

國立交通大學

資訊科學系

碩士論文

一個快速的眨眼偵測演算法



A Fast Algorithm of Eye Blink Detection

研究生：翁崇荳

指導教授：陳玲慧 教授

中華民國九十四年六月

一個快速的眨眼偵測演算法

A Fast Algorithm of Eye Blink Detection

研究生：翁崇荏

Student: Chung-Ren Weng

指導教授：陳玲慧

Advisor: Ling-Hwei Chen

國立交通大學

資訊科學系



Submitted to Department of Computer and Information Science  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in  
Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

# 一個快速的眨眼偵測演算法

學生：翁崇荳

指導教授：陳玲慧 教授

國立交通大學資訊科學學系（研究所）碩士班

## 摘要

眨眼的資訊顯示一個人的疲勞程度，眨眼的偵測可用於監控或警示系統，例如當駕駛員太過勞累時提出警告可以避免車禍的發生。本論文將提出一套快速的人眼定位以及眨眼偵測演算法，適用於複雜背景以及受測者戴眼鏡的情況。我們首先利用膚色的資訊對全圖做出膚色遮罩，然後以形態學運算填補膚色遮罩中的缺陷，由填補後的遮罩計算出人臉範圍區塊，在此範圍中經由水平投影的統計配合人臉的特徵算出眼睛的位置。經由眼睛範圍內深色點數變化量的統計，我們標記出眨眼的時間點。

# A Fast Algorithm of Eye Blink Detection

Student: Chung-Ren Weng      Advisor : Professor Ling-Hwei Chen

Institute of Computer and Information Science

National Chiao Tung University

## ABSTRACT

The information about eye blinks shows one's fatigue. Detecting blink is useful to monitoring system or warning system. Automatically detecting fatigue of drivers may save them from accidents. We propose a fast algorithm for eye location and eye blink detection. This algorithm is adapted to the complex background and the situation which people wear glasses. In the first place, we classify pixels as skin color and non-skin color to build a skin mask of original image. Second, we patch the skin mask with morphological operations and locate the face bounding block. Applying horizontal projection to the bounding block and filtering with face feature conditions, we locate eye region. After gathering statistics of pixels in eye region, we can find out which frames contain eye blinks.

# 目錄

第一章 簡介	1
1.1 研究動機	1
1.2 相關文獻探討	1
1.3 論文架構	4
第二章 人眼定位演算法	5
2.1 流程概要	5
2.2 膚色	6
2.3 型態學	9
2.4 人臉範圍區塊	10
2.5 水平投影及眼睛線選擇	13
第三章 眨眼偵測演算法	16
3.1 流程概要	16
3.2 二值圖	17

3.3	點數變化	18
3.4	眨眼深度門檻值	20
第四章	實驗結果	22
第五章	結論	24



# 圖目錄

圖一 人眼定位流程圖。	5
圖二 人眼定位流程示意圖。	6
圖三 膚色在 Cr 上的效果。	7
圖四 膚色的訓練。	8
圖五 膚色遮罩擷取結果。	9
圖六 鏡框造成人臉的不連續。	10
圖七 圖六(b)中最大的連通元件及人臉範圍區塊。	12
圖八 不同形狀下畫出的範圍區塊。	13
圖七 水平投影。	14
圖十 人眼區域統計。	18
圖十一 連續時間內眼睛條帶點數的變化。	19
圖十二 眨眼深度統計圖。	20

# 第一章

## 簡介

### 1.1 研究動機

眼睛是人臉上相當重要的器官，能夠透露出一個人的生理(例如疲勞)和心理狀況(例如注意力)，例如凝視方向偵測可用於人機介面(Human computer interface)，眨眼偵測可用於駕駛監控及警示系統，基於此原因，已經有許多和眼睛相關的研究。在應用層面上人眼定位(Eye location)是此類研究的第一個步驟，人眼定位在人臉偵測(Face detection)、人臉辨識(Face recognition)、表情分析(Facial expression)、和人機介面這些不同的應用都佔有重要的地位。由人眼能夠收集的資訊之一是眨眼，利用眨眼的偵測能計算眨眼頻率(Blink rate)和閉眼比率(Eye closure rate)，這二項資訊可做為疲勞狀態的指標，常用於駕駛員監控及警示系統。由於人眼定位及眨眼偵測為其他更深入主題的前置工作，所以發展一個快速的演算法是必要的。

### 1.2 相關文獻探討



由於人眼偵測是許多領域的前置工作，大部分眨眼偵測的文獻，也必須是基於良好的人眼偵測情況下，再統計人眼所在區域的資訊才能有好的結果，所以人眼偵測是相當重要的基礎工作，也因此有許多論文以不同的方向來找尋眼睛。

有些論文採用主動式紅外線(IR, infrared)攝影機，由於瞳孔會反射紅外線回到攝影機，瞳孔位置會有高亮度的反光(bright pupil effect)，經由紅外線的開關，可以很容易地找到瞳孔位置[2-3,13-17]，所以眼睛的定位就由瞳孔的定位來完成。然而，紅外線攝影機的硬體不夠普遍，在實際使用上有困難，另外使用者戴眼鏡的情況，會影響紅外線反射路徑而造成瞳孔可能沒有出現反光。這二個問題是使用紅外線攝影機需要解決的。

另外，有些論文使用模版(Template)的方法，模版法就是在偵測之前事先決定模版，模版可以是眼睛的圖樣(Pattern)或形狀(Shape model，例如橢圓和圓)，經由計算原圖各個位置和模版的誤差，來找出眼睛的位置[5-7]。模版分為可變形的(Deformable)和不可變形的，不可變形的模版在偵測上受到很多限制，頭部的轉動會使影像變為和模版不同，導致偵測失敗或者需要額外的模版；而可變形的模版方法需要計算大量的轉換，有計算耗時的缺點。

Kawato 等人提出偵測與追蹤二眼中間的點(稱為 between-eyes)代替偵測眼睛[4,9-11]，作者認為此點比眼睛具有穩定、容易偵測的優點，實作上運算速度也很良好，但實驗結果發現此方法容易受頭髮的影響，而且受測者閉眼時很容易失敗(無法偵測到眼睛)[2]。

Kawaguchi 等人使用邊線(Edge)的資訊偵測人臉邊界，配合人眼附近高反差的特徵，使用分離性過濾器(Separability filter)偵測眼睛[21]。分離性過濾器作法為以一個點為中心，用不同尺寸的圓形計算內外的亮度差距。這個方法在某些圖可以得到很高的正確率，但由於使用邊線的資訊，此演算法只能用於單純背景(素色背景)，在複雜背景中，由於有過多的邊線影響，無法正確找到人臉的邊界，另外分離性過濾器要對各個可能為眼睛的位置做各種尺寸的運算，需要大量的計算時間。

除了先偵測人眼再判斷眨眼的方法之外，有些論文利用連續畫面的差值(Difference)來直接偵測眨眼(也繼續利用眨眼尋找眼睛)，差值法的做法為將二張(時間上)相鄰的畫面相減，找出有變化的位置。差值法的主要想法在於人會不由自主的眨眼，而且眨眼會使前後畫面在眼睛的部分產生較大差值且可以偵測出來[4,8]。差值法問題在於系統必需等待受測者眨眼，而且當眨眼不是畫面中唯一的差值時(例如受

測者有移動的情形)，眨眼的偵測可能失敗，必需等待下一次的眨眼才能再度偵測，所以差值法會有部分時間處於找不到眼睛和眨眼的狀態。此外差值法有時也會將眉毛的抖動等非眨眼的差值誤判為眨眼。

本論文提出的方法，是基於膚色和人臉的特徵，並且使用普通的攝影機(無 IR 功能)取像來做人眼定位，不論受測者是否戴眼鏡、閉眼、或在複雜背景下皆可運作正常。在眨眼偵測部分，我們對人眼區域使用統計的方法計算出門檻值，使用門檻值可以快速的偵測眨眼是否發生。

### 1.3 論文架構



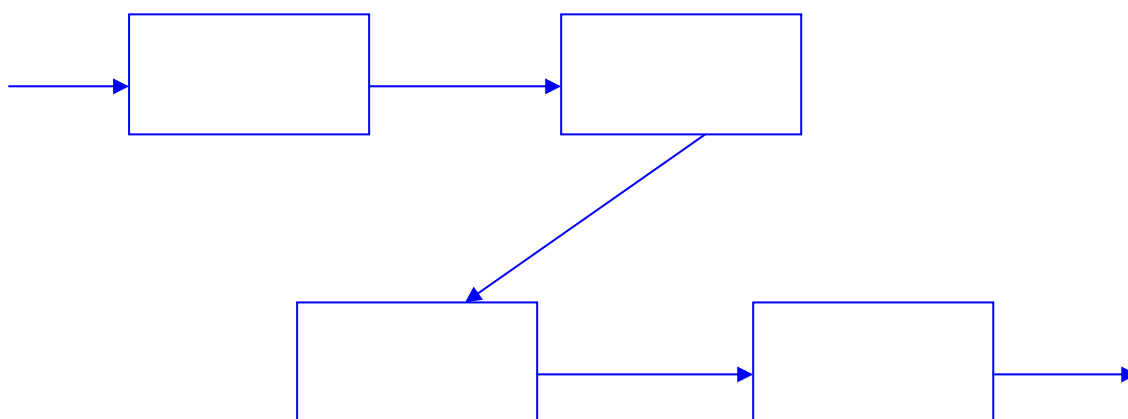
在本篇論文中，我們將提出一套快速且有效率的人眼定位及眨眼偵測演算法。此演算法首先對攝影機取得的每張影像取出人眼的區域，此部分將在第二章中描述。在第三章中，描述如何針對所找到的人眼區域統計眨眼的資訊。第四章為實驗數據與結果，最後第五章為總結。

## 第二章

# 人眼定位演算法

### 2.1 流程概要

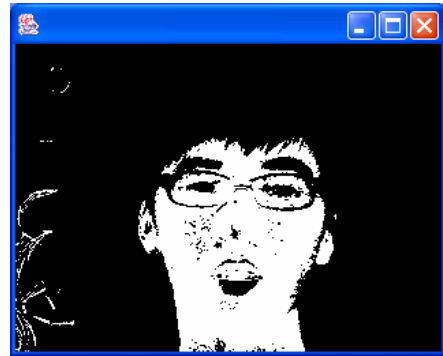
由攝影機或由影片檔取進一張影像之後，我們首先對整張影像判斷哪些點是膚色(Skin color)，做出膚色遮罩(Skin mask)，由於在膚色遮罩上會有部分缺陷，我們以形態學運算(Morphological operation)填補膚色遮罩，接著由填補後的遮罩計算出人臉範圍區塊(Face block)，再對此範圍做水平投影運算，並配合人臉上的特性找出眼睛的位置。圖一為人眼定位的流程圖，而圖二為每步驟的圖例。



圖一 人眼定位流程圖。



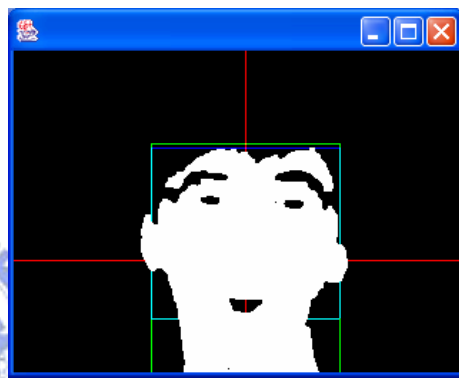
(a)原圖



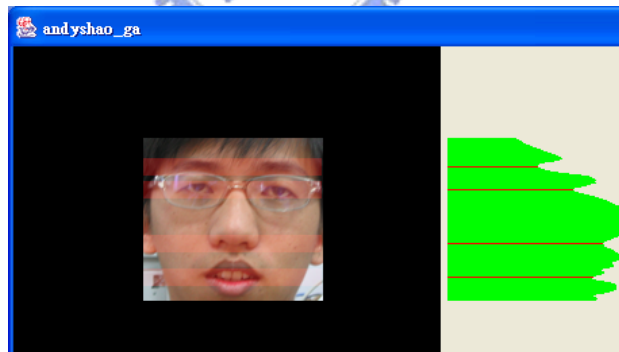
(b)膚色遮罩



(c)形態學運算



(d)人臉範圍區塊



(e)水平投影

圖二 人眼定位流程示意圖。

## 2.2 膚色

色彩在電腦上有數種表示模式 (Color model)，常見的有 RGB、CMYK、HSI、和 YCbCr 等等。在膚色的辨視上，以 YCbCr 的表示模式有較好的效果[1]。在 Y、Cb、Cr 三個頻道(Channel)中，又以 Cr 對膚色的表現最好，如圖三所示。圖三(a)為輸入的原始影像，圖三(b)為圖三(a)的 Cr 值以灰階表示的結果。相當明顯地膚色區域都有相似的 Cr 值。圖三(a)中二個圓圈內的顏色看起來不相似，但在圖三(b)中則很接近、因此利用 Cr 值，膚色能正確地與非膚色區別出來。



Different

Similar

(a) 原始影像

(b)圖(a)的 Cr 值以灰階表示的結果

圖三 膚色在 Cr 上的效果。

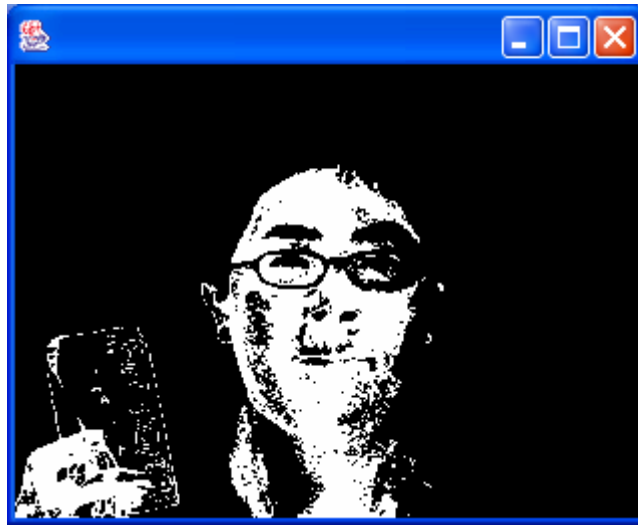
利用 Cr 在膚色辨視上的良好效果，我們可以用很少量的資料訓練(Training)出膚色的範圍，如圖四所示，二小塊紅色區域的 Cr 值做

為訂膚色範圍的依據。在本篇論文中，我們使用訓練資料庫(Training Data)中最高和最低的 Cr 值做為膚色的界限，在此範圍內的 Cr 值皆判斷為膚色，做出膚色遮罩(Skin mask)，圖五顯示一膚色遮罩擷取結果。此方法運用簡單的規則達到相當良好的效果，另一個附加的好處則是只需二個整數的儲存空間，比起模版(Template)法節省數倍到數十倍的空間。



原圖，其中二小塊紅色區域的原值將使用為訓練資料

圖四 膚色的訓練。



圖五 膚色遮罩擷取結果。

## 2.3 型態學

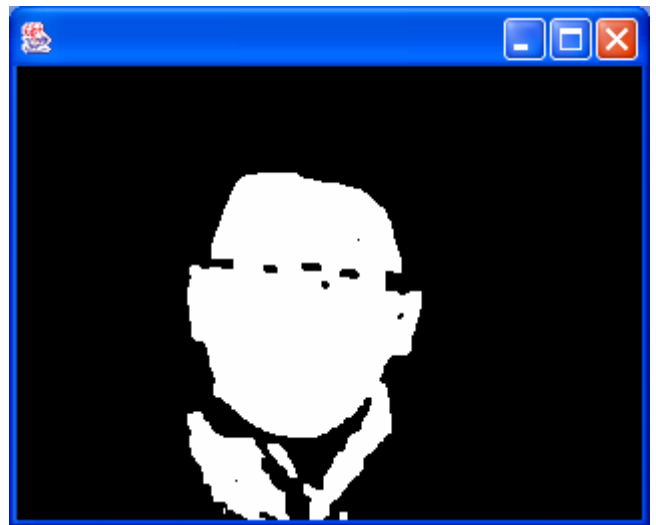


只使用一個頻道(Cr)來辨視本來由三個頻道表示的色彩資訊，在資訊上難免有所遺漏，所以膚色遮罩上的臉部區域會有部分的空洞，如圖五所示。另外，當受測者臉上戴有眼鏡時，會因為鏡框不是膚色，所以鏡框會在膚色遮罩上形成黑色的線條，如圖六所示。





(a) 形態學運算前的膚色遮罩



(b) 形態學運算後的膚色遮罩

圖六 鏡框造成人臉的不連續。


由於上述的二個原因，使得膚色遮罩上的人臉部分，可能無法形成完整的區域，而在後續的處理上需要將人臉視為一個完整區域，所以我們使用形態學的技巧將上述的缺陷補滿。在本論文中使用  $5 \times 5$  的擴張運算(Dilation)後，接著使用一個  $3 \times 3$  的侵蝕運算(Erosion)，如此可將原遮罩上的空洞及黑線補滿，如圖六(b)所示。因為擴張運算和侵蝕運算尺寸上的不對稱，會使得原遮罩區域微微擴大，在之後畫定人臉範圍的步驟時，可以更確保人臉全都在區塊範圍內。

## 2.4 人臉範圍區塊

在複雜背景的情況中，有時背景存在膚色的物件。當人臉是畫面

中最大的膚色物件時，我們對做完型態學運算的膚色遮罩，找出最大的連通元件(Connected component)，即為人臉所在的元件。

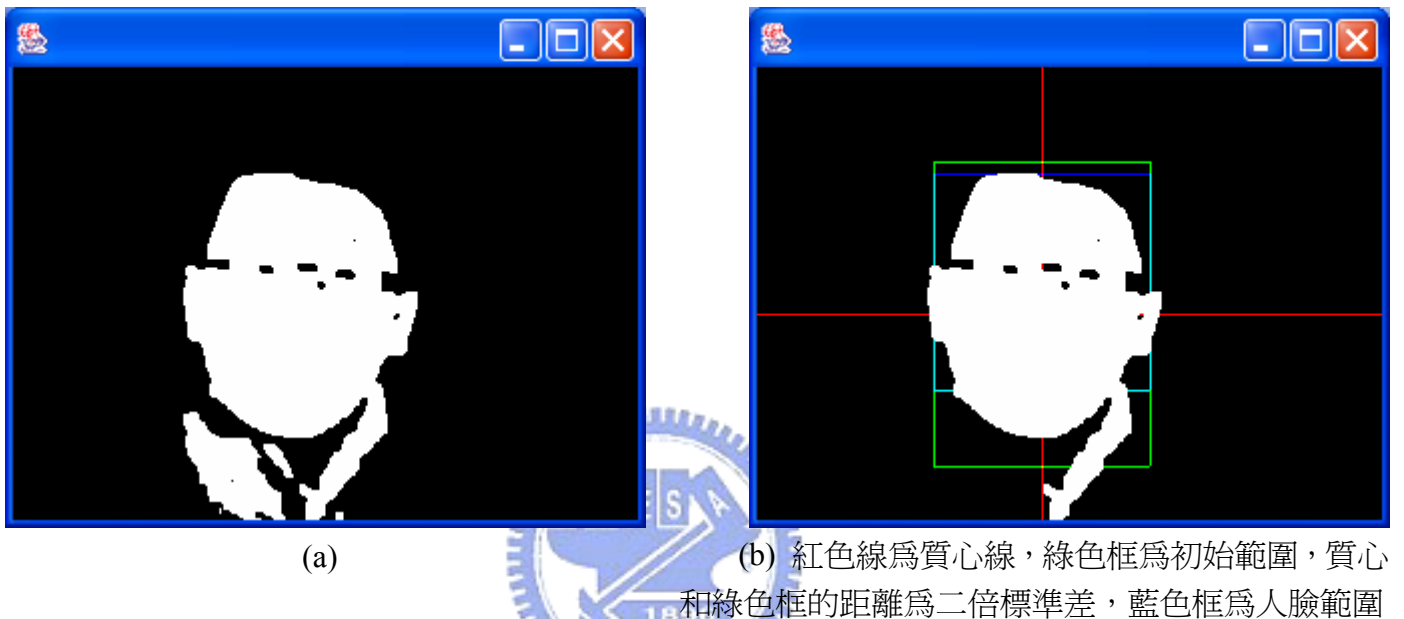
由連通元件畫出人臉範圍，一般常使用邊界盒(Bounding box)的作法，也就是由上下左右四個方向，分別找到第一條符合條件的掃描線作為邊界，就人臉而言，邊界的條件可以是含有一個或數個膚色點。考慮到在複雜背景的狀況，背景中的膚色物件有機會和人臉相連或出現在人臉附近，以邊界盒的方式找到的範圍會過大，本論文採用另一種方式畫出較合適的人臉範圍。



我們的方法稱為二倍標準差法，此方法先對膚色遮罩中找到的最大連通元件，計算出它的質心( $M_x, M_y$ )和水平垂直二方向的標準差( $\sigma_x, \sigma_y$ )，由質心往上下左右四方向各張出二倍標準差的範圍，水平方向取( $M_x - 2\sigma_x$ )到( $M_x + 2\sigma_x$ )之間，垂直方向取( $M_y - 2\sigma_y$ )到( $M_y + 2\sigma_y$ )之間，以此範圍作為人臉的初始範圍，如圖七(b)。二倍標準差法的界線在無雜訊時雖不如邊界盒法準確，但在人臉附近有雜訊影響時，找到的範圍較為準確。

因為人眼位於人臉的上半部，而且下半部的下巴與脖子部分在本論文中不會用到，所以我們取人臉初始範圍的上半四分之三的部分，

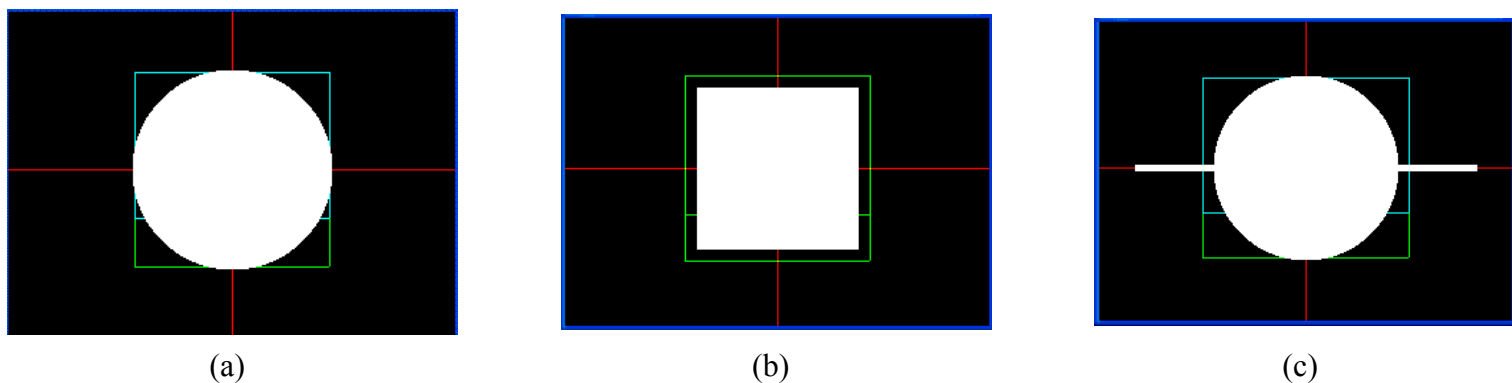
以節省運算範圍的大小，對這部分再做邊界盒運算得到一個範圍，以此範圍當作最後的人臉範圍。



圖七 圖六(b)中最大的連通元件及人臉範圍區塊。

依切比雪夫不等式(Chebyshev inequality)，對於任意的隨機變數，至少有  $3/4$  的機率在  $(M-2\sigma, M+2\sigma)$  區間內[18]，對於較集中的隨機變數，落在此區間的機率會提高。當連通元件是往中央集中的形狀，如圖八(a)和圖八(b)，此方法算出的範圍相當接近邊界盒法作出的範圍。當連通元件為一個集中主體加上部分突出物時，此方法算出的範圍會受突出物影響向外擴張，在突出物較小的情況下，突出物對範圍

的影響相當有限，如圖八(c)。因此這個方法可以有效的找出人臉範圍，比邊界盒法找出的範圍小且準確。



圖八 不同形狀下畫出的範圍區塊。

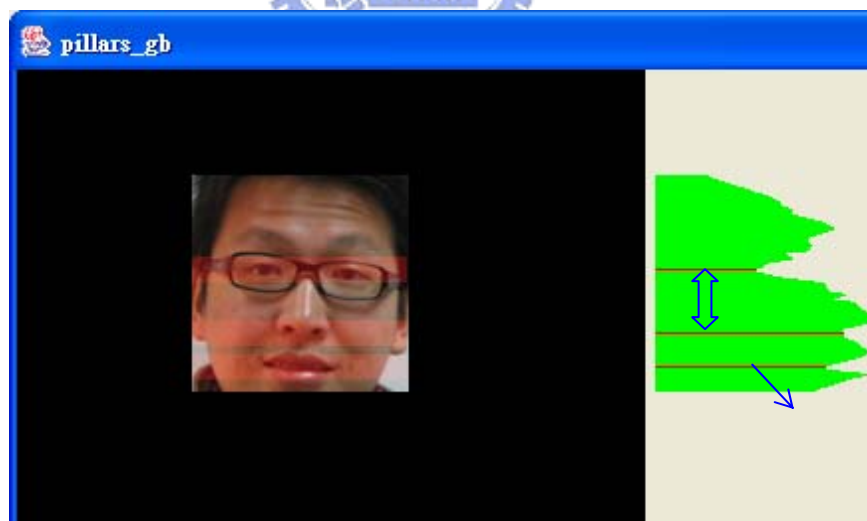
## 2.5 水平投影及眼睛線選擇

水平投影指的是，將原圖每一水平線上的數值累加做出統計圖。由於人臉上的幾個部分，包括眉毛、眼睛、鼻孔、和嘴巴，有較暗的顏色在同一水平線上，所以對人臉取亮度的數值做水平投影時，這幾個部分會在統計圖上形成局部最小值(Local minimum)，或稱為谷底線(Valley Line)，如圖九。經由水平投影統計圖上的谷底位置，我們可以找到眉毛等部位，其所在的水平線分別稱為眉毛線(Eyebrow line)、眼睛線(Eye line)、鼻孔線(Nostril line)、嘴巴線(Mouth line)。此時由水平投影統計圖，我們只能找到許多谷底線，尚不知哪條線對應到哪

個部位，利用這些線之間距離的關係，我們可以找出代表眼睛的眼睛線位置。



(a) 有眉毛線情形



(b) 無眉毛線情形

圖七 水平投影，右邊紅線為谷底位置。

人臉上，眉毛和眼睛的距離近，而眼睛和鼻子的距離遠。當計算每二個相鄰谷底間距(valley interval)的比值時，上述的特徵會造成眼睛線上下的谷底間距產生很大的比值。因此經由計算每二個相鄰的谷底間距比值，我們可以容易地找到眼睛線的位置，如圖九(a)。計算比值的做法可行是基於眉毛線、眼睛線、和鼻孔線都有出現，在受測者眉毛較淡或眉毛不是水平形狀時，由水平投影谷底的偵測不會出現眉毛的位置，因此利用谷底間距比值的方法無法找到眼睛線。

在沒有眉毛線的情況下，額頭的部分因為沒有眉毛線的分割，會形成臉上最大的谷底區間，所以我們找到最大的谷底區間代表額頭，此區間的下緣即可找到眼睛線。綜合以上二個方法，首先我們以比值法尋找眼睛線，當沒有區間的比值符合需求時，我們斷定眉毛線沒有出現，改以最大區間法尋找眼睛線，以此演算法，我們就能經由水平投影統計圖找到眼睛線的位置。

# 第三章

## 眨眼偵測演算法

### 3.1 流程概要

第二章的演算法主要目標，是在單一張畫面找出眼睛的位置，本章繼續對各張畫面的眼睛區域做統計，以偵測眨眼的發生。

由人眼定位部分眼睛線往上下擴張，可括出包含眼睛的條帶，使用臨界值將此條帶做成二值圖(Bi-level image)，經由計算二值圖點數的變化，當變化超過臨界值時，我們判斷這段時間受測者正在眨眼。流程中將使用到的二個臨界值，可以預先設定或由影片的最前面一小段自動計算出。

## 3.2 二值圖

人眼定位部分已經找到眼睛線的位置，將眼睛線往上下各擴張 6 個像素作為上下邊界，再加上 2.4 節中人臉範圍區塊的左右邊界，畫出眼睛條帶，如圖十(b)所示。在此條帶內，我們以一個門檻值將眼睛條帶以如下公式做成二值圖，如圖十(c)所示。

$$I(x, y) = \begin{cases} 255 & , \text{if } f(x, y) < \text{Darkness\_Theshold} \\ 0 & , \text{otherwise} \end{cases}$$

$I(x, y)$ : value in bi-level image

$f(x, y)$ : value in original image



上式中門檻值是使用膚色亮度的一半，實際的計算方式，是在人臉範圍區塊(2.4 節)內，取出所有膚色的點(2.2 節的結果)，將這些點的亮度取平均後再乘上二分之一，如下公式。 $\text{Darkness\_Threshold}$

$$\text{Darkness\_Threshold} = \frac{1}{2} \times \frac{1}{N} \sum_{i \in L \cap S} X_i$$

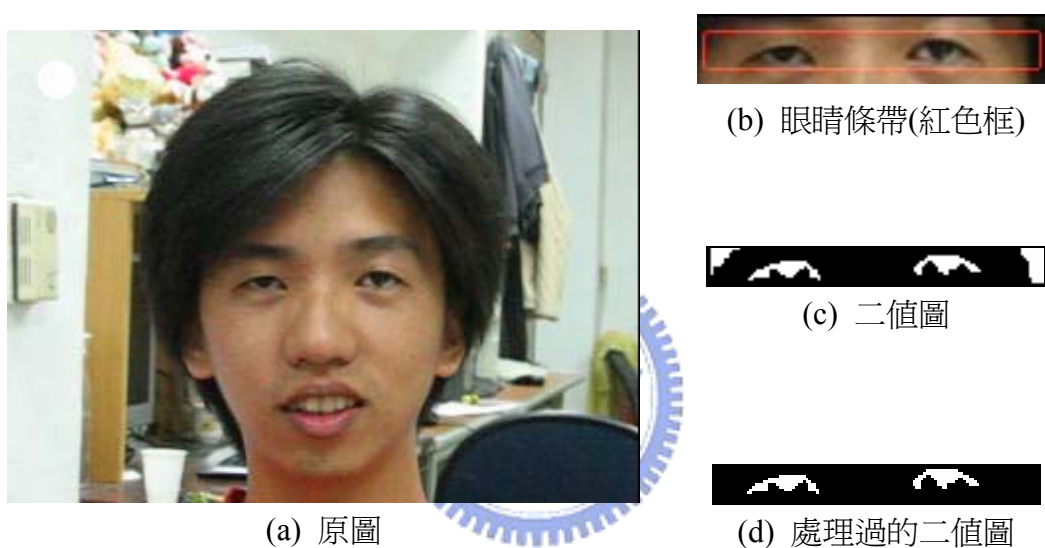
$X_i$  = gray level of pixel  $i$ ,  $N = |L \cap S|$

$L$  = pixels in face block,  $S$  = skin-color pixels

眼睛條帶中有些在邊緣的頭髮部分，在不同的畫面中會因為偵測



到的邊界有些微誤差而使這些邊緣的部分大小有所變動，由於本論文將以眼睛條帶內黑色點數的變化做為偵測眨眼的根據，邊界上黑色點的變化將影響到偵測的結果，所以需要去除頭髮的影響。頭髮的部分至少會連到上邊界或下邊界的其中一個，我們掃描二值圖的每一垂直線，當在上下邊緣有白點時，移除連續的白點，結果如圖十(d)。



圖十 人眼區域統計。

### 3.3 點數變化

將上一節處理後的眼睛條帶，計算白點的個數，對於連續的畫面做出點數的統計圖，如圖十一所示。相當清楚地，在眨眼時點數會有急劇的變化，此變化量我們以深度(Deep)來做測量。對於每個局部最

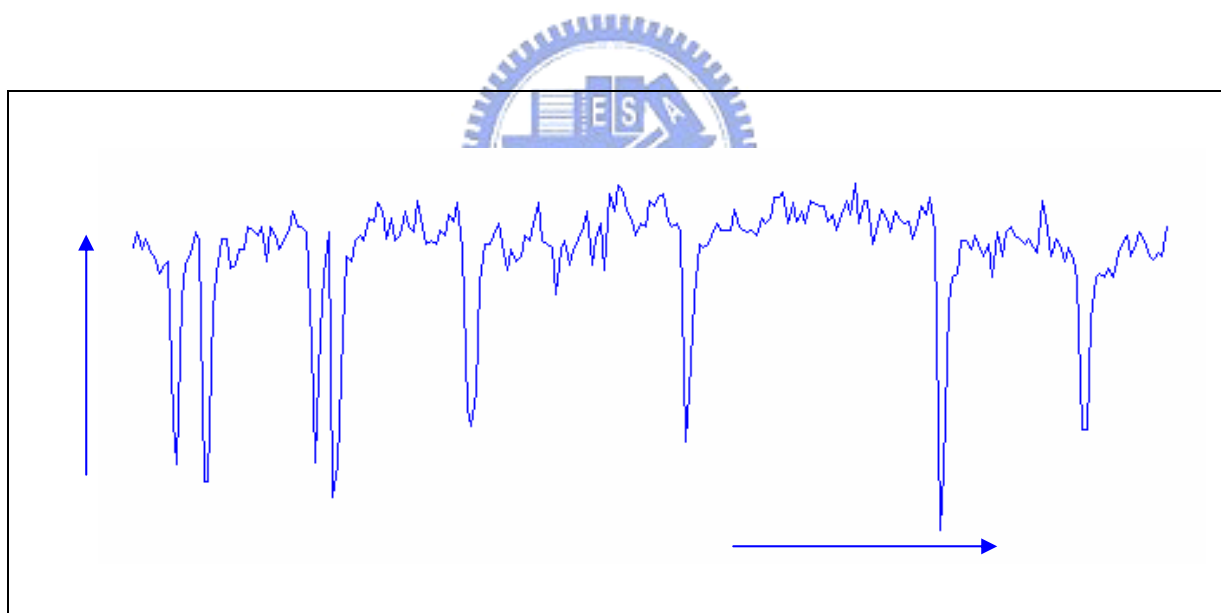
小值(Local minimum)，我們尋找其左邊最接近的局部最大值(Local maximum)和右邊最接近的局部最大值，二個局部最大值之間的區間稱為一個山谷(Valley)。對每個山谷計算左邊的落差和右邊的落差，取較大者定義為此山谷的深度，如下公式所示。

$$Deep = \max(V_{left-max} - V_{min}, V_{right-max} - V_{min})$$

$V_{min}$  : the local minimum

$V_{left-max}$  : the nearest local maximum in the left of  $V_{min}$

$V_{right-max}$  : the nearest local maximum in the right of  $V_{min}$



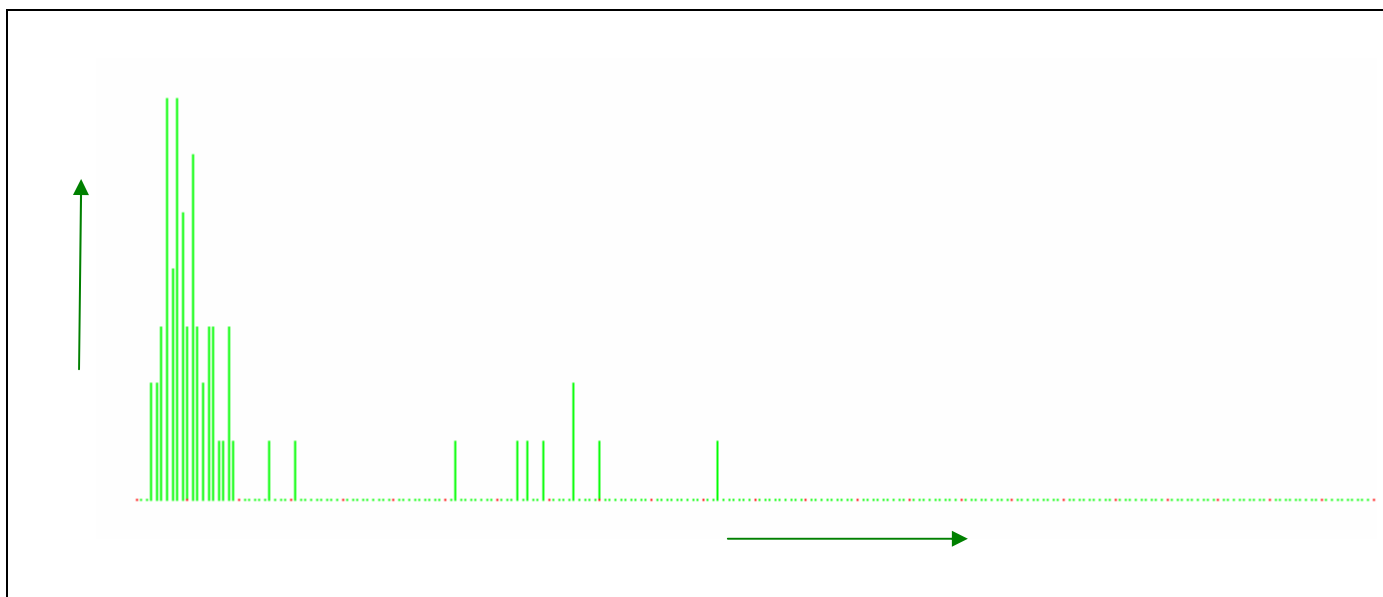
圖十一 連續時間內眼睛條帶點數的變化。本圖為 233 張畫面的統計，其中受測者有 8 次眨眼，即圖中有較大落差的八個位置

以山谷為單位，若深度大於一個門檻值，我們將整個山谷所代表

的時間內標記為眨眼。

### 3.4 眨眼深度門檻值

上一節用來分別眨眼的眨眼深度門檻值，可以由一小段影片訓練出。首先我們計算訓練資料中，眼睛條帶內白點數的變化，對每一山谷的深度做出統計圖，如圖十二。



圖十二 眨眼深度統計圖。

由於睜開眼睛時，眼睛條帶內點數變化量不大，在深度統計圖的低數值區會有密集的区域，在高數值區的稀疏区域則代表眨眼時的數值。我們使用 K 平均值分群法(K-mean Clustering Algorithm)或稱為最近平均值分群法(Nearest Mean Reclassification Algorithm)將深度分為

二群[19]，詳細流程如下。

步驟一： 選擇一個初始門檻值(Initial Guess)  $T$

步驟二： 以  $T$  將深度分為二類，小於  $T$  的為第一類，計算其平均值  $M_1$ ；大於  $T$  的為第二類，計算其平均值  $M_2$ 。

步驟三： 計算新門檻值  $T_{\text{new}} = (M_1 + M_2) / 2$

步驟四： 如果  $T_{\text{new}}$  和  $T$  相等，則演算法結束， $T$  為結果值

如果  $T_{\text{new}}$  和  $T$  不相等，將  $T$  設為  $T_{\text{new}}$  的值重新執行步驟二



# 第四章

## 實驗結果

本論文的實驗方式為使用攝影機取像，錄成一段影像檔後再做分析。影像檔的格式為每秒 15 張畫面，320x240 像素。測試資料庫包含 4 段影片，資料如下表。

影片	畫面數(張)	眨眼次數(次)
受測者 1	2692	46
受測者 2	195	6
受測者 3	232	8
受測者 4	202	9
受測者 5	742	13

測試使用 AMD 1.8G 的電腦，以 Java 實作，執行速度為每秒運算 4 張。

首先依第二章的流程對每張畫面選出人眼區域，再依第三章的方法計算人眼區域內眼數變化的深度，我們取影片的前 10 秒鐘來訓練眨眼深度的門檻值，再對全部影片套用本論文的演算法偵測眨眼。下表為偵測的正確率。

影片	偵測正確數/總數 (正確率)	非眨眼誤判 為眨眼個數
受測者 1	45/46 (97.8%)	0
受測者 2	5/6 (83.3%)	0
受測者 3	8/8 (100%)	0
受測者 4	9/9(100%)	0
受測者 5	12/13(92.3%)	1



# 第五章

## 結論

偵測眨眼可以計算眨眼頻率、閉眼比率，作為疲勞程度的指標。一個快速的眨眼偵測演算法可以使後續的工作有更充足的時間。

本論文提出一個人眼定位及眨眼偵測的演算法，在人眼定位方法中，我們以膚色資訊配合形態學運算找出人臉位置，再以水平投影克服眼鏡和閉眼的問題，有效率地成功找到眼睛。在眨眼偵測方法中，對人眼定位找出的區域，基於自動計算出的門檻值做判斷，可快速的偵測出受測者是否有眨眼。由實驗結果顯示，此演算法判斷眨眼的準確率高，而且時間複雜度低，未來可用於即時運算的偵測系統。

## 參考文獻

- [1] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face Detection in Color Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 696-706, 2002.
- [2] Z. Zhu, Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Computer Vision and Image Understanding*, Vol. 98, Issue 1, pp. 124-154, April, 2005.
- [3] Q. Ji, "3D Face pose estimation and tracking from a monocular camera," *Image and Vision Computing*, Vol. 20, Issue 7, pp. 499-511, May 1, 2002.
- [4] S. Kawato, N. Tetsutani, "Detection and tracking of eyes for gaze-camera control," *Image and Vision Computing*, Vol. 22, Issue 12, pp. 1031-1038, October 1, 2004.
- [5] X. Xie, R. Sudhakar, and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, Vol. 27, pp. 791-799, 1994.
- [6] M. Kampmann, L. Zhang, "Estimation of eye, eyebrow and nose features in videophone sequences," *Int. Workshop on Very Low Bitrate Video Coding (VLBV 98)*, Urbana, USA, 1998.
- [7] G.C. Feng, P.C. Yuen, "Variance projection function and its application to eye detection for humanface recognition," *Int. Conf. Computer Vision*, Vol. 19, pp. 899-906, 1998.
- [8] T.N. Bhaskar, F.T. Keat, S. Ranganath, and Y.V. Venkatesh, "Blink detection and eye tracking for eye localization," *TENCON 2003 Conf. on Convergent Technologies for Asia-Pacific Region*, Vol. 2, 15-17, pp. 821-824, October, 2003.
- [9] S. Kawato, J. Ohya, "Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes," *Proc. 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 40-45, 2000.
- [10] S. Kawato, N. Tetsutani, "Real-time detection of between-the-eyes with a circle frequency filter," *Proc. ACCV2002*, Vol. 1, pp. 442-447, 2002.



- [11] S. Kawato, J. Ohya, "Two-step approach for real-time eye tracking with a new filtering technique," 2000 IEEE Int. Conf. on Systems, Man, and Cybernetics, Vol. 2, 8-11, pp. 1366-1371, October 2000.
- [12] Q. Ji, X. Yang, "Real time visual cues extraction for monitoring driver vigilance," Proc. Int. Workshop on Computer Vision Systems, Vancouver, Canada, 2001.
- [13] Y. Ebisawa, S. Satoh, "Effectiveness of pupil area detection technique using two light sources and image difference method," Proc. the 15<sup>th</sup> Annual Int. Conf. of the IEEE Eng. In Medicine and Biology Society, San Diego, CA, pp. 1268-1269, 1993.
- [14] Y. Ebisawa, "Improved video-based eye-gaze detection method," IEEE Trans. Instrument. Meas. 47 (2), pp. 948-955, 1998.
- [15] C. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," Technical Report RJ-10117, IBM Almaden Research Center, 1998.
- [16] C. Morimoto, M. Flickner, "Real-time multiple face detection using active illumination," Proc. fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition 2000, Grenoble, France, 2000.
- [17] A. Haro, M. Flickner, and I. Essa. "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," Proc. IEEE CVPR 2000, Hilton Head Island, South Carolina, 2000.
- [18] R. Bartoszynski and M. Niewiadomska-Bugaj, Probability and Statistical Inference, Wiley Inter. Science.
- [19] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2<sup>nd</sup> Edition, Academic Press.
- [20] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2<sup>nd</sup> Edition, Prentice Hall.
- [21] T. Kawaguchi, D. Hidaka, and M. Rizon, "Detection of Eyes from Human Faces by Hough Transform and Separability Filter," Proc. Int. Conf. on Image Processing, Vol. 1, 10-13, pp. 49-52, September 2000.