做檔

名，用之前的輸入法，所以”一聲”座標(1,1)。(圖 4.16)

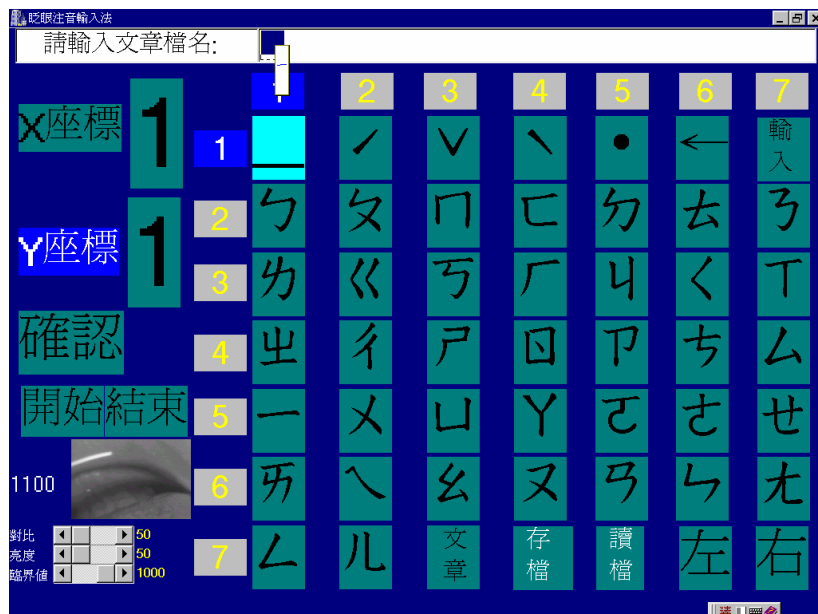


圖 4.16”請輸入文章檔名:”輸入”一”字

五. 需按下”輸入”鍵才能完成輸入所要存檔或讀檔檔案名稱(圖 4.17)。

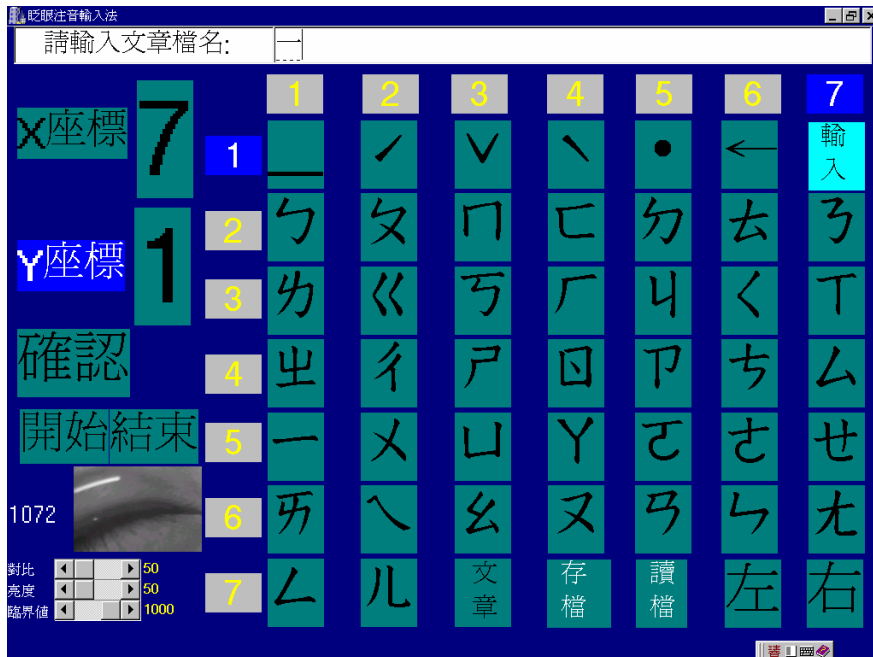


圖 4.17 按下”輸入”鍵完成檔名

6.1 選擇”存檔”按下，將完成之文章完成存檔(圖 4.18)。



圖 4.18 選擇”存檔”，將文章儲存

6.2: 按下”存檔”鈕，跑出存檔對話框，此時電腦會自行儲存(圖 4.19)。



圖 4.19 按下存檔鈕，跑出”存檔對話框”

6.3: 若選擇”讀檔”按下，將完成之文章完成讀檔。(圖 4.20)



圖 4.20 選擇”讀檔”，將之前打的文章載入

6.4: 按下”讀檔”鈕，跑出讀檔對話框，此使電腦會自行讀檔(圖 4.21)。



圖 4.21 按下讀檔鈕，跑出”讀檔對話框”

6.5: 完成”讀檔”，將之前所編輯的文章讀取。(圖 4.22)

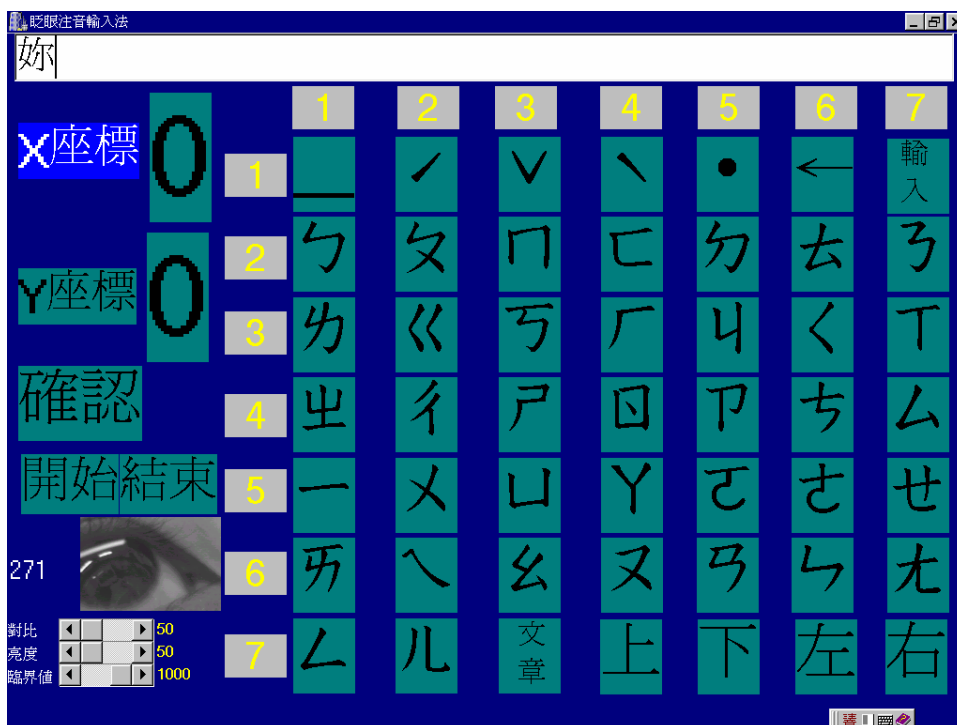


圖 4.22 完成”讀檔”，將之前所編輯的文章讀取

參考文獻

- [1]李勁、鄭浩，”精通 C++ Builder 4.0”，文魁資訊，1999
- [2]李勁，”精通 C++ Builder 4.0”，文魁資訊，1999
- [3]林宸生，”數位信號：影像與語音處理”，全華科技，1997
- [4]余明興、吳明哲、黃世陽、黃豐龍，”Borland C++ Builder 4.0”，松崗電腦，1999

第五章 結果與討論

本系統最大的成果在於對病友的臨床測試，而我們現已接觸一脊錐損傷之病友，這位病友經測試此系統，是可以達到準確的利用此系統來表達其所想要表達的意思，而因為其為脊錐損傷患者，可以言語，所以其測試本系統時，可以了解病友在運用時，有無不適，可以適時跟我們表達，見 Fig 6.1



圖 5.1 病友使用圖

在測試期間，由於此程式屬新開發之新程式，而接觸的這位病友，他也很樂意配合，我們在一開始做測試，他覺得長期戴著此輔具會感到發熱而不舒服，因此會晃動而偏移，造成不穩，但再一次又一次，經過他建議，及修改之下，我們程式判斷越趨精準，而我們也比較這系統跟正常人使用有無差別，在我們經測試以後，正常人使用跟他使用，並無差別，但我們有一點遺憾，因為這系統是最適合運用在『運動神經元』病友，但我們因為沒機會接觸，所以不知道這系統對她們是否更適合來使用。

而在做這專題期間，感到比較困擾的還有，因為電腦運算並不是很強，再影像處理方面有時會因此當機，常常會造成測試困擾，在修改之後，雖然可以順暢的執行主要程式，但也忍痛拿掉一部分方便的副程式，當然我們會在努力將程式改的更簡單，讓副程式也可運用。