

# 第一章

## 緒論

### 1.1 研究背景

跟隨著電腦的日益進步，電腦已經是我們日常生活中不可或缺的工具。不論是交通、工程、經貿、通訊或休閒等方面，電腦實際上提供了非常大的幫助。而今，我們身處在資訊爆炸的時代電腦可以說扮演了非常重要的角色。

因此，我們希望電腦不僅能處理簡單、複雜或重複的計算工作之外，我們更希望它能幫助人類作各項的工作或從旁協助提醒人類作一些電腦無法取代的智慧性的工作；甚至如果可能的話要做到人類與電腦之間友善的溝通，使電腦了解每個人都有不同的情緒；依此作出適當的反應。而現今電腦能夠辨別誰是誰已是目前一項非常重要的課題。如果電腦內部的資料庫儲存著每一個人的臉部影像，而又可以辨識每個人臉部的器官時，電腦便能由人的五官判斷出身份，而且也可由此知道此人目前的精神狀況。理所當然的，如果電腦可以清楚辨識出眉毛、眼睛、鼻子和嘴巴時，電腦也將會有更寬廣的應用空間。

例如，我們可舉幾個例子來看：

1. 我們可以讓電腦經由人臉部五官的位置判斷人的喜、怒、哀、樂等情緒變化，進而增進電腦跟人類之間友善的溝通。然而這個領域已經普遍的被許多人所研究討論，甚至使用不同的方法來辨識人的情緒變化，例如：Charles Darwin [1]、Ekman [2], [3]、Morishima and Harashima [4]、Mase [5]、Kobayashi and Hara [6]-[9]、Rosenblum 等人[10]、Yuille 等人[11], [12]、Shackleton and Welsh [13]、Xie, 等人[14]。
2. 我們也可由人的眼睛大小變化判斷其精神狀況。若將此項技術運用於捷運、小客車、大貨車、計程車駕駛，只要是人操作的交通工具。我們就可以使用這樣的應用來提醒駕駛人是否已達到該休息的時機了。

3. 枯燥乏味的重複性動作，但卻又要人的智慧作判斷時。如生產線的小姐她的工作是枯燥乏味且重複性的動作，當她完成一項工作而需要她做其他工作時；機器人便無法取代作業員。也因為如此我們需要藉由電腦來判斷操作者的精神狀況，進而提醒作業員以免鑄成大錯。
4. 甚至我們可以廣泛的應用到由電腦前的 CCD 攝影機輸入影像後，再作比對的工作來判斷此人是否認真的工作或已經昏昏欲睡了。此時只要有這些提醒的裝置我們相信就可以減少一些不必要的錯誤產生。
5. 當電腦可以識別人臉時，電腦也可以作大樓的保全系統，掌控進出大樓人員的出入。

而這些想法都可以有無限的想像空間，進而實現這些想法。只要我們能夠找到一個有效率又準確的辨識眼睛的方法，我們便能將以上的想法付諸實現了。然而辨識臉部的方法有很多，於是我們挑選下面幾種方法來做評估與比較。

一是 RCER 配合變形樣本；二是邊緣檢測。而在邊緣檢測中我們選擇了 MATLAB 中提供的幾種演算法，就是 Sobel、Prewitt、Canny 三種邊緣檢測的方法，另外再使用 MATLAB 中沒有提供的 Frei-Chen 的演算法，總共五種方法去做評估與實驗。

本來 RCER 跟變形樣本是來判斷人情緒變化的演算法，但在做了實驗與比較後發現它的準確率並不低於邊緣檢測的演算法，於是把它跟邊緣檢測做一比較。

## 1.2 論文大綱

這篇論文共分為五個章節。在第一章節裡先介紹整個論文研究的動機，並介紹本篇論文使用的技術、原理概論。並於第一章節的最後將整個研究的流程作一個流程圖的說明，如圖 1.1 所示。

第二章節裡將介紹本篇論文中會使用到的演算法跟技術方面上的基本概念，包含了數學形態學裡的集合論、膨脹(dilation)、腐蝕(erosion)、開啟(opening)、閉合(closing)、擴展至灰階影像(Extensions to Gray-Scale Images)

等技術概念跟 RCER (Rough Contour Estimation Routine) 粗略輪廓預測副程式以及“連續物件區域搜尋”(Contiguous Object Region Finding)演算方法的介紹。

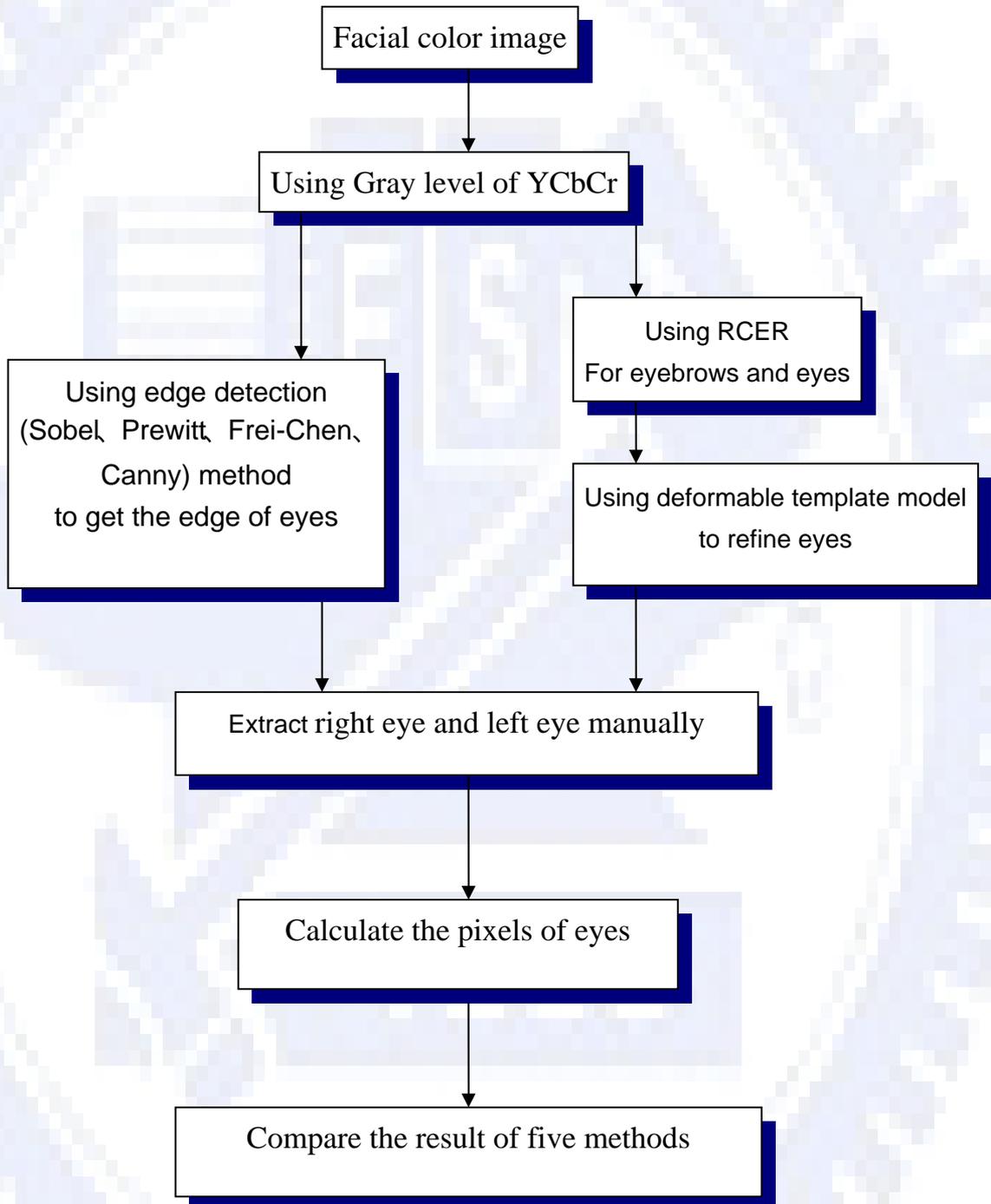


圖 1.1 本篇論文的设计流程

在第三章裡因為有了第二章的基本概念後，我們將把在這論文裡會用到的幾種技術與演算法逐一作較詳細的說明，首先對變形樣本(deformable template model)的理論跟演算法作敘述，接著介紹邊緣搜尋演算法的基本演算；然後介紹比較相近的 Sobel 跟 Prewitt 的演算法，然後再介紹改良上述方法的 Frei-Chen 的演算法，最後談到的是將邊緣檢測作一總合討論的 Canny 演算法的理論與技術。

在第四章節裡，我們將所使用到的五種眼睛檢測的方法，對我們所建立人臉影像庫做眼睛的檢測。首先我們利用 YCbCr 將影像資料中的影像取出 Y 的成份，並且將它的眼睛跟眼珠部分以電腦描繪出來後並作為 TRUTH GROUND；然後再用第三章裡提到的技術分別把眼睛給檢測出來，並描繪出影像庫中十張人臉的眼睛及眼珠的輪廓與位置。最後將各種技術所描繪出來的眼睛及眼珠與 TRUTH GROUND 分別作一比較，並將其結果留在本章的最後面提出。

第五章裡將作最後的討論與檢討，也將我們所發現的一些問題逐一做個檢討；以作為未來工作的借鏡，以免重蹈覆轍。

### 1.3 研究理論大綱

本篇論文最主要就是想應用電腦來做 1.1 節中所提到的一些應用工作的前置作業，所以我們將專注於人臉部中最重要的器官-眼睛-來做探討。我們選用的第一類方法就是 RCER(Rough Contour Estimation Routine)[15]，配合使用變形樣本模型(Deformable Template)[16]；第二類方法就是利用邊緣檢測(Edge Detection)的遮罩(template、mask)來做研究討論。

人臉的辨識是屬於影像處理的一部分。而處理人臉辨識的話，邊緣檢測就是在影像處理中一項非常重要的技術。邊緣是影像中最基本的特徵，所以我們一定知道何謂邊緣？邊緣是指影像中的灰階值發生急劇變化的像素的集合。Poggio [17]等在其研究中提到：「邊緣或許對應著影像中物體的邊界或許並沒有對應著影像中物體的邊界，但是邊緣具有十分令人滿意的特性，它能大大地減少所要處

理的資訊量但是又保留了影像中物體的形狀資訊」。邊緣檢測：「為主要是影像灰階值變化的度量、檢測和定位」。邊緣與影像中物體的邊界有關但又不是完全相同；邊緣反映的是影像灰度值的不連續性[18]。

邊緣在邊界檢測、影像切割、模型識別、機器視覺等中都有其重要的作用。邊緣是邊界檢測的重要基礎，也是外形檢測的基礎[19]。邊緣廣泛存在於物體與背景；物體與物體；像素與像素之間，因此它也是影像切割所依賴的重要特徵。

所以當我們要檢測人臉部的器官時，邊緣檢測就是我們首選的方法，而主要有幾個原因：

1. 人眼通過追蹤未知物體的輪廓(輪廓是由一段一段的邊緣所組合而成的)，而得知該物體是為何物。
2. 如果我們能得到影像的邊緣，那麼影像分析就會被大大的簡化，影像識別就會變的容易許多。
3. 很多的影像並沒有實質具體的物體，對這些影像的理解將取決於它們的紋理性質，而提取這些紋理性質則與邊緣檢測有極其密切的關係 [20]。

關於邊緣檢測的方法主要有以下幾類：

1. 計算梯度的最大值：因為邊緣發生在影像灰階值變化急劇的地方，所以對應連續的情形就是說函數梯度值比較大的地方。代表的運算法有Robert、Prewitt、Sobel、Frei-Chen，... 等。
2. 檢測二階導數的零交叉點：因為邊緣處的梯度取得的最大值；可能是正的也可能是負的，所以此處便是灰階影像的邊緣。
3. 統計型方法：比如說利用假設檢驗來檢測邊緣 [21]，Marimont在 [22] 中利用對二階零交叉點的統計分析得到了影像中各個像素是邊緣的機率，並進而得到邊緣檢測的方法。
4. 小波邊緣檢測：九十年代，隨著小波分析的迅速發展，小波開始用於邊緣檢測。 [23]、[24]。
5. 其他：例如模糊數學的方法，最近提出來的利用邊緣流(Edge flow)來檢測邊

緣[25]，基於積分變換的邊緣檢測方法[26]等...方法。

然而儘管檢測邊緣的方法有這麼多，但是卻礙於演算法的複雜度跟準確度以及檢測影像的大小、解析度等原因，我們只選擇了以下幾種方法 Sobel、Prewitt、Frei-Chen 跟 Canny 四種演算法來做比較。

## 第二章

### 基礎概念 (研究內容與方法)

#### 2.1 數學形態學 (Morphology)

數學形態學是一種應用於影像處理(image processing)和模型識別(model recognition)領域的一種方法，基本的想法是用具有一定形態的架構元素去度量和檢測影像中的對應形狀以達到對影像分析和識別的目的。用於描述數學形態學的語言是集合論，因此對集合論要有一定的了解。它可以用一個統一而強大的工具來處理影像處理中所遇到的問題。利用數學形態學對物體幾何架構的分析過程就是主客體相處逼近的過程。利用數學形態學的幾個基本概念和運算，將架構元素靈活地組合分解，應用形態變換達到分析的目的。

數學形態學方法比其他空間域或頻率域影像處理和分析方法，具有一些明顯的優勢。如在影像恢復處理中，基於數學形態學的形狀濾波器可借助於先前的幾何特徵訊息，利用形態學運算元有效的濾除雜訊，又可以保留影像中的原有訊息；另外，數學形態學演算法易於用平行處理的方法來有效地實現，而且硬體實現也比較容易；基於數學形態學的邊緣訊息檢測處理優於基於微分運算的邊緣檢測演算法，它不像微分演算法對雜訊那樣地敏感，同時檢測的邊緣也比較平滑，利用數學形態學方法檢測影像的輪廓也比較連續、斷點也比較少。

集合論是數學形態學的基礎，首先我們對集合論的一些基本概念作一個概括性的介紹。

集合(集)：具有某種性質的、確定的有區別的事物的全體（它本身也是一個事物）。常用大寫字母如： $A$ 、 $B$ 、 $\dots$ 表示。如果某種事物不存在，就稱這種事物的全體是空集合。規定任何空集合都只是同一個集合，記為 $\varphi$ 。在以下的介紹中設 $A, B, C$ 等都是歐幾里德空間 $E^N$ 空間的集合。

1. 元素：構成集合的每個事物。常用小寫字母如  $a$ 、 $b$ 、 $\dots$ 表示。任何事物都不是  $\varphi$  中的元素。
2. 子集合：若且為若集合  $A$  的元素都屬於集合  $B$  時，稱  $A$  為  $B$  的子集合。
3. 聯集：若  $A$  和  $B$  是集合，則  $A$  和  $B$  聯集是有所有  $A$  的元素和所有  $B$  的元

素，而沒有其他元素的集合。A 和 B 的聯集通常寫作 “ $A \cup B$ ”。

4. 交集：兩個集合 A 和 B 的交集是含有所有既屬於 A 又屬於 B 的元素，而沒有其他元素的集合，A 和 B 的交集通常寫作 “ $A \cap B$ ”。
5. 補集合(Complement)：A 和 B 是集合，則 A 在 B 中的補集合，是這樣一個集合，其元素屬於 B，但不屬於 A。A 在 B 中的相對補集合通常寫作  $B - A$  (或  $B \setminus A$ )。
6. A 的補集合，記為  $A^c$ ，定義為：

$$A^c = \{x | x \notin A\} \quad (2.1)$$

7. 位移 (Translation)：A 用  $x=(x_1, x_2)$  位移，記為  $(A)_x$ ，定義為：

$$(A)_x = \{c | c = a + x, \text{ for } a \in A\} \quad (2.2)$$

8. 映射 (Reflection)：A 的映射，記為  $\hat{A}$ ，定義為：

$$\hat{A} = \{x | x = -a, a \in A\} \quad (2.3)$$

9. 差集(Difference)：兩個集合 A 和 B 的差，記為  $A - B$ ，定義為：

$$A - B = \{x | x \in A, x \notin B\} = A \cap B^c \quad (2.4)$$

## 2.1.1 Dilation and Erosion

### 1. 膨脹(Dilation)

A, B 為  $Z^2$  中的集合， $\varphi$  為空集合，A 被 B 的膨脹，記為  $A \oplus B$ ， $\oplus$  為膨脹運算元，膨脹的定義為：

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \emptyset\} \quad (2.5)$$

該式表明的膨脹過程是 B 首先做關於原點的映射，然後平移 x，A 被 B 的膨脹過程是  $\hat{B}$  所有 x 以後至少有一個非零的公共元素。

在其他的形態處理中一樣，集合 B 在膨脹操作中通常被稱為架構元素。

膨脹運算圖解：

圖 2.1(a). 表示一個簡單的集合 A，圖 2.1(b). 表示一個架構元素及其“映射”。在此圖情況下，因為元素 B 關於原點對稱，所以，結構元素 B 及其映射  $\hat{B}$  相

同。圖 2.1(c). 中的陰影表示作為參考的原始集合；實線標示出若  $\hat{B}$  的原點平移至  $x$  點超過此界限，則  $\hat{B}$  與  $A$  的交集為空，這樣實線內所有點構成了  $A$  被  $B$  的膨脹。

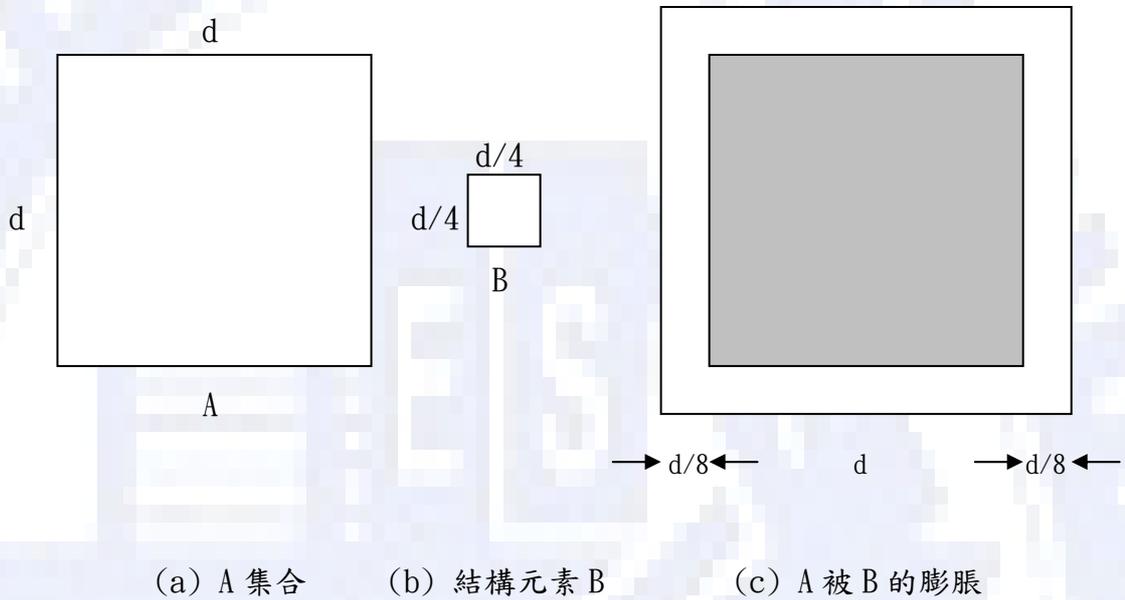


圖 2.1 膨脹運算圖解

## 2. 腐蝕(Erosion)

$A, B$  為  $Z^2$  中的集合， $A$  被  $B$  腐蝕，記為  $A \ominus B$ ，其定義為

$$A \ominus B = \{x | (B)_x \subseteq A\} \quad (2.6)$$

也就是說  $A$  被  $B$  腐蝕的結果為所有使  $B$  被  $x$  平移後包含於  $A$  點的集合。

腐蝕運算圖解：

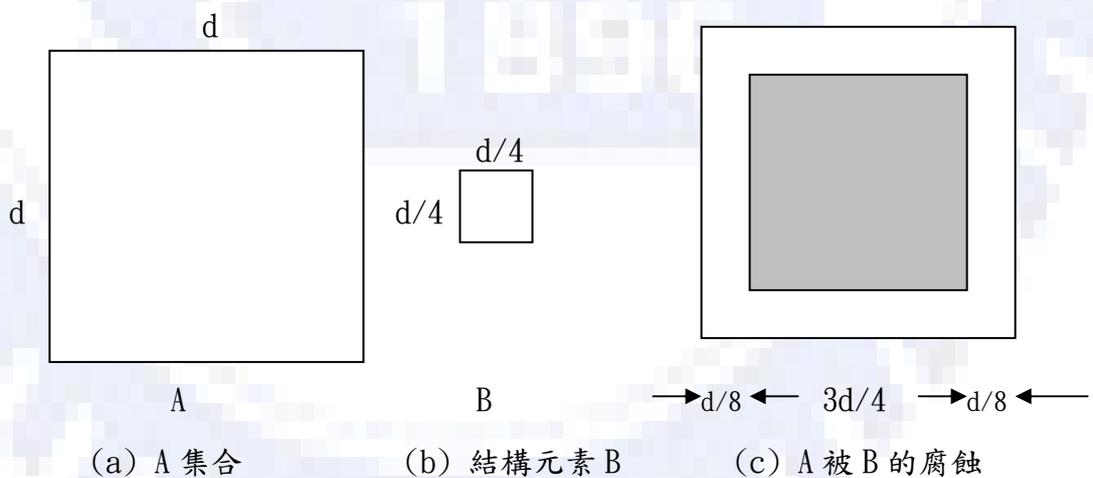


圖 2.2 腐蝕運算圖解

類似於圖 2.1，集合 A 在圖 2.2(c). 用陰影表示作為參考。實線表示若 B 的原點平移至 x 點超過此界限，則 A 不能完全包含 B。在這個實線邊界內的點構成了 A 被 B 的腐蝕。

膨脹和腐蝕是關於補集合和反轉的對偶，也就是， $(A-B)^c = A^c \oplus B^c$ 。

## 2.1.2 Opening and Closing

### 開運算和閉運算(Opening and Closing)

開運算一般能平滑影像的輪廓，削弱狹窄的部分，去掉微小的凸出。閉運算也是平滑影像的輪廓，與開運算相反，它一般能融合窄的缺口和細長的轉彎口，去掉小洞，填補輪廓上的縫隙。

圖 2.3(a). A 是原始影像，圖 2.3(b). B 是結構元素影像，則集合 A 被結構元素 B 做開運算，記為  $A \circ B$ , 其定義為：

$$A \circ B = (A \ominus B) \oplus B \quad (2.7)$$

就是 A 被 B 腐蝕後的結果再被 B 膨脹。

設 A 是原始影像，B 是結構元素影像，則集合 A 被結構元素 B 做閉運算，記為  $A \bullet B$ , 其定義為：

$$A \bullet B = (A \oplus B) \ominus B \quad (2.8)$$

就是 A 被 B 膨脹後的結果再被 B 腐蝕。

圖 2.4 和圖 2.5 表示了集合 A 被一個圓盤形架構開運算和閉運算的情況。圖 2.3(a). 表示集合 A，圖 2.3(b). B 是結構元素影像，圖 2.4(a). 表示腐蝕過程中圓盤形架構元素的各個位置，當完成這一過程時形成分開的兩個圖形表示於圖 2.4(b). 注意，A 的兩個主要部分之間的橋樑被去掉了。“橋”的寬度小於架構元素的直徑，也就是架構元素不能完全包含於集合 A 的這一部分。同樣，A 的最右邊的部分也被切掉了。圖 2.4(c). 給出了對腐蝕的結果進行膨脹的過程。圖 2.4(d). 表示出了開運算的最後結果。同樣，圖 2.5(a)~(d) 示出了用同樣的架構元素對 A 做閉運算的結果。結果去掉了 A 左邊對於 B 來說是較小的轉彎。用同一個圓形的

架構元素對集合 A 作開運算和閉運算，使 A 的一些部分平滑。

關於開運算和閉運算的幾何解釋，把圓盤形結構元素 B 看作一個（平面的）滾動球， $A \circ B$  的邊界為 B 在 A 內滾動所能達到的最遠處的 B 的邊界所構成， $A \bullet B$  的邊界為 B 在 A 外滾動所能達到的最遠處的 B 的邊界所構成。

開運算和閉運算是關於補集合和反轉的對偶，表示式如下：

$$(A \bullet B)^c = A^c \circ B^c \quad (2.9)$$

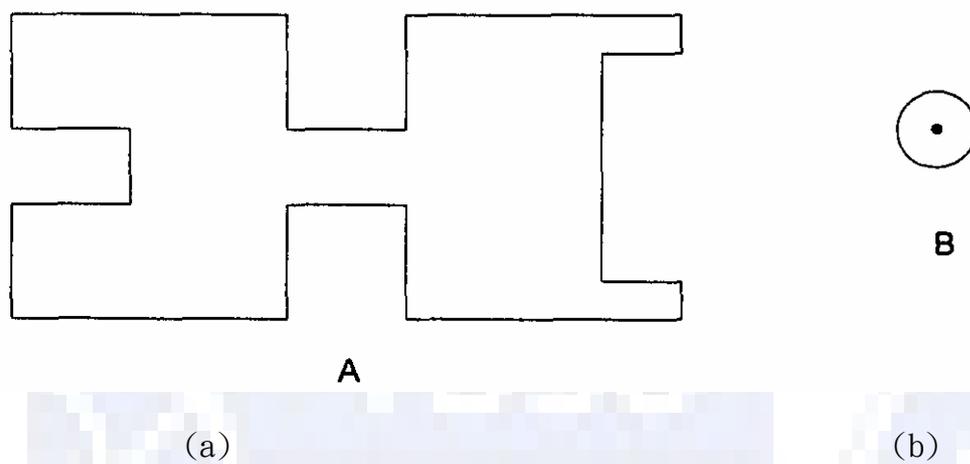


圖 2.3 原始影像跟結構元素(a) A 集合 (b) 結構元素 B

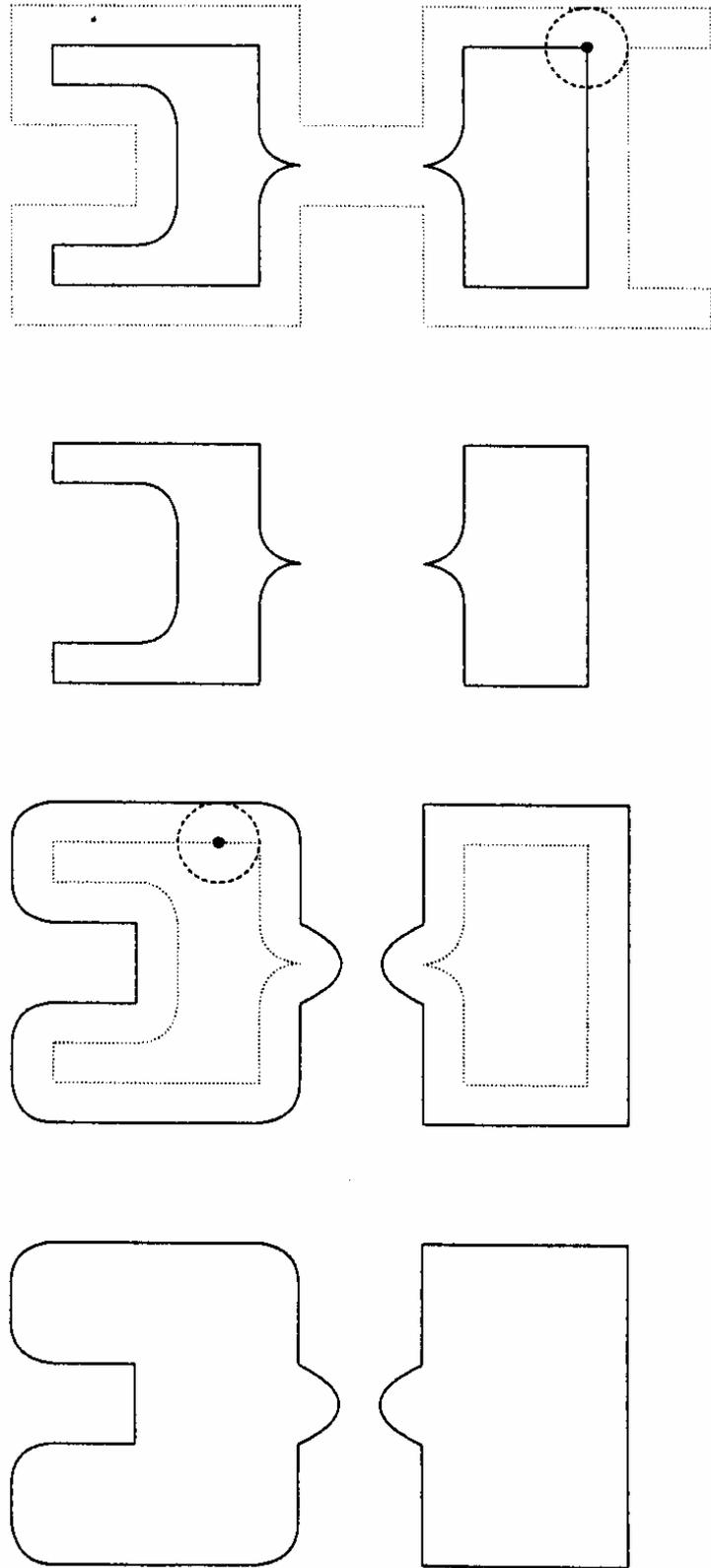


圖 2.4 開啟(Opening)運算圖解(a)、(b)、(c)、(d)

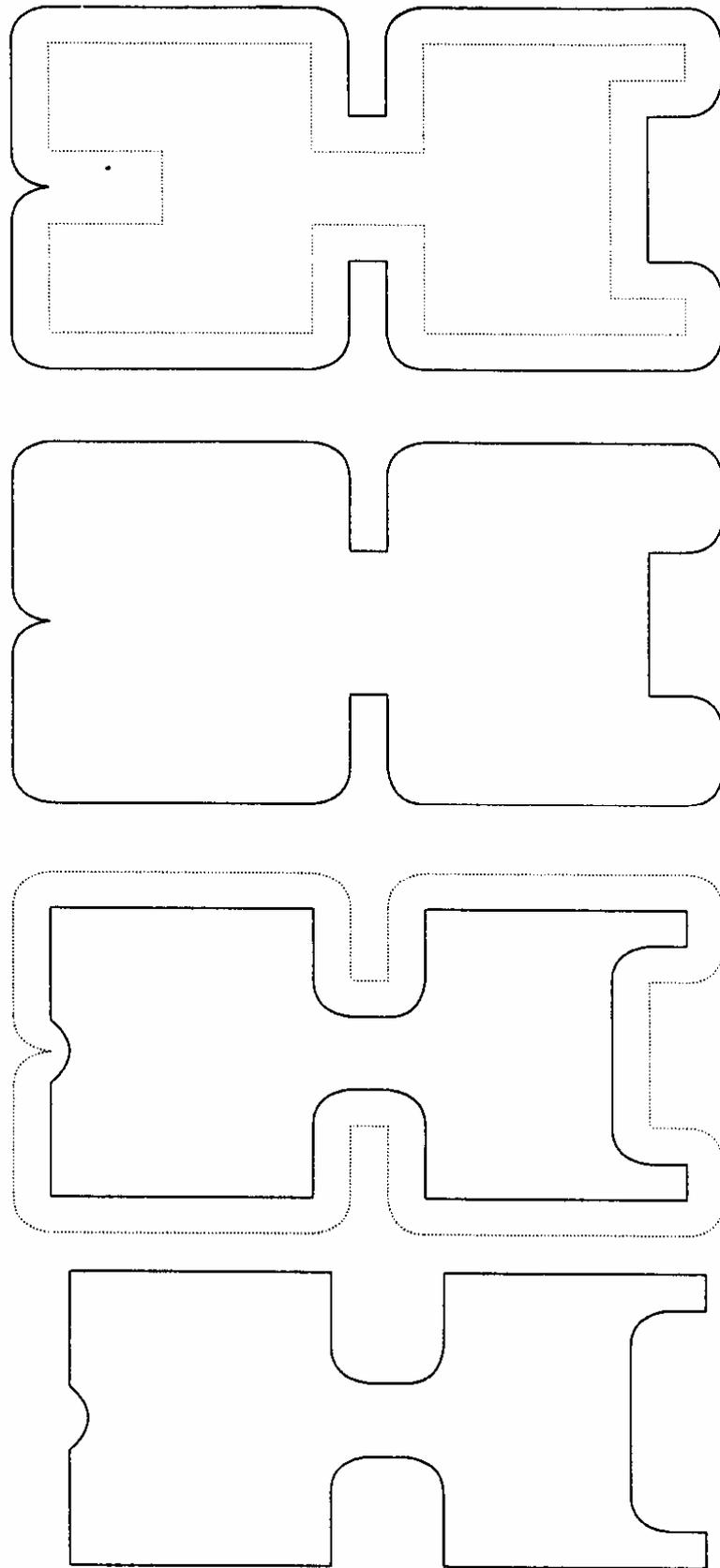


圖 2.5 關閉(Closing) 運算圖解(a)、(b)、(c)、(d)

## 2.1.3 Extensions to Gray-Scale Images

將二值數學形態學推展至灰階形態學，與前面不同的是以下的討論中將處理數學影像函數而不是集合。設  $f(x, y)$  是輸入影像， $b(x, y)$  是架構元素，它可被看作是一個子影像函數。如果  $Z$  表示實整數集合，同時假設  $(x, y)$  是來自  $Z * Z$  的整數， $f$  和  $b$  是對坐標為  $(x, y)$  像素灰階值的函數。如果灰階值也是整數，則  $Z$  可由整數  $R$  所代替。

### 1. 膨脹

用結構元素  $b$  對輸入影像  $f$  進行灰階膨脹記為  $(f \oplus b)$ ，其定義為：

$$(f \oplus b)(s, t) = \max\{f(s-x, t-y) + b(x, y) \mid (s-x, t-y) \in D_f \text{ 和 } (x, y) \in D_b\} \quad (2.10)$$

其中  $D_f$  和  $D_b$  分別是  $f$  和  $b$  的定義域。這裡限制  $(s-x)$  和  $(t-y)$  在  $f$  的定義域之內，類似於在二值膨脹定義中要求 2 個運算集至少有 1 個（非零）元素相交。

灰階膨脹：

膨脹的計算是在由架構元素確定的鄰域中選取  $(f \oplus b)$  的最大值，所以對灰度影像的膨脹操作有兩種效果：第一、如果架構元素的值都為正的，這輸出影像會比輸入影像亮；第二、根據輸入影像中暗的細節的灰度值以及它們的形狀相對於架構元素的關係，他們在膨脹中或被消滅或被除掉。

### 2. 腐蝕

用架構元素  $b$  對輸入圖像  $f$  進行灰階腐蝕記為  $f \ominus b$ ，其定義為：

$$(f \ominus b)(s, t) = \min\{f(s+x, t+y) - b(x, y) \mid (s+x, t+y) \in D_f \text{ 和 } (x, y) \in D_b\} \quad (2.11)$$

其中  $D_f$  和  $D_b$  分別是  $f$  和  $b$  的定義域。這裡限制  $(s-x)$  和  $(t-y)$  在  $f$  的定義域之內，類似於在二值腐蝕定義中要求架構元素完全包括在被腐蝕集合中。

灰階腐蝕：

腐蝕的計算是在由架構元素確定的鄰域中選取  $f-b$  的最小值，所以對灰階影像的腐蝕操作有兩種效果：第一、如果架構元素的值都為正的，這輸出圖像會比輸入影像暗；第二、如果輸入影像中亮的細節的尺寸比架構元素小，則其影響會被減弱，減弱的程度取決於這些亮細節周遭的灰階值和架構元素的形狀和振幅值。

### 3. 開啟和閉合

灰階數學形態學中關於開啟和閉合的表達與它們在二值數學形態學中的對應

運算是一致的。

$$\text{用 } b \text{ 開啟 } f \text{ 記為 } f \circ b, \text{ 其定義為: } f \circ b = (f \ominus b) \oplus b \quad (2.12)$$

$$\text{用 } b \text{ 閉合 } f \text{ 記為 } f \bullet b, \text{ 其定義為: } f \bullet b = (f \oplus b) \ominus b \quad (2.13)$$

灰階開啟消除了尺寸較小的亮細節，灰階閉合消除了尺寸較小的暗細節。

## 2.2 RCER (Rough Contour Estimation Routine) 粗略輪廓預測副程式

在一幅表面影像內的每個器官的粗糙的輪廓必須透過估計而被捕獲特徵的輪廓。粗糙的輪廓估計表面器官輪廓 Hwang 和陳的預測副程式，這是相似灰階值的一個像素增長演算法[27].

在利用此種方法檢測眼睛的輪廓之前，我們必須估計眼睛的位置。估計的過程如下：

- (a)、根據我們的觀察，左眉的位置大約面部左邊的 4 分之 1 寬度處。因此，我們能計算  $x$  值，約等於左眉尖的輪廓位置。
- (b)、已經認為左眉的輪廓尖同等  $x$ ，再從發現同等  $y$  的左側的頂端向下搜尋整個左眉。
- (c)、我們再使用那些“連續物件區域搜尋”(Contiguous Object Region Finding) 拿到左眉的粗糙的輪廓。
- (d)、再讓電腦的程式往下搜尋可以從左眉的下方，我們可以發現左眼的輪廓尖。然後我們使用“連續物件區域搜尋”得到左眼的粗糙的輪廓。
- (e)、類似於左邊，我們能得到右邊的粗略的左眉和左眼睛的輪廓。

### 2.2.1 連續物件區域搜尋(Contiguous Object Region Finding)

“連續物件區域搜尋”的目標發現將搜尋連結地區。我們使用演算法得到面部的眼睛和眉毛的粗糙的輪廓。

其演算法由兩個步驟所組成：

1. 搜尋一個連結的區域，並且找到屬於此區域的每一個  $x$  座標上極大和極小的  $y$  座標的點。

2. 結合此區域的每一個  $x$  座標上極大和極小的  $y$  座標的點，就可得到粗糙的輪廓。

我們使用一個  $3 \times 3$  遮罩(mask)搜尋粗略的輪廓。這個遮罩被顯示在圖 2.6 裡，並且以  $(x, y)$  描述重心點。我們使用圖 2.7 跟 2.8 來解釋搜尋的副程序。圖 2.7 是以程式的方式來顯示整個流程；而圖 2.8 則利用一個流程架構圖來表示“連續物件區域搜尋”的方法。而我們使用 RCER 跟“連續物件區域搜尋”可以找出眼睛的粗略輪廓，顯示於圖 2.9

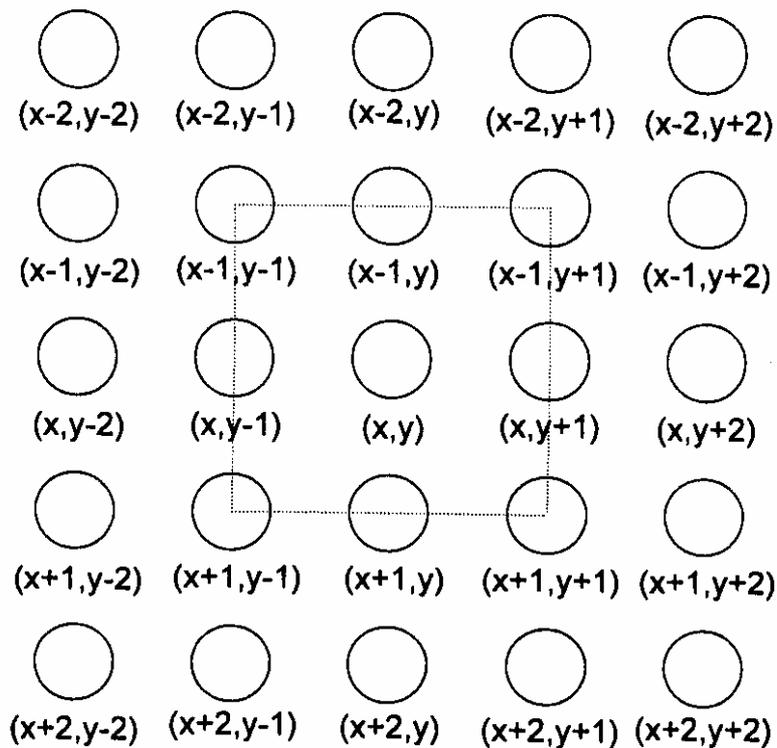


圖 2.6 The searching mask

The searching subroutine is shown in following.

```
search (x, y)
{
clear point $(x, y)
  if ( I(x-1, y-1) <= threshold)
search (x-1, y-1)
  if ( I(x-1, y) <= threshold)
search $(x-1, y)
  if ( I(x-1, y+1) <= threshold)
search $(x-1, y+1)
  if ( I(x, y-1) <= threshold)
search (x, y-1)
  if ( I(x, y+1) <= threshold)
search (x, y+1)
  if ( I(x+1, y-1) <= threshold)
search (x+1, y-1)
  if ( I(x+1, y) <= threshold)
search (x+1, y)
  if ( I(x+1, y+1) <= threshold)
  search (x+1, y+1)
}
```

圖 2.7 搜尋的次程序

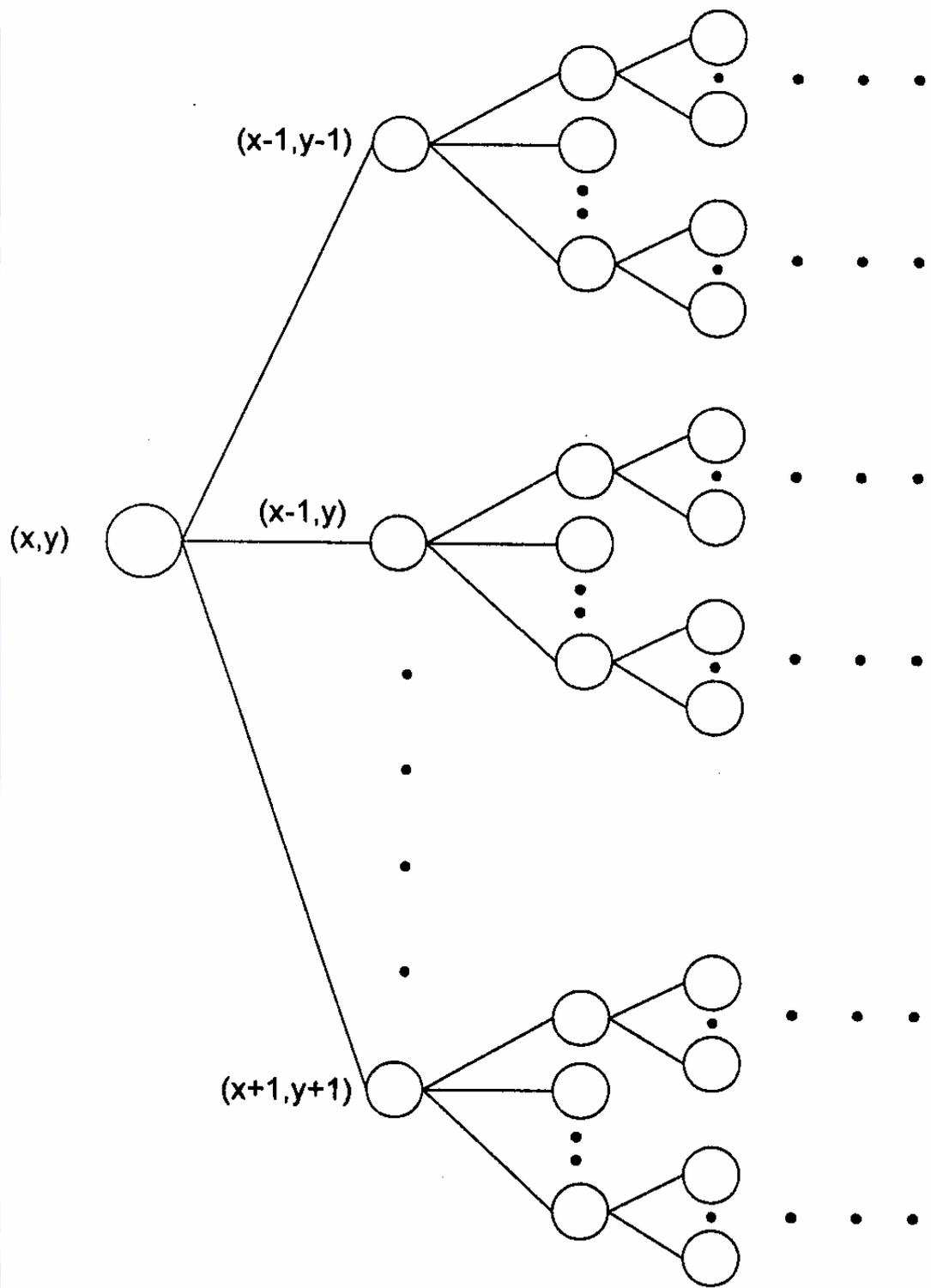


圖 2.8 Using the figure to explain the searching subroutine

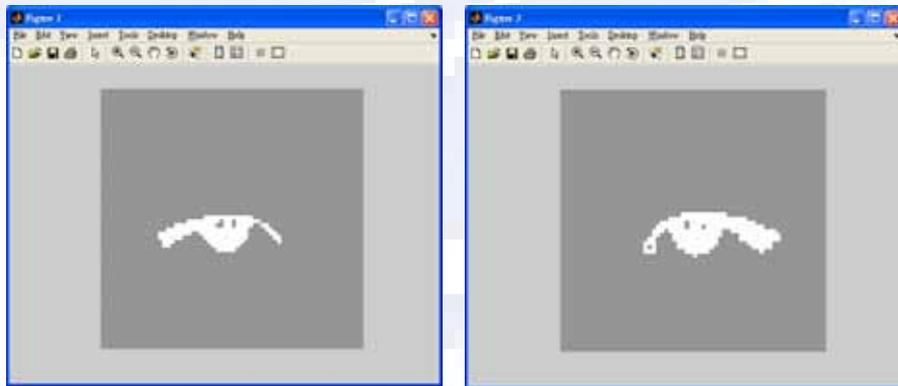


圖 2.9 C 君眼睛的粗略輪廓

## 2.3 邊緣檢測(edge detection)的基本概念

### 邊緣檢測(edge detection)概述：

在本章節中我們將對選用幾種較具代表性的邊緣檢測的方法作為檢測我們眼睛輪廓的方式。在各領域上常用的邊緣檢測我們做一個介紹[28], [29]。基本上我們按照這些邊緣檢測提出的時間順序來作一個概述。由於在1986年Canny[38]提出了邊緣檢測的三準則，對於邊緣檢測也有很大的效率，故我們也將於在最後作一概述。

在檢測整個影像時，影像的物體邊界須先檢測出EDGE和LINE的連接，之後便可以用一些簡單的法則來做以上的工作。例如，先去除獨立而無相鄰邊緣的點和平行的線；將鄰近的EDGE點串成線。

接著便使用 Template [MASK]偵測出EDGE的著名方法，來檢測邊緣。例如：

- (1) Robert [30]
- (2) Kirsch [31]
- (3) Sobel [32]
- (4) Prewitt [33]
- (5) Robinson [34]
- (6) Frei-Chen [35]

以上的方法皆用同樣原理，如果我們用較大的 Template 就可以得到更好的效果，但是卻也花費了更多的電腦計算時間。而Kirsch跟Robinson都是使用羅盤式(compass)的Template來做邊緣檢測，雖然比較準確但是此種方法浪費太多的電腦計算的時間。

所以本篇論文就選擇了Sobel、Prewitt、Frei Chen 三種 Template來做檢測眼睛的方法。

在本章節中我們先作一些基本的定義：

1. 在圖2.10中我們將一個影像稱為：Continuous Image
2. 而內部每一個3X3的矩陣都稱為：Image sub-area
3. 影像內部像素與像素之間就稱為(Sampling intervals)，如圖2.10。

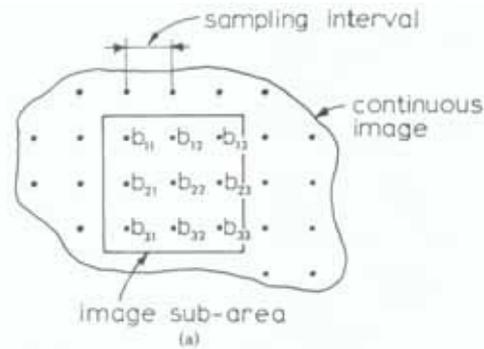


圖2.10 子區域跟像素與像素之間的關係圖

4. 當一個影像的像素之間有很大的明暗差距就稱為邊緣(Edge)，如圖2.11。  
當一個邊界通過一個區域時，我們給一個任意的一條線則可以將它劃分為兩個明暗的區間，此時我們就可以定義出它的大小跟方向。

$$magnitude = |\bar{b}_1 - \bar{b}_2|, \quad 0 \leq \phi_e \leq 2\pi \quad (2.14)$$

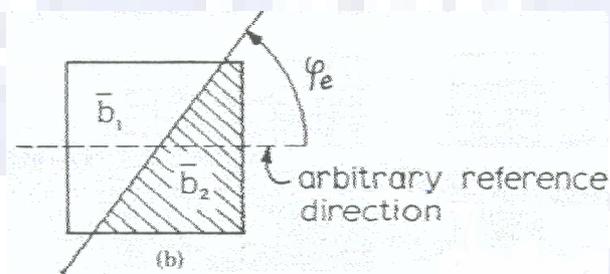


圖2.11 邊緣(Edge)示意圖

5. 當一個有寬度的直條(亮度為 $\bar{b}_1$ )通過一個區域時，而在此寬度的直條的兩邊(亮度為 $\bar{b}_2$ )，此時我們就可以定義出它的大小跟方向。

$$magnitude = |\bar{b}_1 - \bar{b}_2| \quad , \quad 0 \leq \phi_e \leq \pi \quad (2.15)$$

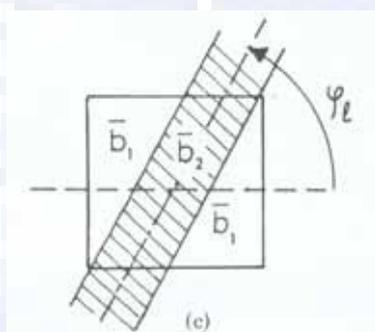


圖2.12 線條示意圖

6. 當我們做邊緣演算法的時候，大部分都分為兩部分：
- 計算出影像中每一個像素最大的平均梯度值。
  - 若  $B$  為影像中的區域影像(sub-area)， $W_1$ 、 $W_2$ 為各邊緣搜尋的 Template，我們可以利用下列公式計算出它的平均梯度值(Average Gradient：AG)跟梯度方向，當計算出的AG大於臨界值時就視為邊緣(Edge)；並且可由其角度判斷與其它邊緣的連結方向。

$$AG \approx [(B, W_1)^2 + (B, W_2)^2]^{\frac{1}{2}} \quad (2.16)$$

$$\phi = \tan^{-1} \left[ \frac{(B, W_1)}{(B, W_2)} \right] \quad (2.17)$$

## 第三章

### 理論：研究內容與方法

#### 3.1 變形樣本模型(deformable template model)

當我們運用 RCER 檢測出眼睛跟眉毛的粗略輪廓後發現，眉毛的輪廓位置已經相當準確了，但是眼睛卻還沒那麼準確，為了得到更精確的眼睛位置，於是我們就使用變形樣本模型來完成眼睛位置、形狀跟大小的判斷。變形樣本模型最先是由 Yuille 所提出來的。而這方法主要是以一組參數集合來描述一個樣本，用這個樣本來代表特徵的形狀。而這個樣本是很有彈性的，可以藉由調整參數值來改變它的尺寸、偏角和形狀。最後，我們找一組最符合的參數集合來代表特徵的形狀。

##### 3.1.1 眼睛樣本：

我們使用兩個拋物線來描述眼睛上、下邊界的輪廓。圖 3.1 為一眼睛樣本模型。這兩個拋物線的中心是  $(x_c, y_c)$ ，寬度是  $2b$ ，上拋物線的高度是  $c$ ，下拋物線的高度是  $a$ ，而且和水平線有一個  $\theta$  角度的偏角。這個樣本總共有六個參數，可以表示成  $\bar{g} = (x_c, y_c, b, a, c, \theta)$ 。

由於先前我們曾使用 RCER 獲得粗略的輪廓，因此我們可以假設出這六個變數的可能範圍。

$$(a) \quad a \in [a - \alpha_1, a + \alpha_1] \quad (3.1)$$

$$(b) \quad b \in [b - \beta_1, b + \beta_1] \quad (3.2)$$

$$(c) \quad c \in [c - \gamma_1, c + \gamma_1] \quad (3.3)$$

$$(d) \theta \in [\theta - \theta_1, \theta + \theta_1] \quad (3.4)$$

$$(e) x_c \in [x_c - \delta_1, x_c + \delta_1] \quad (3.5)$$

$$(f) y_c \in [y_c - \varepsilon_1, y_c + \varepsilon_1] \circ \quad (3.6)$$

$(P_x, P_y)$  表示眼睛的中點。舊座標  $(x', y')$  經過平移和旋轉後所得到的新座標可以表示如下：

$$x + P_x = x' \cos \theta + y' \sin \theta \quad (3.7)$$

$$y + P_y = -x' \sin \theta + y' \cos \theta \quad (3.8)$$

使用新的座標系統，用來描述眼睛上邊界的拋物線可以表示成：

$$y = a - \frac{a}{b^2} x^2 \quad (3.9)$$

在此  $x \in [-b, b]$ 。

相似地，描述眼睛下邊界的拋物線可以表示成：

$$y = -c + \frac{c}{b^2} x^2 \quad (3.10)$$

在此  $x \in [-b, b]$ 。

接著，我們定義由樣本參數所組成的能量函數。總能量函數定義如下：

$$E_{\text{total}} = E_{\text{edge}} + E_{w-b} \quad (3.11)$$

$E_{\text{edge}}$  和  $E_{w-b}$  函數的定義為：

$$E_{\text{edge}} = \frac{w_1}{\text{Upper-length}} \int_{\text{upper-bound}} \phi_{\text{edge}}(x, y) dS - \frac{w_2}{\text{Lower-length}} \int_{\text{lower-bound}} \phi_{\text{edge}}(x, y) dS \quad (3.12)$$

在此

Upper-bound 代表眼睛的上半部份

Lower-bound 代表眼睛的下半部份

$\phi_{\text{edge}}$  : the potential field for the edge

$$E_{w-b} = \frac{1}{Area} \iint_{Para-area} (-w_b N_{black}(x, y) + N_{white}(x, y)) dA \quad (3.13)$$

在此  $N_{black}(x, y)$  代表黑點的數目

黑點： $I(x, y)$  of points  $\leq$  threshold - tolerance,

， $N_{white}(x, y)$  代表白點的數目。

黑點和白點的定義如下：

白點： $I(x, y)$  of points  $\geq$  threshold + tolerance,

模糊點：otherwise.

$w_b$ ：黑點的權重； $w_{white}$ ：白點的權重

我們在這些可能的參數集合中找出使得總能量最小的那組集合，用那組參數來描述精準的眼睛輪廓。

圖 3.2 是原始的灰階影像的人臉，圖 3.3 是利用上述方式所檢測出來人臉的眼睛輪廓。

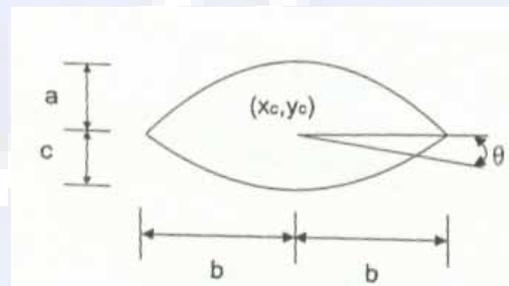


圖 3.1 為一眼睛樣本模型



圖 3.2 原始的灰階影像的人臉



圖 3.3 RCER 配合變形樣本所檢測出的眼睛輪廓

## 3.2 邊緣搜尋 (Edge detection)

### 3.2.1 Robert 的運算方法

首先介紹的是Robert的運算方法，在作各項遮罩[Template;mask]的介紹時，我們必須先對遮罩的運算方法作一個簡單的介紹。

邊緣是指周圍影像的像素的灰階值(gray level)有所謂的 ramp edge、step edge、line edge、roof edge的變化，如圖 3.4。影像的邊緣對應著影像灰階的不連續性。顯然影像的邊緣很少是從一個灰階跳到另一個灰階的這樣的理想狀況。真實影像的邊緣通常都是具有有限的寬度呈現出陡峭的斜坡狀。

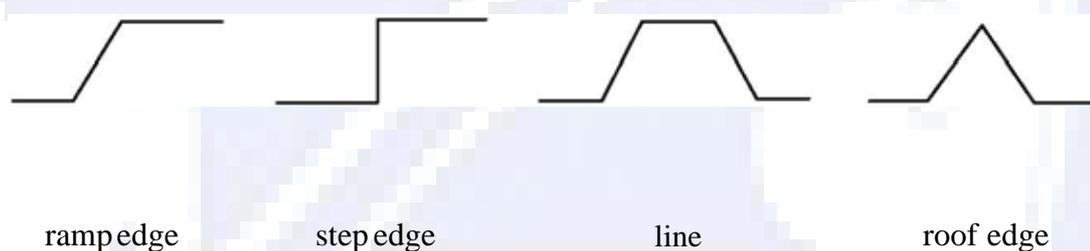


圖3.4 各種的邊緣介紹

邊緣的銳利程度由影像灰階的梯度來決定。梯度是一個向量，而  $\nabla f$  指出灰階變化最快的方向和大小。

$$\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad (3.14)$$

梯度的大小：

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (3.15)$$

梯度的方向：

$$\theta = \tan^{-1} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \quad (3.16)$$

因此最簡單的邊緣檢測運算方法是用影像的垂直和水平差分來近似梯度的運算：

$$\nabla f = (f(x, y) - f(x-1, y), f(x, y) - f(x, y-1)) \quad (3.17)$$

因此當我想搜尋邊緣的時候，最簡單的方法就是對每一個影像的像素計算出 (2.4) 的向量，然後求出他的絕對值，然後進行臨界值的設定就可以了。利用這種想法就可以得到 Robert 的演算式，它是兩個  $2 \times 2$  的遮罩運算的結果：

$$R(i, j) = \sqrt{(f(i, j) - f(i+1, j+1))^2 + (f(i, j+1) - f(i+1, j))^2} \quad (3.18)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

圖3.5 Robert 的遮罩

能查到的最早的有關邊緣檢測的文獻就是 1959年 B. Jules 的[37]，這算是最早提出邊緣檢測和邊緣檢測的Template的文章了。1963年Robert提出了邊緣檢測和邊緣檢測的遮罩這個非常簡單運算的演算法。“Machine Perception of 3-D Solids”是Robert在1963年寫的(MIT)博士畢業論文。它是最早分析影像中的邊緣、線、模型和圖形學的文章。Robert提出的系統是第一個 3D 視覺系統，其中有許多後來被大家常用的運算式。它使用了三維物體的多邊形模型，通過計算影像中的灰階值的數據尋找影像中的Segment“段”然後再將它予以標示，然後經過一些方法把物體給呈現出來。一個複雜的物體是由很多“段”所組成的，而“段”是由邊緣所組成的。Robert使用簡單的 2\*2 的遮罩得到了邊緣。而相鄰的邊緣點組成了線，如果線是比較短，不足以視為線，那麼這條線就被刪除，如果線比較長，那麼我們就把它延長直到它與其它的邊緣線相交於一點。然後通過對這些線的組成來辨識物體。一旦物體識別出來後，表示這個物體的邊緣就可以從影像的邊緣線上刪除；然後再對下一個物體進行辨識。Robert的遮罩是很直觀也很簡單的。也因為如此，所以它的檢測結果並不理想，故在本論文中並沒有使用此方法來辨識眼睛的位置。

### 3.2.2 Prewitt、Sobel 的邊緣搜尋演算法

#### Prewitt 和 Sobel 的運算方法

由於Robert發現了這個理論後，接著有更多的人投入這樣的研究與實驗，發現了其它類似運算的遮罩。

在1970 年左右Prewitt和Sobel分別提出了一個Template，這就是 Prewitt 的遮罩和Sobel的遮罩。Prewitt邊緣檢測的遮罩使用兩個有方向性的遮罩(一個是水平的，一個是垂直的)，每一個都可以算出一個偏導數：

$$Prewitt_{\text{垂直}} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad Prewitt_{\text{水平}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

圖3.6 Prewitt 的遮罩

如果我們用 Prewitt 的運算方法來檢測影像 M 的邊緣的話，我們可以先分別用水平的遮罩和垂直的遮罩對影像進行捲積(Convolution)，得到的是兩個矩陣，在不考慮邊界的情形下也是和原影像同樣大小的 M1、M2，他們分別表示影像 M 中相同位置處的兩個偏導數。然後把 M1和M2 對應的位置的兩個數取平方和開根號得到一個新的矩陣 G，G 表示 M 中每一個像素的灰階的梯度值。然後  $Edge \text{ Magnitude} > Threshold$  就視為邊緣，反之則不是。

運算式如下：

$$Edge \text{ Magnitude} = \sqrt{((M \otimes P_{\text{垂直}})^2 + (M \otimes P_{\text{水平}})^2)} > Threshold \quad (3.19)$$

$$Edge \text{ Magnitude} = \sqrt{(P_{\text{垂直}})^2 + (P_{\text{水平}})^2)} > Threshold \quad (3.20)$$

$$\theta = \tan^{-1} \left( \frac{P_{\text{垂直}}}{P_{\text{水平}}} \right) \quad (3.21)$$

而Sobel的運算方法和Prewitt的運算方法的差別只在於使用的遮罩不一樣：

$$Sobel_{\text{垂直}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad Sobel_{\text{水平}} = \begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & -2 \\ -1 & 0 & -1 \end{bmatrix}$$

圖3.7 Sobel 的遮罩

運算式如下：

$$Edge \text{ Magnitude} = \sqrt{\left((M \otimes Sobel_{\text{垂直}})^2 + (M \otimes Sobel_{\text{水平}})^2\right)} > Threshold \quad (3.22)$$

$$Edge \text{ Magnitude} = \sqrt{\left((Sobel_{\text{垂直}})^2 + (Sobel_{\text{水平}})^2\right)} > Threshold \quad (3.23)$$

$$\theta = \tan^{-1} \left( \frac{Sobel_{\text{垂直}}}{Sobel_{\text{水平}}} \right) \quad (3.24)$$

Prewitt 跟 Sobel 兩種尋邊的方式，基本上是大同小異的。完全取決於所要尋邊影像的品質決定。但是它們卻存在一些共同的問題：

- A. 他們的結果對雜訊很敏感，影像的離散差分對雜訊比對原影像更敏感。
- B. 可以通過先對影像做平滑以改善結果，但是又會產生一個問題：會把一些靠在一起的邊緣平滑掉，而且會影響對邊緣的定位。
- C. 用這些遮罩捲積以後得到的邊緣可能是跨越了好幾個點而不是一個點。

所以我們就又找了Frei-Chen跟Canny演算法來作實驗，原理說明在下面章節在予以介紹。

### 3.2.3 Frei-Chen 的邊緣搜尋演算法

#### Frei-Chen 演算法

1977 年，Frei-Chen [35] 又提出了一種新的邊緣運算法，這種算法有 9 個遮罩。而這種運算方式是由九個 Template 做運算，每一個像素都需要運算九次。前四個遮罩組成“邊”的空間( $f_1$ 、 $f_2$  稱為 Isotropic smoothed Gradient,  $f_3$ 、 $f_4$  稱為 ripple); 可由  $f_1$ 、 $f_2$ 、 $f_3$ 、 $f_4$  等四個遮罩找出影像的邊緣點，後四個遮罩組成“線”的空間( $f_5$ 、 $f_6$  稱為 line;  $f_7$ 、 $f_8$  稱為 discrete laplacian); 可由  $f_5$ 、 $f_6$ 、 $f_7$ 、 $f_8$  等四個遮罩找出影像的線條，最後一個遮罩用於平均化的空間。其九個遮罩如圖 3.8 所示。

$$\begin{matrix} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} & \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} \\ f_1 & f_2 & f_3 & f_4 \end{matrix}$$

(a) 邊緣的Template

$$\begin{matrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \\ f_5 & f_6 & f_7 & f_8 \end{matrix}$$

(b) 直線的Template

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f_9$

(c) 平均化的空間的Template

圖3.8 Frei-Chen 的 Template

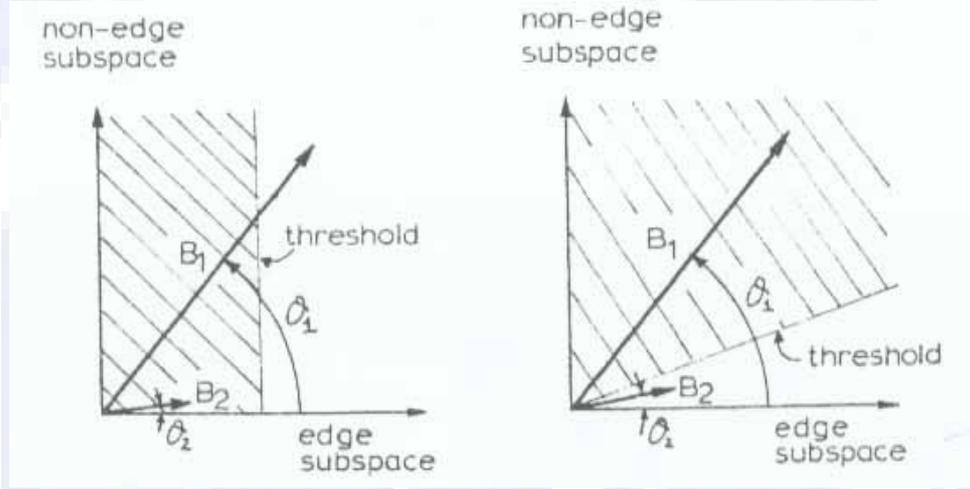


圖3.9 一般邊緣臨界值與 Frei-Chen 的臨界值的比較

我們可由下面的公式跟圖3.9中了解到一個邊緣的選擇與臨界值的決定。首先，我們先認定有一個子區域(sub\_area)的邊緣在這影像中，而且並不知道它的定位(orientation)；再者發現有一組擴展的子區域正交邊緣的向量 $[T_1, \dots, T_e]$ ，當然就有一組  $n^2-e$  的 “non\_edge” 非邊緣的向量。

於是我們依據前面敘述所定義的，將公式列舉如下：

$$\phi = \cos^{-1} \left[ \frac{\sum_{i=1}^e (B, T_i)^2}{\sum_{j=1}^{n^2} (B, T_j)^2} \right]^{1/2} \quad (0 \leq \theta \leq \pi) \quad (3.25)$$

$$threshold = \sum_{i=1}^e \frac{(B, T_i)^2}{(B, B)} \quad (3.26)$$

在表3.1 中我們明確的計算出當Sobel、Prewitt、Frei-Chen三種演算法再做邊緣檢測時所需的計算次數的比較，當我們在決定使用哪一種邊緣檢測的遮罩時，都必須在電腦計算時間與檢測的效果中作一個取捨。也因為如此，所以本論文才不採取更複雜計算的Kirsch和Robinson而較準確的輪盤式(compass)的邊緣檢測的遮罩。

表3.1 Sobel、Prewitt、Frei-Chen三種演算法的比較[35]

	Sobel	Prewitt	Frei & Chen
Multiply & Divide	4	4	13
Add & Subtract	11	11	27
Compare	1	1	1

而在使用 Frei-Chen 的演算法時發現他比 Sobel 跟Prewitt 的遮罩在做邊緣檢測的時候有以下幾個優點：

- (1) 有更多的邊緣的子區域被偵測出來。
- (2) 較粗的邊緣會被當成細瘦的邊緣線條被檢測出來。
- (3) 在黑暗中的邊緣此法比其他的方法更容易可以檢測出來。
- (4) 但是此法對於影像的雜訊太過於靈敏。

### 3.2.4 Canny 的邊緣搜尋演算法

#### Canny邊緣檢測—最優的階梯型邊緣檢測演算法

除了Sobel、Prewitt、Frei-Chen三種方法外，在1986年 Canny 在 IEEE 上發表了劃時代意義的文章 “A Computational Approach to Edge Detection”。這篇文章承前啟後，作者對過去的一些方法和應用做了一個整理[38], [39]，並在此基礎上提出了邊緣檢測的三條準則：這就是著名的 Canny 準則(Canny's Criteria)，並在此基礎上得到了一個很不錯的實用演算法。本章節主要就是要介紹 Canny 準則及其演算法。

Canny 準則的目的就在於：在對信號和濾波器做出一定假設的條件下利用數值計算方法求出最優濾波器並對各種濾波器的性能進行比較。

## 1. 邊緣檢測的 Canny 準則

Canny 考察了以往的邊緣檢測運算元和邊緣檢測的應用，他發現儘管這些應用出現在不同的領域，但是他們都有一些共同的要求：

- (1) 良好的邊緣檢測能力：找出具灰階變化的邊緣，要達此目的必須先去除雜訊，使訊號雜訊比(Signal-to-Noise)增大。換句話說就是對邊緣的錯誤檢測率要盡可能的低：就是在影像上邊緣出現的地方檢測結果中不應該沒有；另一方面也不要出現假的邊緣。這是顯然的，所有使用邊緣檢測做更深入工作的系統，它的性能都依賴於邊緣檢測的誤差。
- (2) 好的定位能力：希望真正灰階變化的位置和偵測找出的邊緣線間距離越進越好，距離小代表定位準確。也就是我們標記出的邊緣位置要和影像上真正邊緣的中心位置充分接近。
- (3) 多重感應：一般邊緣檢測中都針對前兩項設計，但Canny認為應考慮單一邊緣產生多個感應的問題，這會造成同一個邊緣被視為多個邊緣的情況，因此Canny在此多加一個約制條件，即從單一邊緣線產生的多重感應中找出正確的邊緣線。也就是對同一邊緣要有低的回應次數。

這就是Canny三準則。事實上在Canny之前就有人提出了類似的要求[40], [41], 但Canny是第一個明確提出這三條並完整解決這個問題的。更重要的是Canny給出了這三條準則的數學運算式(以一維為例)。這就使得我們在檢測邊緣的時候有了更明確的方法。

以下是關於三個準則的演算法：

**(1) 良好的邊緣檢測能力 (Good detection) :**

$G(x)$  : 檢測邊緣的曲線

$n(x)$  : 雜訊

邊緣以外的影像訊號，如下：

$$H_G = \int_{-w}^w G(-x)f(x)dx \quad (3.27)$$

雜訊的平方根，如下：

$$H_n = n_0 \left[ \int_{-w}^w f^2(x)dx \right]^{1/2} \quad (3.28)$$

$$SNR(f) = \frac{H_G}{H_n} = \frac{\left| \int_{-w}^w G(-x)f(x)dx \right|}{n_0 \left[ \int_{-w}^w f^2(x)dx \right]^{1/2}} \quad (3.29)$$

**(2) 好的定位能力 (Good localization) :**

a.  $H_G(x) + H_n(x)$  在  $x_0$  處得到最大值，所以

$$H_n'(x_0) + H_G'(x_0) = 0 \quad (3.30)$$

b.  $H_G(x)$  在  $x=0$  時有最大值，所以

$$H'_G(x_0) = 0 \quad (3.31)$$

c. 於是就有  $H'_G(x_0) = H'_G(0) + H''_G(0)x_0 + 0(x_0^2) \approx H''_G(0)x_0$

$$\text{即： } H''_G(0)x_0 = -H'_n(x_0) \quad (3.32)$$

$$\text{從而 } E(x_0^2) = \frac{E[(H''_n(x_0))^2]}{H''_G(0)^2} = \frac{n_0^2 \int_{-W}^W f'^2(x) dx}{[\int_{-W}^W G'(-x) f'(x) dx]^2} \quad (3.33)$$

$E(x)$ :  $x$  的期望值， $x_0$  越小 localization 越準確

$$Loc(f) = \frac{\left| \int_{-W}^W G'(-x) f'(x) dx \right|}{n_0 \int_{-W}^W f'^2(x) dx} \quad (3.34)$$

而目標為求一函數  $f(x)$ ，使得下面式子達到最大值

$$J(f) = \frac{\left| \int_{-W}^W G(-x) f(x) dx \right| \left| \int_{-W}^W G'(-x) f'(x) dx \right|}{n_0 \left[ \int_{-W}^W f^2(x) dx \right]^{1/2} n_0 \int_{-W}^W f'^2(x) dx} \quad (3.35)$$

### (3) 單一的響應 (Single response) :

在理想狀況下，我們用濾波器對雜訊響應的兩個峰值間的距離來近似濾波器對一個邊緣點響應的長度。因為輸出信號中相鄰兩個極大值點的距離是相鄰兩個零交叉點距離的兩倍，公式如下：

$$x_{zc}(f) = \pi \left( \frac{-R(0)}{R''(0)} \right)^{1/2} \quad (3.36)$$

$$R(0) = \int_{-\infty}^{+\infty} g^2(x) dx, \quad R''(0) = \int_{-\infty}^{+\infty} g''^2(x) dx \quad (3.37)$$

所以雜訊在  $f(x)$  濾波後兩個相鄰極大值點的距離為

$$x_{\max}(f) = 2x_{zc}(f) = kW \quad (3.38)$$

$W$  為濾波器的半寬度，故在  $2W$  長的區域裡出現最大值的個數的期望值為

$$N_n = \frac{2W}{x_{\max}} = \frac{2W}{kW} = \frac{2}{k} \quad (3.39)$$

我們下面以步階邊緣 (STEP EDGES) 為例，解釋 Canny 的第一準則與第二準則：

我們將第一跟第二準則的表示式寫為

$$SNR(f) = \frac{A \left| \int_{-w}^0 f(x) dx \right|}{n_0 \left[ \int_{-w}^w f^2(x) dx \right]^{1/2}} \quad (3.40)$$

$$Loc(f) = \frac{|f'(0)|}{n_0 \int_{-w}^w f^2(x) dx} \quad (3.41)$$

再令  $f_w(x) = f(x/w)$ ，可得

$$SNR(f_w) = \sqrt{w}SNR(f) \quad Loc(f_w) = Loc\sqrt{w}(f) \quad (3.42)$$

亦即當增大濾波器寬度時， $SNR(f)$  會提高，而  $Loc(f)$  會下降

$$\text{則 } J(f_w) = J(f) = \frac{A \left| \int_{-w}^0 f(x) dx \right|}{n_0 \left[ \int_{-w}^w f^2(x) dx \right]^{1/2}} \frac{A |f'(0)|}{n_0 \int_{-w}^w f'^2(x) dx} \quad (3.43)$$

$$\text{再簡化為 } J(f) = \frac{\left| \int_{-w}^0 f(x) dx \right|}{\left[ \int_{-w}^w f^2(x) dx \right]^{1/2}} \frac{|f'(0)|}{\int_{-w}^w f'^2(x) dx} \quad (3.44)$$

利用差分求得最優化的濾波器

## 2. Canny 邊緣檢測的技術

Canny 在做邊緣檢測的時候更使用了兩種技術，來增加他的演算法的準確率。一是雙臨界值技術；二是多尺度技術，我們將在下面作一個介紹。

(1) 雙臨界值技術：

在  $T_1$  中收取邊緣，將  $T_2$  中所有間隙聯接起來。選取雙臨界值  $\tau_1$  和  $\tau_2$ ，且  $\tau_1$  和約等於  $2\tau_2$ 。得到兩個臨界值的邊緣影像  $T_1[I, j]$  和  $T_2[I, j]$ ，在 matlab 中是取  $\tau_2$  等於  $0.4\tau_1$ 。判斷其邊緣的原則如下：

(a)、大於高臨界值  $\tau_1$  則為 edge。

(b)、於低臨界值  $\tau_2$  則為不是 edge。

(c)、介於高、低臨界值(1) 和(2)之間則需觀看他的八個鄰接像素有沒有 edge；

若有則為 edge，若無則不是 edge。

在圖3.10 便是雙臨界值技術的圖解。

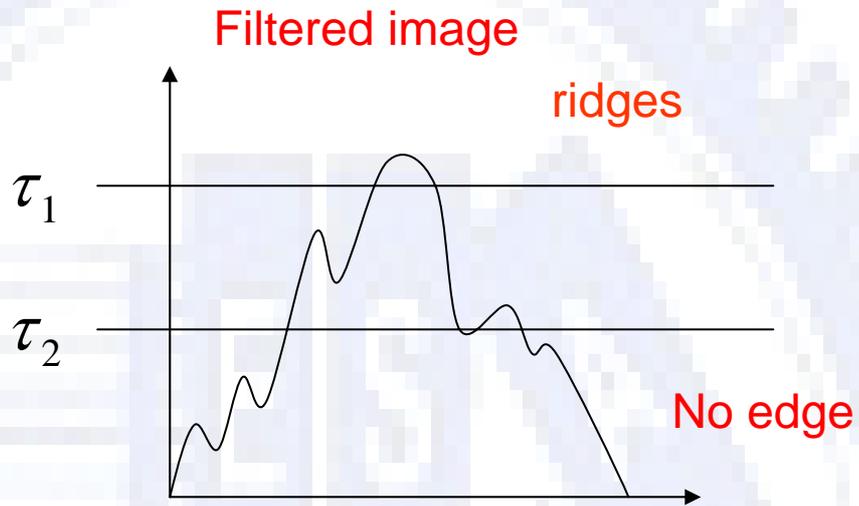


圖3.10 雙臨界值技術的圖解

(2) 多尺度技術：

所謂的多尺度技術就是指遮罩的維度  $W$ ，也就是  $W$  越大則檢測效果越好，雜訊影響越小，但 LOCALATION 則越不準確。

雖然有此缺點，但我們有時仍需使用此技術，原因為：

- (a)、我們不可能因為節省計算時間而選擇無窮小的遮罩，故大部分都以一定的維度做檢測的工作；通常是  $3 \times 3$  的遮罩。
- (b)、遮罩的維度，會影響邊緣的檢測。
- (c)、不同維度的遮罩會包含不同的邊緣訊息。故想要很好的求出 edge 則必須在多種維度下的遮罩作檢測。

(3) 綜觀以上Canny的邊緣檢測，我們可以將其演算法歸納為幾個步驟。

步驟一、用高斯濾波器平滑圖像：

$$H(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.45)$$

$$G(x, y) = f(x, y) * H(x, y) \quad (3.46)$$

步驟二、用一階偏導的有限差分來計算梯度的幅值和方向：

利用 Sobel 等類似的遮罩，算出梯度大小與方向。

步驟三、對梯度幅值進行非極大值抑制：

偏偏得到全部的梯度並不足以確定邊緣，因此為確定邊緣，必需保留局部梯度最大的點，而抑制非極大值 (non-maxima suppression, NMS)。圖3.11 乃是抑制非極大值的說明圖。

而解決方法就是利用梯度的方向，將梯度 $0^\circ \sim 180^\circ$ 分為四種方向[水平( $0^\circ \sim 22.5^\circ$ 、 $157.5^\circ \sim 180^\circ$ )、垂直( $67.5^\circ \sim 112.5^\circ$ )、45度( $22.5^\circ \sim 67.5^\circ$ )、 $135^\circ$  ( $112.5^\circ \sim 157.5^\circ$ )]，如圖3.11所示：這些方向是梯度的方向，通過此方向我們可以找到下一個鄰接的 pixel。

四個扇區的編碼為0到3，對應 $3 \times 3$ 領域的四種可能組合。在每一點上，領域的中心像素 M 與沿着梯度線的兩個像素相比。如果 M 的梯度值不比沿梯度線的兩個相鄰像素的梯度值大，則令  $M=0$ 。

即：

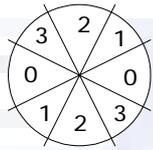
$$\xi[i, j] = \text{Sector}(\theta[i, j]) \quad (3.47)$$

$$T[i, j] = \text{NMS}(M[i, j], \xi[i, j]) \quad (3.48)$$

而其判斷過程如下：

1. 從每個位置(x, y)中，到與邊緣定位方向 $\theta(x, y)$ 垂直的兩個方向。

2. 標示最初像素C (x, y)，兩相鄰像素在垂直處標示以A和B。
  3. 如果 $M(A) > M(C)$ 或者 $M(B) > M(C)$ ，確定 $M(C) = 0$  去掉像素C。
- $M(C) > M(A)$ 且 $M(C) > M(B)$ 像素A and B被去除，像素C則保留。



1	2	3
8		4
7	6	5

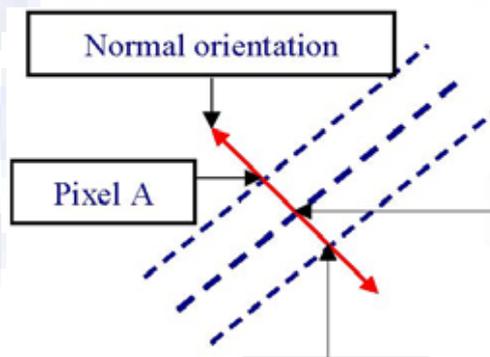
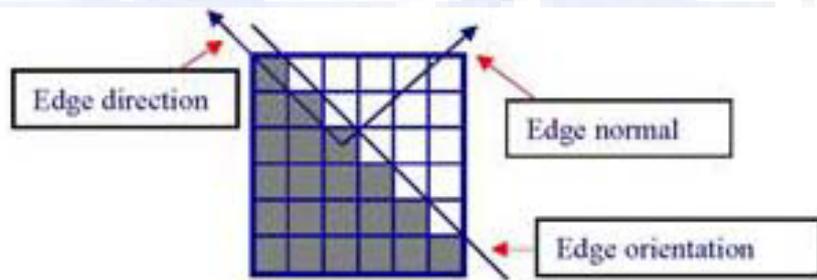


圖3.11 抑制非極大值 (non-maxima suppression, NMS) 圖解

步驟四、用雙閾值演算法檢測和連接邊緣：

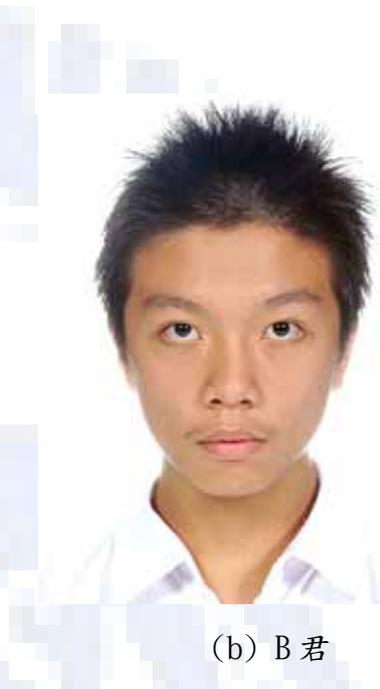
減少假邊緣段數量的典型方法是對  $T[i, j]$  使用一個閾值。將低於閾值的所有值賦零值。但問題是如何選取閾值？

解決方法：雙閾值演算法。雙閾值演算法對非極大值抑制圖像作用兩個閾值  $\tau_1$  和  $\tau_2$ ，且  $2\tau_1 \cong \tau_2$ ，從而可以得到兩個閾值邊緣圖像  $T_1[i, j]$  和  $T_2[i, j]$ 。由於  $T_2[i, j]$  使用高閾值得到，因而含有很少的假邊緣，但有間斷(不閉合)。雙閾值法要在  $T_2[i, j]$  中把邊緣連接成輪廓，當到達輪廓的端點時，該演算法就在  $T_1[i, j]$  的 8 鄰點位置尋找可以連接到輪廓上的邊緣，這樣，演算法不斷地在  $T_1[i, j]$  中收集邊緣，直到將  $T_2[i, j]$  連接起來為止。

## 第四章 實驗結果

### 4.1 臉部影像資料庫 (Database of the face images) 與實驗步驟

在本節中，我們首先建立臉部影像的資料庫，然後用模擬來驗證我們所提出的方法。我們需要的是正面的臉部影像，而照相的背景、照明和距離最好能固定。影像的解析度是  $128 \times 128$ 。在臉部影像資料庫裡包含了五個不同的男性與五個不同的女性，每個人各有一張全彩的照片如圖 4.1 跟一張經過 YCbCr 的方法取出 Y 值的灰階影像圖，如圖 4.2 所示。





(g) G 君



(h) H 君



(i) I 君

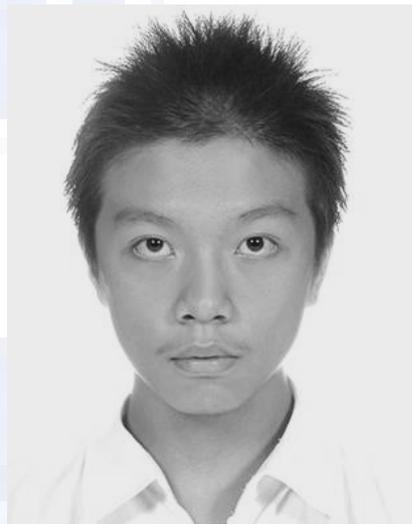


(j) J 君

圖 4.1 十張原始的人臉影像



(a) A 君



(b) B 君



(c) C 君



(d) D 君



(e) E 君



(f) F 君



(g) G 君



(h) H 君



(i) I 君



(j) J 君

圖 4.2 十張取出 Y 訊號的原始的人臉影像

我們將依下列的步驟，一步一步的利用五種檢測眼睛的演算法來將眼睛的輪廓找出來，並計算他們的像素值與準確度。

1. 首先我們先將十張全彩的影像取出 Y 成份出來，如圖 4.5 所示。
2. 使用 RCER 獲得眉毛和眼睛的粗略輪廓。由於所得到的眉毛輪廓已經十分精準，足夠讓我們做眼睛的搜尋動作，所以不需要再強化其輪廓。圖 4.3 顯示用 RCER 所得到眼睛的粗略輪廓。因為所得到眼睛的輪廓不夠精確，所以我們使用變形樣本模型來加強其輪廓。
3. 將在上面所得到的粗略輪廓，使用變形樣本把眼睛描繪的更清楚。如圖 4.4 顯示用變形樣本模型所得到的眼睛的輪廓。
4. 再將相同的十張原始影像，以 YCbCr 的方法取出 Y 訊號的影像，然後以邊緣檢測的方法(Sobel、Prewitt、Frei Chen、Canny)將眼睛的輪廓檢測出來。
5. 接著將原始的影像資料庫中的十張影像用電腦將眼睛跟眼珠的像素值計算出來。
6. 然後將變形樣本跟邊緣檢測所檢測出來的眼睛跟眼珠的輪廓平移至原始影像圖上，可做一個比對。
7. 依照同樣的方法把變形樣本跟邊緣檢測所檢測出來的眼睛跟眼珠的輪廓的大小，算出他們的像素值。
8. 將測量的眼睛及眼珠分別對應到我們原始資料庫的影像作比對跟計算，計算出的統計值作成一個比對表，如表 4.1~4.10。
9. 最後將表 4.1~4.10 五種演算法的眼睛跟眼珠的平均準確率算出後統計於表 4.11。
10. 分別將五種檢測眼睛演算法檢測眼睛所花費的時間以電腦計算出來，再加以平均做成表 4.12。

使用的電腦配備為：

CPU：Intell Pentium III 569 MHz

RAM 記憶體：384 M

表 4.1~4.12 數據皆以上述的電腦配備所做出來的結果。

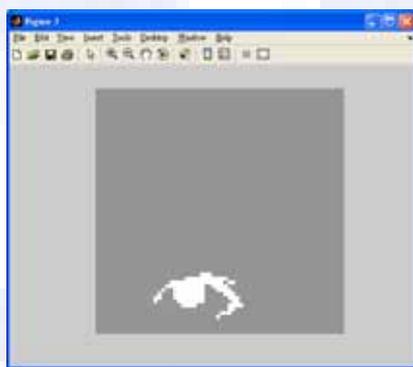
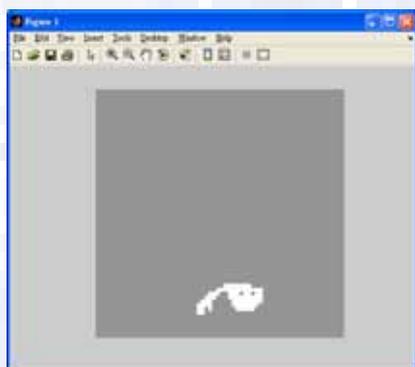


圖 4.3(a) RCER 所得到 B 君眼睛的粗略輪廓

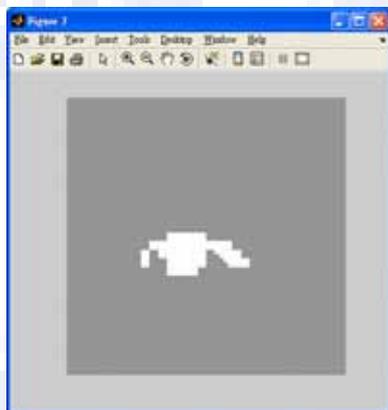
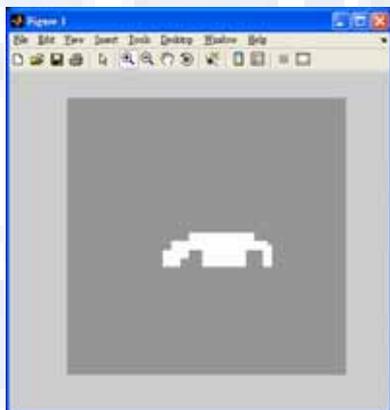


圖 4.3(b) RCER 所得到 D 君眼睛的粗略輪廓

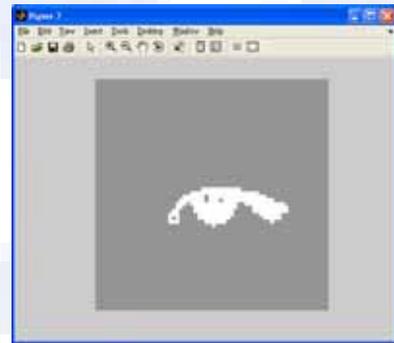
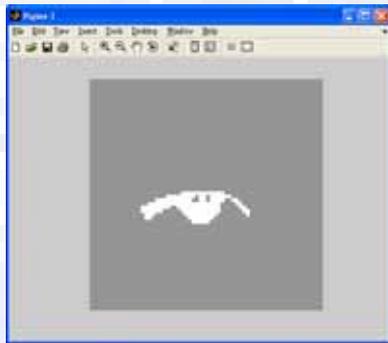


圖 4.3(c) RCER 所得到 E 君眼睛的粗略輪廓

圖 4.3 所得到眼睛的粗略輪廓



圖 4.4(a) 變形樣本模型所得到 B 君眼睛的輪廓

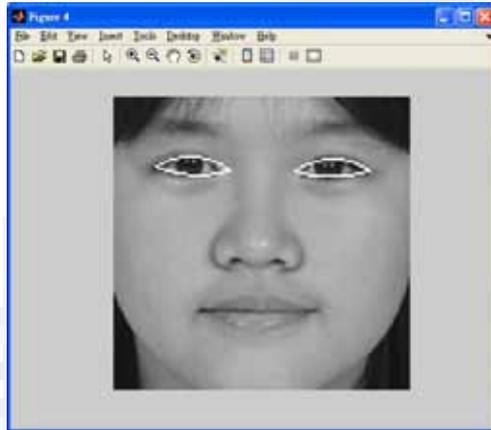


圖 4.4(b) 變形樣本模型所得到 D 君眼睛的輪廓

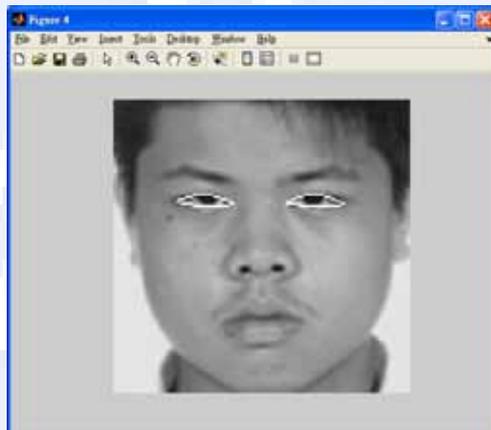


圖 4.4(c) 變形樣本模型所得到 E 君眼睛的輪廓

圖 4.4 變形樣本模型所得到眼睛的輪廓



圖 4.5(a) A 君的全彩影像與 Y 成分影像

表 4.1 A 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y (原圖)			121	116	70	65				
Prewitt			101	109	61	57	0.835	0.940	0.871	0.877
Canny			110	100	46	52	0.909	0.862	0.657	0.800
Sobel			94	96	64	57	0.777	0.828	0.914	0.877
Frei Chen			89	96	64	60	0.736	0.828	0.914	0.923
變形 樣本			113	109	65	57	0.934	0.940	0.929	0.877

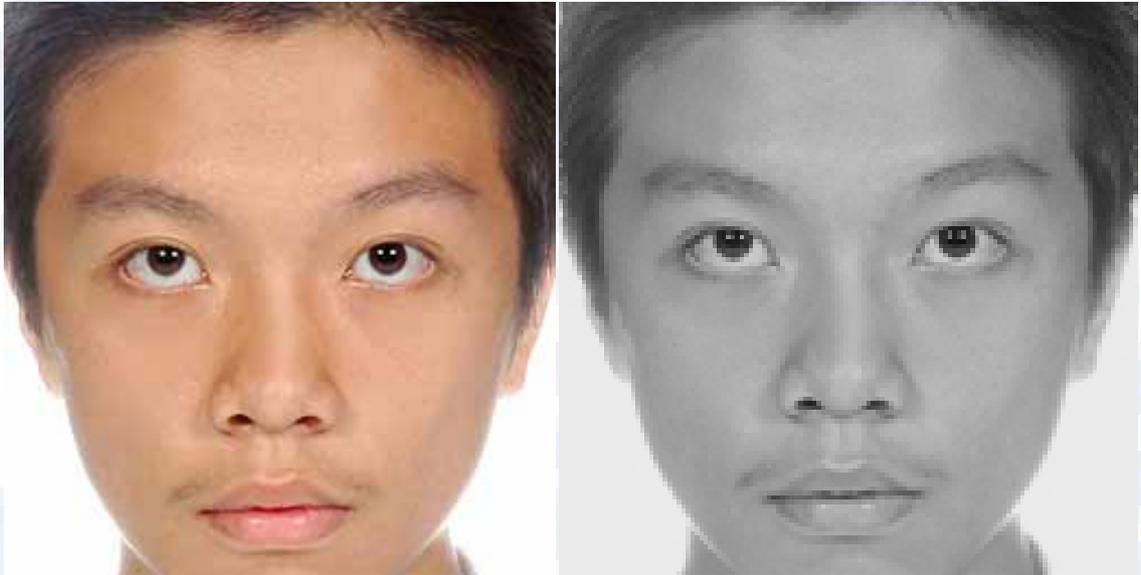


圖 4.5(b) B 君的全彩影像與 Y 成分影像

表 4.2 B 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y (原圖)			175	165	60	62				
Prewitt			165	161	58	62	0.943	0.956	0.967	1.0
Canny			157	146	50	59	0.897	0.885	0.833	0.952
Sobel			165	141	58	62	0.943	0.855	0.967	1.0
Frei Chen			138	120	60	59	0.789	0.727	1.0	0.952
變形 樣本			152	148	58	51	0.869	0.897	0.967	0.823



圖 4.5(c) C 君的全彩影像與 Y 成分影像

表 4.3 C 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			196	205	103	107				
Prewitt			181	204	102	83	0.924	0.995	0.990	0.776
Canny			195	204	93	94	0.995	0.995	0.903	0.879
Sobel			177	183	83	83	0.903	0.893	0.806	0.776
Frei-Chen			155	180	97	104	0.791	0.878	0.942	0.972
變形樣本			174	183	76	85	0.888	0.893	0.738	0.794



圖 4.5(d) D 君的全彩影像與 Y 成分影像

表 4.4 D 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y (原圖)			197	195	97	99				
Prewitt			185	172	92	83	0.939	0.882	0.949	0.838
Canny			186	164	82	75	0.944	0.841	0.845	0.758
Sobel			175	161	78	77	0.888	0.826	0.804	0.778
Frei-Chen			178	171	96	87	0.904	0.877	0.990	0.879
變形樣本			153	159	69	79	0.777	0.815	0.711	0.800



圖 4.5(e) E 君的全彩影像與 Y 成分影像

表 4.5 E 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			125	111	56	57				
Prewitt			108	103	35	45	0.864	0.928	0.625	0.790
Canny			119	108	44	44	0.952	0.973	0.786	0.772
Sobel			116	110	45	45	0.928	0.991	0.804	0.790
Frei-Chen			114	111	47	54	0.912	1.0	0.839	0.947
變形樣本			101	103	37	36	0.808	0.928	0.661	0.632



圖 4.5(f) F 君的全彩影像與 Y 成分影像

表 4.6 F 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			186	168	71	75				
Prewitt			142	138	67	72	0.763	0.821	0.944	0.960
Canny			165	167	68	68	0.887	0.994	0.958	0.907
Sobel			173	147	71	74	0.930	0.875	1.0	0.987
Frei-Chen			162	168	70	75	0.871	1.0	0.986	1.0
變形樣本			161	165	51	52	0.866	0.982	0.718	0.693



圖 4.5(g) G 君的全彩影像與 Y 成分影像

表 4.7 G 的五種方眼睛檢測法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			121	144	51	53				
Prewitt			113	116	49	42	0.934	0.806	0.961	0.792
Canny			108	129	40	48	0.893	0.896	0.784	0.906
Sobel			102	126	44	41	0.843	0.875	0.863	0.774
Frei-Chen			109	124	51	53	0.901	0.861	1.0	1.0
變形樣本			99	121	34	53	0.819	0.840	0.667	1.0



圖 4.5(h) H 君的全彩影像與 Y 成分影像

表 4.8 H 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			213	208	76	71				
Prewitt			147	170	57	60	0.690	0.817	0.750	0.845
Canny			207	205	52	59	0.972	0.986	0.684	0.831
Sobel			211	186	59	60	0.991	0.894	0.776	0.845
Frei Chen			175	200	75	71	0.822	0.962	0.987	1.0
變形樣本			194	192	59	67	0.911	0.923	0.776	0.944



圖 4.5(i) I 君的全彩影像與 Y 成分影像

表 4.9 I 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			152	192	54	64				
Prewitt			140	168	41	47	0.921	0.875	0.759	0.734
Canny			152	178	45	47	1.0	0.927	0.833	0.734
Sobel			140	146	38	47	0.921	0.760	0.704	0.734
Frei Chen			150	161	53	64	0.987	0.839	0.982	1.0
變形樣本			134	153	51	64	0.882	0.797	0.944	1.0



圖 4.5(j) J 君的全彩影像與 Y 成分影像

表 4.10 J 君的五種眼睛檢測方法的比較表

尋邊方法比較	右眼睛辨識位置	左眼睛辨識位置	右眼睛輪廓像素個數	左眼睛輪廓像素個數	右眼珠輪廓像素個數	左眼珠輪廓像素個數	右眼睛輪廓準確率	左眼睛輪廓準確率	右眼珠輪廓準確率	左眼珠輪廓準確率
Y(原圖)			133	137	57	53				
Prewitt			111	115	32	37	0.835	0.839	0.561	0.698
Canny			122	124	29	31	0.917	0.905	0.509	0.585
Sobel			114	124	36	39	0.857	0.905	0.632	0.736
Frei Chen			134	133	56	53	0.933	0.971	0.983	1.0
變形樣本			118	117	38	30	0.887	0.854	0.667	0.566

## 4.2 辨識結果(Recognition Results)

表 4.11 五種檢測眼睛方法的準確率

尋邊方法	右眼睛輪廓 準確率	左眼睛輪廓 準確率	眼睛輪廓 準確率	右眼珠輪廓 準確率	左眼珠輪廓 準確率	眼珠輪廓 準確率	眼睛跟眼 珠的準確 率%
Prewitt	0.865	0.888	87.7%	0.838	0.831	83.5%	85.6%
Canny	0.937	0.926	93.2%	0.779	0.812	79.6%	86.4%
Sobel	0.898	0.870	88.4%	0.827	0.830	82.7%	85.6%
Frei Chen	0.871	0.894	88.3%	0.962	0.967	96.5%	92.4%
變形樣本	0.864	0.887	87.6%	0.778	0.813	79.6%	83.6%

經過上述實驗的步驟後，我們把整個結果呈現於表 4.11。

表 4.12 五種檢測眼睛方法的計算時間

影像 edge detect Cpu-time	A 君	B 君	C 君	D 君	E 君	F 君
Prewitt	0.0701	0.0501	0.0501	0.0601	0.0501	0.0501
Canny	0.1803	0.2203	0.1702	0.1502	0.2103	0.1602
Sobel	0.03	0.0401	0.0401	0.0401	0.0501	0.0501
Freichen	0.2303	0.2303	0.2203	0.2103	0.2504	0.2704
變形樣本	88.0053	90.9908	86.3542	91.3514	80.1452	85.3348
影像 edge detect Cpu-time	G 君	H 君	I 君	J 君	average	
Prewitt	0.0501	0.0501	0.0401	0.0501	0.521	
Canny	0.1903	0.1602	0.1602	0.1702	0.17724	
Sobel	0.0501	0.0401	0.0601	0.0401	0.04409	
Freichen	0.2003	0.2303	0.2504	0.2504	0.23434	
變形樣本	81.2373	89.208	79.2247	84.2935	85.61452	

## 第五章

### 結論(Conclusion)

最後我們可以由實驗的結果知道眼睛跟眼珠的準確率與眼睛與眼珠的平均準確率如下：

#### 1. 眼睛的準確率

Canny > Sobel > Frei-Chen > Prewitt > 變形樣本

#### 2. 眼珠的準確率

Frei-Chen > Prewitt > Sobel > 變形樣本 = Canny

#### 3. 眼睛及眼珠的平均準確率

Frei-Chen > Canny > Sobel = Prewitt > 變形樣本

我們可從表 4.11 的實驗數據評估出以下幾件事：

1. 整體而言 Frei-Chen 的邊緣檢測最優。
2. 對於面積較大的邊緣來說，Canny 演算法較好。
3. 對於較黑、較小的邊緣，Frei-Chen 為最好。
4. Sobel 及 Prewitt 則差不多。

最後我們利用程式計算出五種檢測眼睛的演算德計算時間，並紀錄於表 4.12：我們可從表 4.12 的實驗數據評估出以下幾件事：

眼睛檢測的速度如下：

Sobel > Canny > Frei-Chen > Prewitt > 變形樣本

所以我們可以知道整體性的評估為 Frei-Chen 準確率最好，運算速度也是五種方法中僅次於 Sobel 的演算法。

目前我們只使用粗略輪廓預測副程式 RCER 配合變形樣本模型與四種邊緣檢測(Sobel、Prewitt、Frei-Chen、Canny)的方法來檢測眼睛的準確率。未來我們可以朝幾個方向來做。

- 一、目前是以灰階影像來作判斷，未來的工作可以朝彩色影像做研究。

二、目前以 128\*128 的影像圖來做檢測，有些細微的部份無法偵測出來故可以將像素質提高來得到更準確的偵測。

三、由於電腦的速度越來越快，所以我們也可以嘗試更精密更複雜的的演算法。

如果人臉部的眼睛及眼珠都可以被電腦清楚的辨識了，那我們只要能夠建立人臉部完整的影像資料庫，相信在這個領域將有更大的應用空間。

## 参 考 文 献

- [1] C. Darwin, *The Expression of the Emotions in Man and Animals*, London: John Murray, 1872.
- [2] P. Ekman and W. V. Friesen, *The Facial Action Coding System*, San Francisco, CA: Consulting Psychologist Press, 1978.
- [3] P. Ekman and W. V. Friesen, *Unmasking the Face*, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.
- [4] S. Morishima and H. Harashima, "Emotion space for analysis and synthesis of facial expression," in *Proc. 2nd IEEE Int. Workshop on Robot and Human Communication*, pp. 188-193, 1993.
- [5] K. Mase, "Recognition of facial expression from optical flow," *IEICE Trans.*, pp. 3474-3483, 1991.
- [6] H. Kobayashi and F. Hara, "The recognition of basic facial expressions by neural network," in *Proc. Int. Joint Conf. on Neural Network*, pp. 460-466, 1991.
- [7] H. Kobayashi and F. Hara, "Recognition of six basic facial expressions and their strength by neural network," in *Proc. IEEE Int. Workshop on Robot and Human Communication*, pp. 381-386, 1992.
- [8] H. Kobayashi and F. Hara, "Recognition of mixed facial expressions by neural network," in *Proc. IEEE Int. Workshop on Robot and Human Communication*, pp. 387-391, 1992.
- [9] H. Kobayashi and F. Hara, "Analysis of the neural network recognition characteristics of six basic facial expressions," in *IEEE Int. Workshop on Robot and Human Communication*, pp. 222-227, 1994.
- [10] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human expression recognition from motion using a radial basis function network architecture," *IEEE Trans. on Neural Networks*, vol. 7, no. 5, pp. 1121-1138, 1996.
- [11] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," In *Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog*, pp. 104-109, 1989.
- [12] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Compt. Vision*, pp. 99-111, 1992.
- [13] M. A. Shackleton and W. J. Welsh, "Classification of facial features for recognition," in *Proc. CVPR*, pp. 573-579, 1991.
- [14] X. Xie, R. Sudhakar and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, vol. 27,

- no. 6, pp. 791-799, 1994.
- [15] J.Y. Chang, J.L. Chen "A facial expression recognition system using neural networks," in *Proc. of International Joint Conference on neural networks*, Vol. 5, pp. 3511-3516, 1999.
- [16] C. W. Chen, "Human face recognition using deformable template and active contour," *Master Thesis*, National Tsing-Hua University, Hsin-Chu, Taiwan, R.O.C., Jun. 1991.
- [17] T. Poggio, H. Voorhees and A.Yuille, "A Regularized Solution to Edge Detection," *Tech. Rep. MA, Rep. AIM-833*, MIT Artificial Intell. Lab., May 1985.
- [18] S. Sarkar and K. L. Boyer, "On Optimal Infinite Impulse Response Edge Detection Filter," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 13, no. 11, pp. 1154-1170, Nov 1991.
- [19] H. Moon, R. Chellappa and A. Rosenfeld, "Optimal Edge-Based Shape Detection," *IEEE Trans. On Image Processing*, Vol.11, no.11, pp. 1209-1227, Nov. 2002.
- [20] J.S. Huang and D.H. Tseng, "Statistical Theory of Edge Detection," *Computer Vision, Graphics, and Image Processing* Vol.43, pp. 337-346, 1988.
- [21] Julez, B., "A Method of Coding TV Signals Based on Edge Detection," *Bell System Tech.* Vol. 38, no. 4, pp.1001-1020. July 1959,
- [22] D.H. Marimont and Y. Rubner, "A Probabilistic Framework for Edge Detection and Scale Selection," *Computer Vision*, 1998. Sixth International Conference on , pp. 207-214, Jan 1998.
- [23] 程正興, 「小波分析演算法與應用」, 西安交通大學出版社, 1997.
- [24] 程正興, 林勇平, 「小波分析在圖像科學中的應用」, *工程數學學報*, 第18卷「小波專刊」pp. 57-86, 2001.

- [25] W. Y. Ma and B. S. Manjunath, "A Technique for Boundary Detection and Image Segmentation," *IEEE Trans. Image Processing*, Vol. 9, no.8, pp. 1375-1388, Aug 2000.
- [26] 王宇生等, 「一種基於積分變換的邊緣檢測方法」, *中國圖形圖像學報*, Vol.7(A), no. 2, pp. 145-149, 2002.
- [27] C. W. Chen, "Human face recognition using deformable template and active contour," *Master Thesis*, National Tsing-Hua University, Hsin Chu, Taiwan, R. O. C., Jun. 1991.
- [28] S. Sarkar and K. L. Boyer, "On Optimal Infinite Impulse Response Edge Detection Filter," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.13, no. 11, pp.1154-1170, Nov. 1991.
- [29] Sheng Tang, *Survey of Edge Detection*,  
[http:// www.cs.unr.edu/~tang\\_s/research/image/surv.p](http://www.cs.unr.edu/~tang_s/research/image/surv.p)
- [30] Robert M. Haralick, FELLOW, "Digital Step Edges from Zero Crossing of Second Directional Derivatives, " *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, no. 1, pp.58-68, Jan. 1984.
- [31] R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," *Computer and Biomedical Research*, Vol.18, pp. 113-125, Jan. 1971.
- [32] L. Sobel, "Camera Models and Machine Perception," *PhD theses*, Stanford University, Standford, CA, 1970.
- [33] J. Prewitt, "Object Enhancement and Extraction, " *Picture Process. Psychopict*, pp.75 - 149, 1970.
- [34] Robinson, J.A. "Efficient general-purpose image compression with binary tree predictive coding, " *IEEE Transactions on Image Processing*, Vol. 6, No.4, Apr. 1997.

- [35] W. Frei and C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Transactions On Electronic Computers*, Vol. C-26, Oct. 1977
- [36] John Canny, Member , " A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, no.1, pp. 679-697, Nov. 1986.
- [37] Julez, B., "A Method of Coding TV Signals Based on Edge Detection," *Bell System Tech.* Vol.38, No.4, pp.1001-1020. July 1959.
- [38] D.C.Marr and E.Hildreth, "Theory of Edge Detection," in *Proc. Int. Roy. Soc. London*, Vol.B275, pp.187-217, 1980.
- [39] John Canny, "Finding Edges and Lines in Images, " MIT Artif. Intell. Lab., Cambridge, MA, *Tech. Rep.* AI-TR-720, 1983.
- [40] I.E. Abdou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement and Thresholding Edge Detectors," in *Proc. Int. IEEE*, Vol.67, no.5, pp. 753-763, May 1979.
- [41] W.K.Pratt, *Digital Image Processing*, New York: Wiley-Interscience, 1978.