

以視線軌跡為基礎的人機介面

余立強 廖文宏

國立政治大學資訊科學系

Email: {g9625,whliao}@cs.nccu.edu.tw

摘要—眼動儀是具未來性的人機介面裝置，本研究設計並實作低成本頭戴式眼動儀，並將其應用於以視線軌跡為基礎的人機介面。由於眼動儀的操作易受到雜訊與角膜反射點等影響，本研究利用瞳孔周圍暗色點比例較高的特性，提出新型演算法，增加瞳孔定位之準確性，以改善眼動儀之精確度；此外，本研究利用製作的眼動儀，實作數個以視線軌跡為基礎的人機介面，包括視線軌跡網頁瀏覽器、強化眼睛注視照片區域等應用。

關鍵詞—Pupil Detection、Eye Tracking、Human Computer Interface

一、前言

使用者與電腦溝通方式，主要使用鍵盤與滑鼠，隨著電腦週邊設備的發展，也有使用手寫、語音、肢體動作等較特殊的溝通方法，而使用眼睛的視線軌跡來做為與電腦溝通的橋樑，例如眼控滑鼠、利用視線軌跡操控居家環境(Home and Environment Control)[3]、輸入文字(Eye Typing)[4]與玩遊戲[5][6]等，屬較新的研究領域，也是本研究探討的主題。

本研究利用視線軌跡來與電腦溝通，希望不僅讓有缺陷的障礙者能夠透過視線軌跡向外界傳達訊息，也讓一般使用者，以簡單且自然的方式與電腦互動。主要目標分述如下：

(一) 瞳孔偵測與定位演算法的改善

從先前研究[1]與相關論文[7]發現，目前以影像為基礎的瞳孔偵測演算法，不管在近紅外線攝影機或一般的可見光攝影機拍攝的情況下，都易受角膜反射點、拍攝解析度與光源的影響，導

致偵測到的瞳孔，位置出現誤差，本研究改善近紅外線攝影機下抓取瞳孔的演算法，利用瞳孔在近紅外線下，反射率比眼白、眼皮等瞳孔周圍組織來的低，而呈現較為暗的像素點(dark pupil 情況)，與如果是瞳孔輪廓上的點，其周圍暗點比例會大於一定比例的特性，並透過消除雜點的演算法來減輕角膜反射點、攝影解析度的影響。

(二) 實作以視線軌跡為基礎的人機介面

透過上述建置之眼動儀，實作視線軌跡網頁瀏覽器，讓使用者能使用視線軌跡瀏覽網頁，並透過使用者的視線給予幫助；實作可記錄觀看行為的照片瀏覽器，讓使用者注視照片數秒，並對照片中注視的區域做處理，凸顯照片的主題。

本文於第二段講述相關之文獻與研究，第三段說明改善的瞳孔偵測演算法，第四段則是實驗器材與精確度的分析，第五段介紹實作的人機介面，最後於第六段分享本文的結論與未來可能的後續發展。

二、相關研究

眼動儀的建構目前主要是以影像為基礎，而依影像的來源，又可分為可見光頻譜(visible spectrum)與近紅外線(near-infrared)，如圖 1。

以影像為基礎的眼動儀是使用攝影機拍攝眼睛，藉由影像處理與電腦視覺技術找出眼睛中心(瞳孔或是虹膜中心，Pupil Center)與角膜反射點(Corneal Reflection)(簡稱 PCCR)，並利用校正程序所取得的參數，將眼睛移動與頭部轉動，轉換到螢幕上的座標。

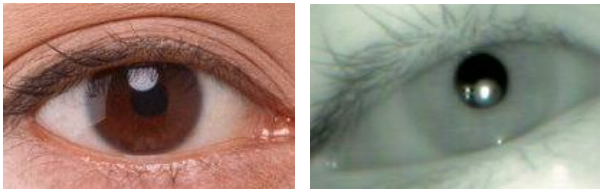


圖1 可見光譜(左)與近紅外線(右)的眼睛圖像

依上述說明，捕捉視線軌跡，大略分成三步驟：找出眼球中心於來源影像的位置、找出眼睛對應螢幕座標參數的校正程序、將眼球中心位置對應到螢幕座標，架構圖如圖 2。

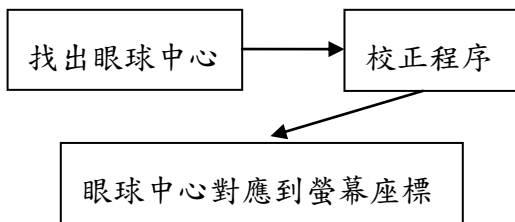


圖2 眼動儀架構圖

(一) 找出眼球中心

依照光線的來源是可見光還是近紅外線，而有不同的作法，近紅外線目前較多人採用，原因是在近紅外線下，瞳孔與虹膜等的差異較大，且瞳孔較不受到眼皮等其他周圍組織所遮蔽，所以要找出眼睛的移動相對容易。

近紅外線光源下找出眼球中心，大都先找出符合瞳孔的橢圓或圓，此橢圓或圓的中心就是要找的眼球中心；要找出符合瞳孔的橢圓與圓，許多方法是先找出代表瞳孔輪廓的特徵點，再利用這些特徵點找出符合瞳孔的橢圓或圓，代表的演算法有[8]、[9]、[10]、[11]、[12]、[13]等。

[8]須先移除角膜反射點，並利用邊緣值(edge)來當作特徵點，然後透過 RANSAC 去找最佳橢圓，但由於眼角等部分，邊緣值也很強烈，RANSAC 需要執行多次才會有好的結果，且當找出的特徵點中，不是瞳孔部分的特徵點過多，處理速度會過慢且抓到不正確的橢圓機率大增。[9]也是利用邊緣值當作特徵點，並使用圓形逼近法(circle approximation method)與座標平均

法(coordinates averaging method)來找出瞳孔中心，但當角膜反射點在瞳孔附近周圍時，角膜反射點周圍會被當成特徵點，造成不準確；[10]須先移除角膜反射點，也使用邊緣值來當特徵點，然後將特徵點依位置的分佈，分成多個區域(grid)，並利用這些區域的特性將雜點移除，最後使用 SVD 找出最佳橢圓。[11]與[12]也是採用邊緣值來做為特徵點，也需先移除角膜反射點，[12]使用 convex hull 將瞳孔內部填滿，再對填滿的圖找邊緣，最後再用 SVD 找出最佳橢圓。以上方法大多需先移除角膜反射點，但當影像中除了角膜反射點外，還有其他亮點區域，或是當角膜反射點一半在瞳孔內部區域，一半在瞳孔外部區域，此時無法將角膜反射點完美移除，使上述方法無法精確的抓出瞳孔。[13]是先保留影像較黑部分(二值化)，並對較黑部分找邊緣值，之後對影像找 connected component，並將特徵點過少的物件(component)移除，連續最多的區塊(component)就是代表瞳孔的區塊，最後也是利用 SVD 找出最佳橢圓，此方法的問題在於無法處理角膜反射點過大且遮蔽到瞳孔周圍部分，當此種情形發生時，角膜反射點的周圍會與瞳孔周圍的特徵點相連，使最後用 SVD 找出的橢圓產生偏差，另外也無法處理瞳孔周圍被切成好幾段的情形。

(二) 校正程序

找出眼球中心後，再做校正程序將眼球中心對應到螢幕座標的參數找出，校正程序可以分成兩種作法，一種是建立 3D 眼睛模型(3D Eye Model)，一種是利用線性迴歸(Regression)。

本研究使用的校正程序是利用線性迴歸方法，線性迴歸的作法是讓使用者看螢幕上 9 個點(圖 3)，每個點會記錄眼球中心點與目前校正點的座標，再將這九對點利用一次多項式、二次多項式或是 Homographic 的方式找出最佳的對應參數，根據 [8] 的研究發現，二次多項式與

Homographic 的結果較佳。

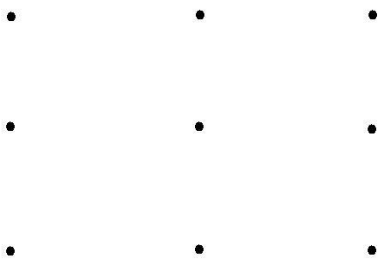


圖3 校正程序的九個點

(三) 眼球中心對應到螢幕座標

做完校正程序，我們就可取得對應參數，之後每個眼球中心點乘上此對應參數，就可找到相對應的螢幕座標。

三、 瞳孔偵測演算法

根據[7]，目前瞳孔偵測演算法受到近紅外線攝影機拍攝眼睛的影像解析度、角膜反射點與光源的影響很大，所以有的演算法會先將角膜反射點移除並做一些影像的前處理[8][12]，再做瞳孔偵測，但是這些步驟會使整體運算速度降低，且當影像中除角膜反射點外，還有其他亮點區域，或是當角膜反射點一半在瞳孔內部區域，一半在瞳孔外部區域，此時去除角膜反射點會變得較為複雜與不容易；目前也有研究在不移除角膜反射點的情況下來抓取瞳孔[13]，但仍無法處理角膜反射點把瞳孔切成兩塊的情況與角膜反射點過大且遮蔽到瞳孔周圍部分的情形，本研究改善目前的瞳孔偵測演算法，希望能夠在相對較低的解析度且不須移除角膜反射點與不做任何的影像前處理的情況下，偵測出瞳孔。

改善的演算法，主要結構與 Starburst 演算法[8]類似，都是先找出代表瞳孔的特徵點，並對這些特徵點找最佳橢圓；演算法與 Starburst 的不同在於找出特徵點的方法不同，Starburst 只用邊緣值(edge)的大小來篩選特徵點，所以必須先移除角膜反射點，否則角膜反射點的周圍會被當成特

徵點，但是當影像較模糊、影像解析度較低與亮點區塊過多時，無法將角膜反射點完整的移除或是無法將移除的角膜反射點完整的填補，會造成找到的特徵點錯誤率提高，使找到的橢圓與實際的瞳孔差異變大，另外與 Starburst 演算法還有一些不同，本研究不使用 RANSAC 來找符合的橢圓，而是用所有找到的特徵點直接去找橢圓，所以找到的橢圓的變化程度較低，但相對的雜點的影響變得較大，因此本研究加入一些去除雜點的方法，並用一些基本的影像處理技巧篩選找到的特徵點，來減輕雜點的影響，本研究不使用 RANSAC 的原因是特徵點數量較多，使用 RANSAC 會使運算的時間增加，較不適合互動式人機介面的應用。依上述說明，改善的瞳孔偵測演算法，主要有三個步驟：偵測特徵點、移除雜點、找最佳橢圓。

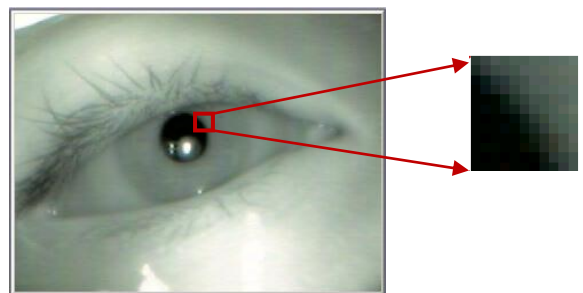


圖4 近紅外線攝影機下的瞳孔特性

(一) 偵測特徵點

偵測特徵點是利用瞳孔在近紅外線攝影機下，會比其他眼球組織來得相對明顯而呈現黑色(如圖 4)，利用此特性，可發現在瞳孔輪廓上的點，以此點為中心的正方形範圍，暗點的數目會占此區域一定的比例(如圖 4 右側)，所以只要找出這些點，就可以找到符合瞳孔的橢圓。

暗點數目在以瞳孔輪廓上的點所形成的正方形範圍 $k \times k$ 中的比例，可以從代表瞳孔的橢圓面積與特徵點所形成的 $k \times k$ 正方形面積推算出來；由於代表瞳孔的橢圓趨近於圓，所以可以假設 k 不大於圓的直徑 $2r$ ，則 k 的值为 $3 \leq k \leq 2r$ ， $k \geq 3$ 是因為 k 太小沒有意義，考慮 $k=3$ 到 $k=2r$

的情況：

$k = 3$ 時，暗色點在正方形面積的分布一定超過 $4 \div 9 \approx 0.44$ ，如圖 5。

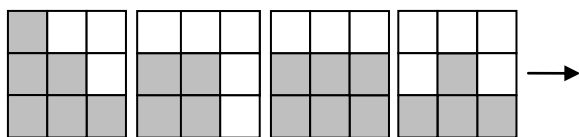


圖5 $k=3$ 時暗色點的分布

$k = 2r$ 時，暗色點在正方形面積的分布如圖 6；由可知，暗色點的面積比正方形面積為 $\pi r^2 / 2 : 4r^2 = \pi : 8$ ，大約是 0.39。

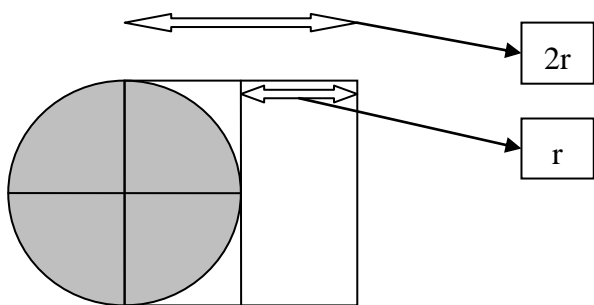


圖6 $k=2r$ 時暗色點的分布

由以上兩種情況可以得知，暗色點的比例會在 0.39~0.44 之間，所以暗色點的比例可以取兩者的最低值當作暗色點的比例，也就是 $\pi/8$ 。

要找出這些特徵點，本研究對拍攝到的瞳孔圖片的每個像素點做判斷，若像素點符合以下兩個條件就是特徵點，一個條件是特徵點會跟周圍的點有相對明顯的變化(也就是找瞳孔輪廓)，可以透過 Sobel 邊緣偵測演算法來實現，另一個條件是特徵點周圍的暗點數目的比例要大於 $\pi/8$ ，但是由於角膜反射點的周圍也具有強烈邊緣，若與瞳孔重疊，這些點也會被當成特徵點，為了處理角膜反射點所造成的影響，我們再加入特徵點周圍不能出現亮點這個條件，希望能減輕角膜反射點的影響，若先將角膜反射點移除，則此條件可以不考慮。

因此要是特徵點必須符合以下兩個條件：

(1) 邊緣值 $ed(x, y)$ 要大於門檻值

$smallestEdge$ ，公式(1)。

$$isEdge(x, y) = \begin{cases} 1, & edge(x, y) > smallestEdge \\ 0, & otherwise \end{cases} \quad (1)$$

(2) 以此點為中心的 $k \times k$ 正方形，此正方形範圍內之暗點數目，要大於正方形面積的 $\pi/8$ ，且亮點不能出現(公式(2))。其中 $n_d(x, y)$ 為暗點數目， $n_b(x, y)$ 為亮點數目，暗點定義為灰階值小於 I_{dark} ，亮點定義為灰階值大於 I_{bright} (公式(3)、(4))。 k 值取決於拍攝的瞳孔照片中，角膜反射點的大小，以 320×240 的影像為例，我們假設瞳孔的大小不超過整張圖的四分之一，且角膜反射點的大小不超過瞳孔大小的五分之一，因此角膜反射點不超過 240 除以 4 再除以 5，也就是 12 個 pixels，因此 k 值可定為 12。

$$features(x, y) = \begin{cases} 255, & isEdge(x, y) = 1 \wedge n_d(x, y) \geq \frac{\pi k^2}{8} \\ \wedge n_b(x, y) = 0 \\ 0, & otherwise \end{cases} \quad (2)$$

$$isDark(x, y) = \begin{cases} 1, & value(x, y) > I_{dark} \\ 0, & otherwise \end{cases} \quad (3)$$

$$isBright(x, y) = \begin{cases} 1, & value(x, y) > I_{bright} \\ 0, & otherwise \end{cases} \quad (4)$$

由於瞳孔大小於整張圖中，大小不會超過四分之一，且要找的特徵點間之距離也都不會太遠(最遠的距離為橢圓的兩倍長軸長)，為了避免尋找特徵點時對整張圖都找，造成處理速度過慢，可先使用二分法(由內而外)對圖片切割的分界線去找一些特徵點，再從這些點中取出任何一點，

以此點為中心的長方形範圍(假設瞳孔的大小不會超過整張圖的四分之一，所以長寬可以設定為原本圖的四分之一)，去找特徵點，就可找到所有瞳孔的特徵點，如圖 7 左側。

取得特徵點後，利用 Singular Value Decomposition(SVD)算齊次解，就可算出符合瞳孔的橢圓方程式，做法是將每個特徵點的座標 x 與 y 帶入橢圓公式(5)，會得到 $a..f$ 的係數，將每個特徵點的 $a..f$ 係數排成一個矩陣 A (公式(6))，並對此矩陣計算齊次解(公式(7))，之後對這個橢圓方程式做整理，即可找出橢圓的長軸、短軸、角度與中心點。找出的結果如圖 7 右側。

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (5)$$

$$A = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \quad (6)$$

$$A\bar{x} = 0 \quad (7)$$

3 1 2 4

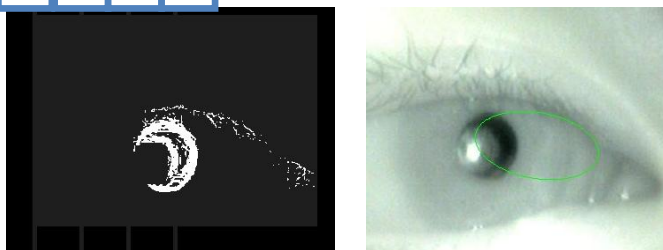


圖7 沒做侵蝕所找到的特徵點與橢圓

雖找到許多瞳孔特徵點，但這些特徵點中，有許多雜點且特徵點過多，會使整體運算速度降低且使找到的橢圓不精確，為了減少雜訊與特徵點，我們對剛剛找到所有特徵點的範圍做侵蝕(Erode)一次，侵蝕後的結果如圖 8。

做完侵蝕後，仍可能殘餘雜點，我們利用目前特徵點的幾何中心，將過遠的雜點移除。

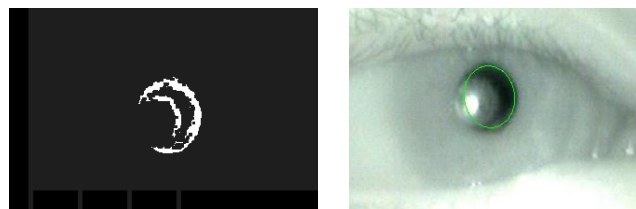


圖8 做侵蝕後所找到的特徵點與橢圓

(二) 消除雜點

經實驗發現，當角膜反射點範圍過大或擴散(戴眼鏡時，鏡片影響)，且與瞳孔邊緣重疊，會讓角膜反射點擴散的最外圍被當成特徵點(三、圖 9)，為了解決此問題，本研究利用瞳孔特徵點在不受角膜反射點等影響的情況下，特徵點會呈現平均分布(圖 10)，並利用此特性將雜點移除。



圖9 角膜反射點擴散時找到的特徵點與橢圓



圖10 最佳的特徵點分布

在此情況下，特徵點的幾何中心點會在瞳孔中心附近，另外代表瞳孔的橢圓會接近圓(當瞳孔在攝影機拍攝位置的中間)，所以若是好的幾何中心點，特徵點與中心點的距離接近平均距離，因此可利用原本特徵點的幾何中心的四周比例，來調整幾何中心點的位置，直到新幾何中心點與大部分特徵點的距離與平均距離相近。

由於侵蝕後的特徵點，密度分佈不一致(如圖 8)，會造成計算出的平移方向錯誤，為減輕此

情況所造成之影響，先將做侵蝕(Erode)後的特徵點範圍，再做一次細化(Thinning)，來解決特徵點密度分布不均的問題，也可讓運算過程減輕，但為避免細化後，原本保有的資訊流失，因此把細化前與細化後的特徵點都保留，保留的做法是把細化後要刪除的特徵點的灰階值，設成較低的值(如 130 等，低於 255)，如此就可在運算時，透過門檻值來取得是侵蝕後的特徵點還是侵蝕與細化後的特徵點；細化後之結果如圖 11。

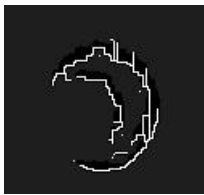


圖 11 做侵蝕與細化後所找到的特徵點

細化後，開始移除雜點，移除雜點的步驟：

- (1) 計算目前特徵點的幾何中心點(圖 12)。



圖 12 找出特徵點的幾何中心點

- (2) 找出細化後每個特徵點，是位在幾何中心點的左上(A1)、左下(A2)、右上(A3)、右下(A4)區域的哪個區域。

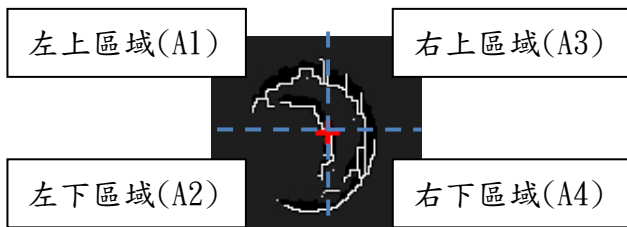


圖 13 根據幾何中心點將圖片切割成四區域

- (3) 計算每區域的特徵點數目，左上區域(A1)的數目為 LT，左下區域(A2)的數目為 LB，右上區域(A3)的數目為 RT，右下區域的數目(A4)為 RB。

- (4) 計算幾何中心點上半部(A1 與 A3)的特徵點個數 upperPoint、左半部(A1 與 A2)的特徵點個數 leftPoint 與所有特徵點的數目 FC。

- (5) 利用細化後的特徵點，在此四區域的數目所產生的相對比例 \overline{Pe} ，調整幾何中心點的位置；利用上區域(1 + 3)與下區域(2 + 4)的比例 Pe_y 來調整幾何中心點的上下移動；利用左區域(1 + 2)與右區域(3 + 4)的比例 Pe_x 來調整幾何中心點的左右移動，如公式(8)、(9)。

$$\overline{Pe} = \left(\frac{\text{upperPoint}}{FC}, \frac{\text{leftPoint}}{FC} \right) - 0.5 \quad (8)$$

$$\overline{add} = \overline{Pe} \times D_{avg} \quad (9)$$

$$D_{avg} = \frac{1}{FC} \sum_{i=1}^{FC} \text{distance}_i \quad (10)$$

平移過程需要平移距離 \overline{add} ， \overline{add} 是利用幾何中心點與所有特徵點的平均距離 D_{avg} (公式(10))來做平移距離的標準。

中心點的平移過程中，必須要有中止條件，假設特徵點 P_i 與中心點 C 的距離接近平均距離的特徵點個數為 N_{Fit} (公式(11)、(12))， $\max N_{Fit}$ 為前幾次平移中最大的 N_{Fit} ，則重複平移中心點，直到新幾何中心點的 N_{Fit} 大於九成，或是 N_{Fit} 大於所有特徵點(FC)的一半且目前的 N_{Fit} 與 $\max N_{Fit}$ 接近且上一個 N_{Fit} 比目前的 N_{Fit} 大，如公式(13)，並於每次平移結束時，更新平均距離 D_{avg} 為目前符合特徵點的平均距離，平移後的幾何中心點如四、圖 15 中十字的圖示。另外 N_{Fit} 大於 FC 一半這個條件，我們設定每做十次平移，就將參數(初始為 50%)減 5%，讓平移一定會中止。

找到符合的中心點後，利用平均距離 D_{avg} ，將距離新的幾何中心點(C_{new})過為偏離的特徵點

(侵蝕沒細化的特徵點)移除(預設為超過平均距離的 140%)與過為靠近的特徵點移除(預設為小於平均距離的 90%)，如圖 15 中灰色的點為消除的雜點；偏離特徵點的範圍取較遠的原因是由於橢圓可能較為扁，與瞳孔內部雜點過多，造成所求出的平均距離太小，所以取遠一點的距離，希望能改善此問題。

$$N_{Fit} = \sum_{i=1}^{FC} isFit(P_i) \quad (11)$$

$$isFit(P_i) = \begin{cases} 1, & |distance_{p,c} - D_{avg}| \leq \frac{D_{avg}}{5} \\ 0, & otherwise \end{cases} \quad (12)$$

$$isGoodC = \begin{cases} 1, & \left[\begin{array}{l} N_{Fit} \geq \frac{1}{2} FC \\ \wedge N_{Fit} \geq \frac{9}{10} \max N_{Fit} \\ \wedge (preN_{Fit} - N_{Fit}) > 0 \end{array} \right] \vee \left[N_{Fit} \geq \frac{9}{10} FC \right] \\ 0, & otherwise \end{cases} \quad (13)$$

另外平移方向的計算方式是用估計的，有時會受雜點影響而使偏移方向產生錯誤，為了避免偏移錯誤，可以利用所有特徵點與平移後的中心點的 X 方向距離與 Y 方向距離差異(公式(14))，與前一次平移的 N_{Fit} 比目前的 N_{Fit} 大，來判斷目前偏移方向有無錯誤；由於每次平移，X 方向與 Y 方向都會平移，如此會造成永遠都不會平移到正確的位置，我們假設 Δ 為 Δ_x 與 Δ_y 中較大者，可以利用 Δ 決定當平移方向錯誤時，哪個方向須要停止移動(Δ_x 與 Δ_y 中較大者，設成 0)，並於發生時將 Δ 改成較小者，公式如(15)。

$$\Delta_x = \sum_{i=1}^n (x_i - C_x)^2, \Delta_y = \sum_{i=1}^n (y_i - C_y)^2 \quad (14)$$

$$add_x = \begin{cases} -add_x, (preN_{Fit} - N_{Fit}) > 0 \wedge \Delta > 0 \\ 0, \left[\begin{array}{l} (preN_{Fit} - N_{Fit}) > 0 \wedge \Delta > 0 \\ \wedge moreAdd = add_x \end{array} \right] \\ add_x, otherwise \end{cases}, \quad (15)$$

$$add_y = \begin{cases} -add_y, (preN_{Fit} - N_{Fit}) > 0 \wedge \Delta < 0 \\ 0, \left[\begin{array}{l} (preN_{Fit} - N_{Fit}) > 0 \wedge \Delta < 0 \\ \wedge moreAdd = add_y \end{array} \right] \\ add_y, otherwise \end{cases}$$

去除雜點後，將剩下特徵點，利用 SVD 找出符合的橢圓方程式，經過移除雜點的後處理，發現能夠將角膜反射點過大或擴散這個問題減輕一些，找出的瞳孔如四、圖 14 與四、圖 15。

四、實驗器材與精確度分析

實驗器具的頭戴式眼動儀外觀圖如圖 16。攝影機是使用本身具有近紅外線 LED 與具有近紅外線濾鏡的設備，最大解析度可以到 640*480，Frame 數最多可以到每秒 30 frames。

精確度分析的作法是在做完校正程序後，回頭去看校正程序所注視的九個校正點，每個校正點紀錄 60 個注視點，並比較此 60 個注視點與正確的校正點之間的差異，差異的算法是使用歐式距離(Euclidean Distance)，令注視的點為 $p_i, 1 < i < numP_j$ ， $numP_j$ 為 3 秒內看的點的個數，校正的點為 $q_j, 1 < j < 9$ ，則每個校正點 q_j 的差異公式如公式(16)：

$$\frac{1}{numP_j} \sum_{i=1}^{numP_j} \sqrt{(p_i(x) - q_j(x))^2 + (p_i(y) - q_j(y))^2} \quad (16)$$

測試結果平均誤差約 23pixels ~ 39pixels，精確度主要是跟影像品質有關係以及頭部移動所造成的誤差；在做實驗的螢幕(96 DPI)上 1 cm 大約是 37.7953 Pixels，實驗螢幕是 SAMSUNG 的

SyncMaster 940BW，點距為 0.285mm；表 1 是三個受測者做的結果(攝影機拍攝的解析度為 640x480)，凝視點誤差圖如圖 27(最後一頁)。

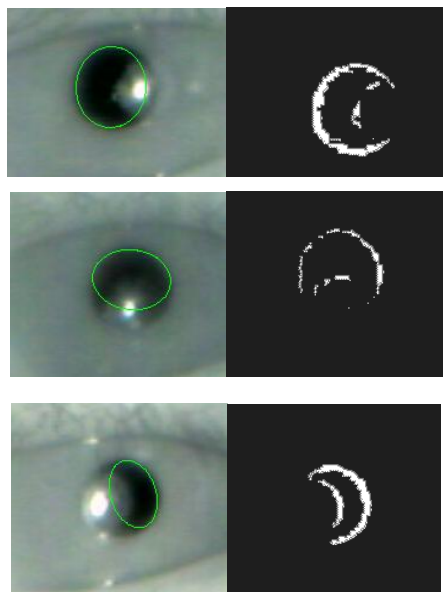


圖 14 沒移除雜點時找到的橢圓

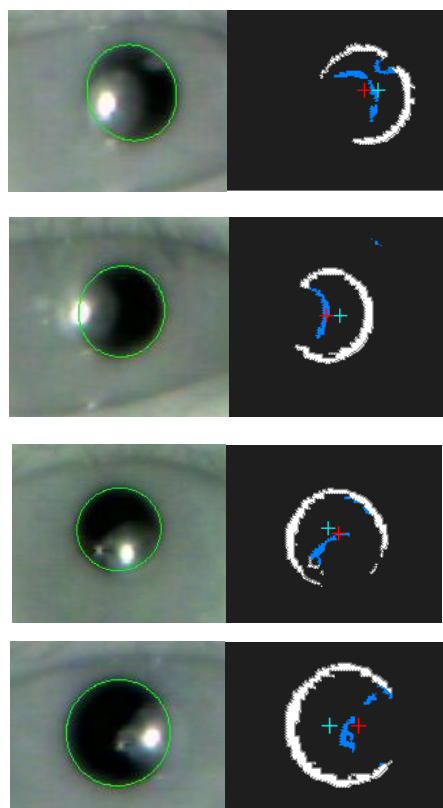


圖 15 移除雜點找到的橢圓與平移前後的中心點
(紅：平移後、青藍：平移前)



圖 16 眼動儀外觀圖

表 1 頭戴式眼動儀的精確度

Point (pixels)	User1	User2	User3	各點平均
1	45.23	13.68	12.06	23.65
2	26.15	24.11	12.73	20.99
3	72.59	20.74	39.17	44.17
4	31.51	22.92	22.07	25.50
5	22.88	32.55	24.09	26.51
6	30.96	22.21	15.32	22.83
7	38.81	39.73	32.50	37.01
8	35.31	35.60	27.29	32.73
9	47.42	29.65	17.59	31.55
平均	38.98	26.80	22.53	29.44

五、以視線軌跡為基礎的人機介面

利用前述所建構的眼動儀，實作數個視線軌跡為基礎的人機介面，包括(1)以視線軌跡為基礎的網頁瀏覽器，(2)可紀錄觀看行為模式的照片瀏覽器。

(一) 視線軌跡為基礎的網頁瀏覽器

利用微軟.NET 平台中的 Internet Explorer(IE)

套件，實作一個視線軌跡的網頁瀏覽器，具備以下的功能：

- (1) 能夠接收眼動儀傳出的眼睛注視座標。
- (2) 能夠將目前觀看的網頁內容做分塊，好讓電腦能夠知道使用者目前觀看的內容，並可限制網頁分塊的大小。(圖 17)
- (3) 標示目前使用者注視的區塊(圖 18)。
- (4) 根據注視的時間，來放大大使用者注視的區塊(圖 19)。
- (5) 依據使用者注視的位置，做適當的捲動(圖 20 與圖 21)。
- (6) 將眼睛注視的時間做為滑鼠的左鍵。



圖17 網頁重建區塊的畫出結果(10x10 pixels)



圖18 注視的座標圖示與注視的網頁區塊



圖19 放大的網頁區塊



圖20 還未注視到螢幕的前 1/5 或後 1/5



圖21 注視到螢幕區域的後 1/5(網頁開始捲動)

(二) 可記錄觀看行為的照片瀏覽器

藉由眼睛注視的焦點，我們可以推估觀看者對於相片的哪些部分較具興趣，而將此互動過程紀錄之後，可做為組織相片的線索，此外，亦可強化眼睛注視照片的區域，由於看照片時，常常會發現自己所拍攝的照片，拍到的部分可能過小或太偏離照片的中間，使得照片的主題常常失焦，所以本研究利用眼睛注視照片的軌跡，將照片中注視的區域做一些處理，希望讓使用者一看到照片，就能夠抓到照片傳達的主題與意境。

可記錄觀看行為的照片瀏覽器的人機介

面，主要具有以下功能：

- (1) 接收眼動儀傳出的座標。
- (2) 讓使用者輸入一張照片。
- (3) 記錄使用者注視照片的視線軌跡。
- (4) 利用視線軌跡對輸入圖片做處理。
- (5) 輸出處理過的圖片。

介面如圖 22 與圖 23，模糊的做法如圖 24，輸出的結果如圖 25 與圖 26。



圖22 可記錄觀看行為的照片瀏覽器介面

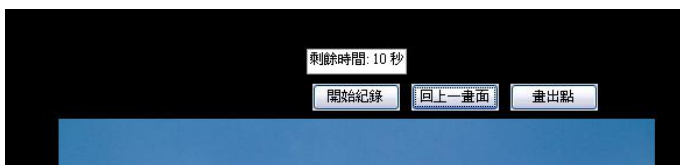


圖23 開始觀看的介面

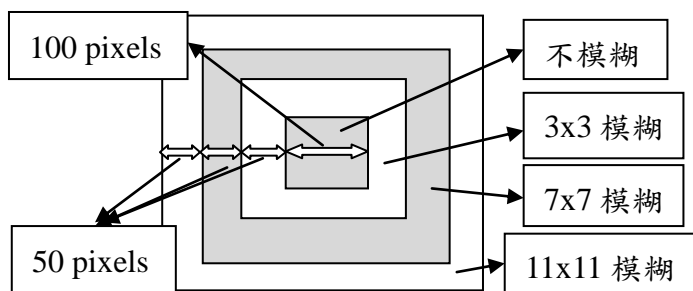


圖24 目前照片模糊的設定

k 點的數目可以利用前後兩個連續的凝視點之間的距離夠遠的個數來決定，兩個連續的凝視點之間的距離，可取頭戴式眼動儀的平均精確度(誤差值)的 2 倍，主要想法是當兩個連續的凝視點之間的距離超過平均誤差值的兩倍時，就代表注視的點從一個區域跑到下一個區域，並記錄換區域時的凝視點當作是 K-means 各群的初始點。



圖25 圖片經過 K-means 的區域中心



圖26 圖片經過模糊的結果

六、 結論

本文針對頭戴式眼動儀進行研究與實作，主要改善於近紅外線下的瞳孔偵測演算法，讓偵測瞳孔時，能減輕角膜反射點的影響並容忍較多的環境變因，並利用建置的眼動儀，實作人機介

面：視線軌跡瀏覽器、可記錄觀看行為的照片瀏覽器人機介面。

由於頭部移動會造成眼動儀精確度產生誤差，可利用動態影像來處理頭部移動造成的問題[2]；頭戴式眼動儀仍需配戴設備，可使用遠距式操作，將攝影機放置於螢幕前固定位置，但瞳孔偵測需做修改[2]。

在眼動儀的精確度分析部分，經實驗發現，主要是受到角膜反射點的影響，而造成對應到螢幕的點，有晃動與不準確的問題，與影像解析度較無關係，未來若能將拍攝眼睛的攝影機的焦距、近紅外線光源 LED 燈的數量與擺放位置等，透過實驗，設計一個讓角膜反射點影響較小的環境，相信能使精確度更穩定且更精確。

隨著科技的發展，硬體與眼動儀的技術的提升，相信眼動儀未來會成為如鍵盤與滑鼠等的人機介面裝置，改善我們操作電腦的習慣。

參考文獻

- [1] 王凱平，移動式眼動儀之實作與視線軌跡分析，政治大學資訊科學系碩士班碩士論文，2008.
- [2] 余立強，以視線軌跡為基礎的人機介面，政治大學資訊科學系碩士班碩士論文，2009.
- [3] Emiliano Castellina, Faisal Razzak, Fulvio Corno, "Environmental Control Application Compliant with COGAIN Guidelines," The 5h Conference on Communication by Gaze Interaction(COGAIN 2009), 2009.
- [4] Jacob O. Wobbrock, James Rubinstein, Michael Sawyer, Andrew T. Duchowski, "Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures," The 3h Conference on Communication by Gaze

Interaction(COGAIN 2007), 2007.

- [5] Howell Istance, Aulikki Hyrskykari, Stephen Vickers, Nazmie Ali, "User Performance of Gaze-Based Interaction with On-line Virtual Communities," The 4th Conference on Communication by Gaze Interaction(COGAIN 2008), 2008.
- [6] Stephen Vickers, Howell Istance, Aulikki Hyrskykari, "Selecting Commands in 3D Game Environments by Gaze Gestures," The 5h Conference on Communication by Gaze Interaction(COGAIN 2009), 2009.
- [7] Droege Detlev, Schmidt Carola, Paulus Dietrich, "A Comparison of Pupil Centre Estimation," The 4th Conference on Communication by Gaze Interaction(COGAIN 2008), 2008.
- [8] Dongheng Li, David Winfield, Derrick J. Parkhurst, "Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW'05) Workshops, pp. 79, 2005.
- [9] Gintautas Daunys, Nerijus Ramanaukas, "The Accuracy of Eye Tracking Using Image Processing," pp.377-380, Proceedings of the third Nordic conference on Human-computer interaction, 2004.
- [10] Dongshi Xia, Zongcai Ruan, "IR Image Based Eye Gaze Estimation," vol. 1, pp.220-224, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), 2007.

[11] Sepehr Attarchi, Karim Faez, Amin Asghari, "A Fast and Accurate Iris Recognition Method Using the Complex Inversion Map and 2DPCA," pp.179-184, Seventh IEEE/ACIS International Conference on Computer and Information Science (ICIS 2008), 2008.

[12] Craig Hennessey, "Eye-Gaze Tracking With Free Head Motion," Masters of Applied Science Thesis, University of British Columbia, 2005.

[13] Somnath Dey, Debasis Samanta, "An Efficient Approach for Pupil Detection in Iris Images," pp.382-389, 15th International Conference on Advanced Computing and Communications (ADCOM 2007), 2007.

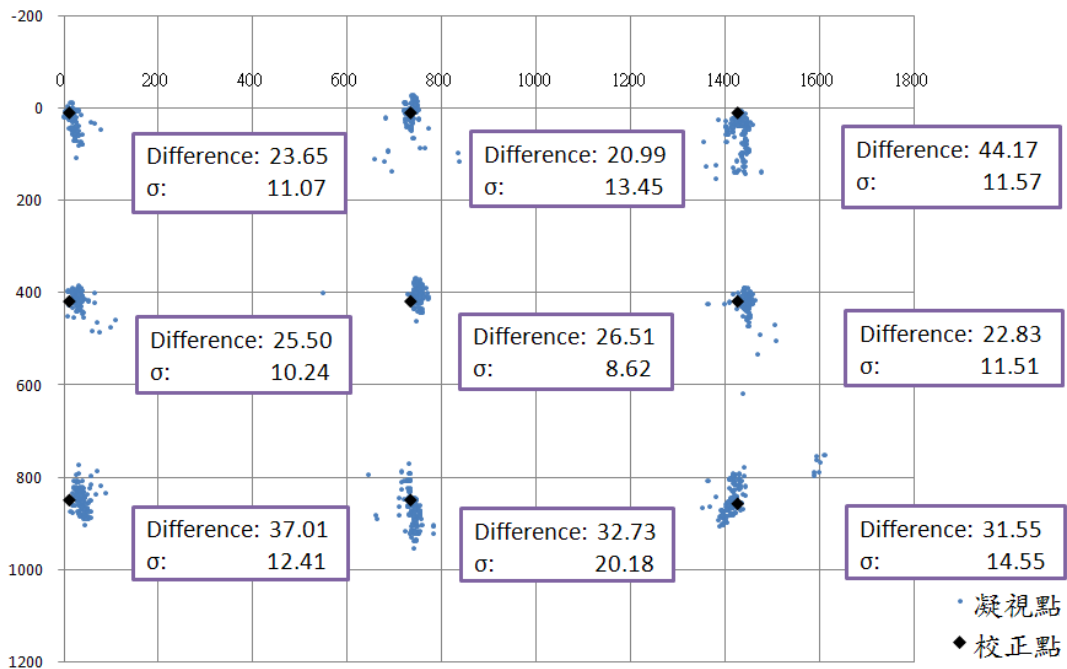


圖27 凝視點誤差圖