

# 第六屆旺宏科學獎 成果報告書

參賽編號：SA6-031

作品名稱：視網膜感光細胞之電腦程式設計分析

姓名：劉康玲

關鍵字：感光細胞、電腦程式、視網膜

# 作品名稱：視網膜感光細胞之電腦程式設計分析

## 摘要

本研究嘗試以自行編輯的電腦程式 (Dev C++ , Builder C++) , 分析比較人類視網膜與電腦螢幕對於色彩的相互關係, 並試圖建立簡易、客觀可將色彩應用於生活中的模式。人體中控制顏色的主要是紅色錐狀細胞(長波)、綠色錐狀細胞(中波)、藍色錐狀細胞(短波)及桿狀細胞(黑白), 由此設計訂出控制顏色的函數:

```
紅色錐狀細胞(長波)      R=0~255
R=Image1->Canvas->Pixels[X][Y]%256;
綠色錐狀細胞(中波)      G=(0~255)*256
G=(Image1->Canvas->Pixels[X][Y]%65536)/256;
藍色錐狀細胞(短波)      B=(0~255)*256*256
B=Image1->Canvas->Pixels[X][Y]/65536;
桿狀細胞(黑白, 灰階)    A*256*256+A*256+A
Image1->Canvas->Pixels[X][Y] =(R+G+B)/3*(256*256+256+1)
```

根據本研究, 可以用數字顯示出視網膜感光細胞的特性, 並可提供使用者或設備(數位相機或印表機)做自行簡易的驗證。透過本研究所設計的圖表及軟體, 可以測出患者是屬於何種類的視覺色彩障礙, 並證實與目前的醫學研究結果一致。另以本研究為基礎, 以 3 項軟體應用實例, 展示不用眼睛“看”是可能的, 而且還有可能“看”的更好: 1. 機器人視覺 (machine vision), 如盲人導航(無人駕駛) 2. 逆向工程 3D 量測 3. 醫學影像處理及量測等。在生物微機電方面 (Micro Electronic Mechanic System , MEMS), 未來終極研究目標則以數位攝影機拍攝提供信號, 再轉換為電流, 刺激視神經或大腦視覺中樞, 希望能使全身癱瘓者透過視網膜意識表達或盲人恢復視覺, 如果能成功, 則代表無須螢幕, 即可傳遞影像訊息至視覺中樞。

## 壹、研究動機

從原始自然界中的兩棲類變色龍及許多軟體動物會隨生活環境中改變體表顏色, 到現在數位世界中電視、電腦、手機、I-pod...等, 彩色螢幕充斥著我們生活週遭, 沒有了彩色, 好像一切都變的很原始、簡陋。究竟彩色螢幕是如何與我們的眼睛溝通呢? 在高一的健康與護理課程中探討眼睛的構造、物理課程中也會提及光線與色彩及資訊課教導程式語言, 因此引發我想將這些不同領域的內容, 整合運用在我的研究上。

## 貳、研究目的

本研究嘗試以自行編輯的電腦程式 (Dev C++ , Builder C++) , 分析比較人類視網膜與電腦螢幕對於色彩的相互關係, 並試圖由此建立簡易、客觀可將色彩應用於研究、醫療、生活中的模式。

## 參、研究設備及器材

系統：Microsoft Windows XP

電腦：Celeron CPU 2.4GHz

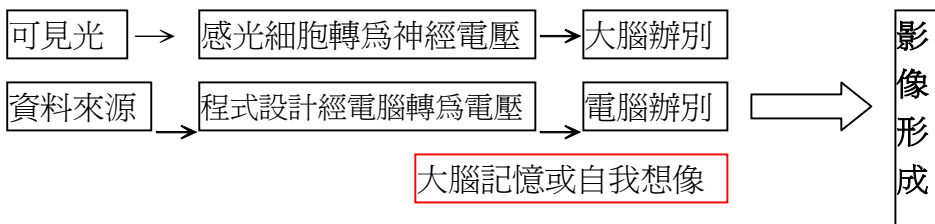
開發軟體：DEV C++，Borland C++ Builder

其他：彩色雷射印表機 數位照相機

## 肆、研究過程及方法

### 一、主要研究過程

試圖將感光細胞感覺的可見光與電腦資料處理形成相同的影像。



### 方法

〈一〉了解視網膜中黑、白、紅、綠、藍等感光細胞的基本結構。

〈二〉電腦程式以視覺化的方式，來分析、呈現視網膜中黑、白、紅、綠、藍等感光細胞的作用，及以本研究為基礎，所發展的應用軟體，展示不用眼睛“看”是可能的，而且還有可能“看”的更好。

#### 1、視覺色彩障礙的檢驗。

2、感光細胞在視網膜內分布的情況及檢視方法。

3、將已設計好之圖樣以雷射印表機（或精密之專業印刷機）以透明投影片印製，覆蓋於電腦螢幕、印好圖樣紙張等上，相互比對檢視差異。

4、不同畫素數位攝影機先拍攝無反光白紙之圖樣，相互比對檢視其差異。

5、撰寫程式設計，模擬視覺色彩障礙者所見的影像。

6、撰寫程式設計，以進行色彩比對測試（color matching）。

7、機器人視覺（machine vision），如盲人導航（無人駕駛）。

8、逆向工程 3D 量測（分別量測 1、5、10、50 元硬幣的厚度）。

9、醫學影像處理及量測。

## 伍、研究結果

### 一、了解視網膜中黑、白、紅、綠、藍等感光細胞的基本結構。

一般來說，人類的視網膜邊緣區具有一億兩千萬個桿狀細胞，中心區具有六百五十萬個錐狀細胞（圖 3）。所以，以眼球的“真實畫素”來說大約是 650 萬畫素至一億兩千萬個畫素，桿狀細胞對感應黑白色極敏感，連微弱到一百萬億分之一瓦特的光線都能看到。而錐狀細胞（圖 6、圖 7）則負責對彩色感覺，對紅綠藍相關之波長最敏感（圖 4），將敏感度

大量集中在黃斑中，也就是調好焦距成像的地方。無論何時，這些感光細胞皆能同時處理 150 萬個訊息。

圖 1 為桿狀細胞（與錐狀細胞的圖類似）不管錐狀細胞或桿狀細胞皆可分為四部分：外段、內段、細胞核與包突結合器。



圖 1

外段：對光敏感的光化學物是在外段發現的。圖 1 中外段之橫線表示細胞膜內許多碟狀物。這些碟狀物如同架子般，光化學物質便是附在這上面。

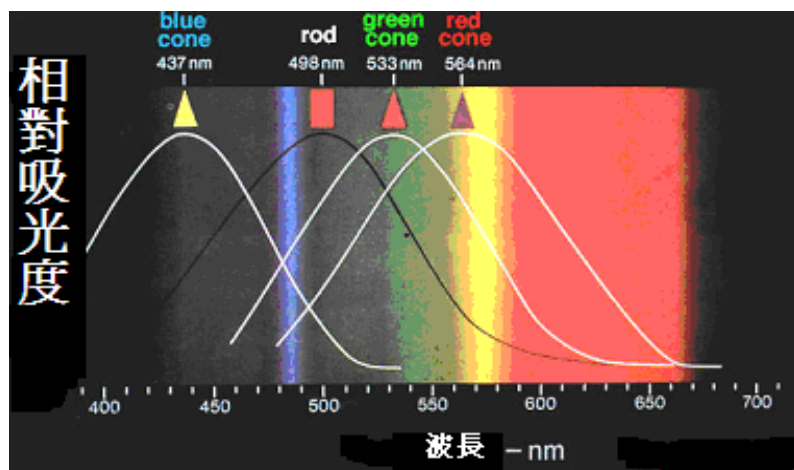
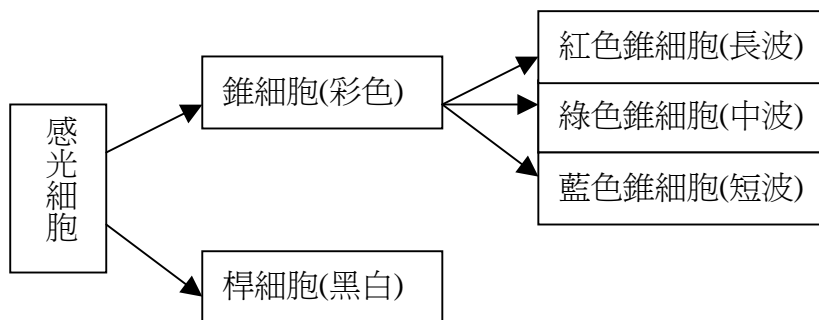


圖 2

人類感光細胞可辨識的波長分部範圍（約為 400~656nm），辨識閾值（discriminate threshold）為 1~2nm。

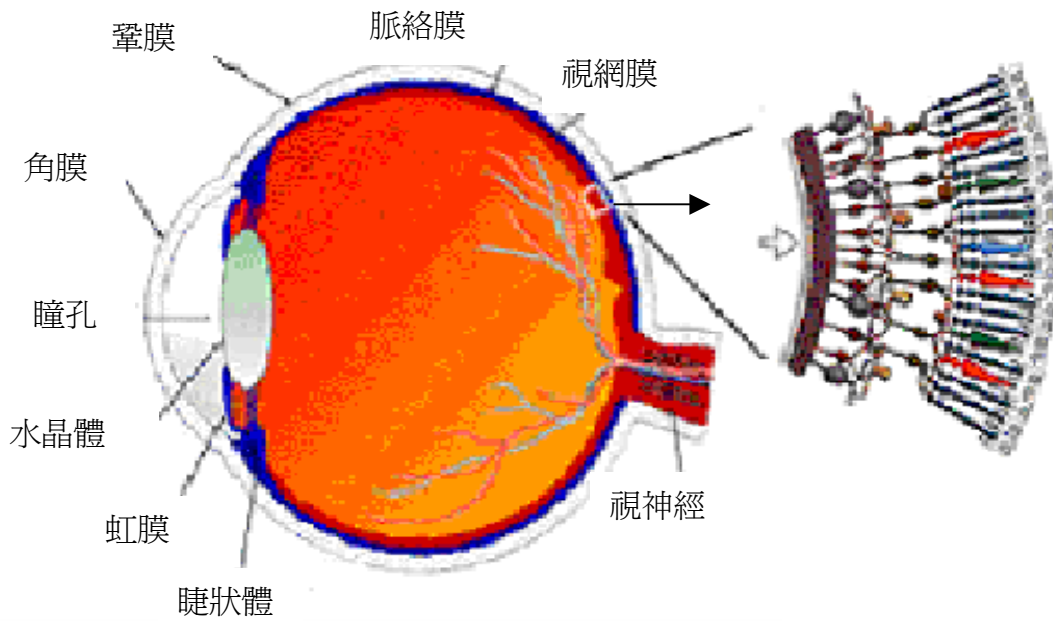


圖 3

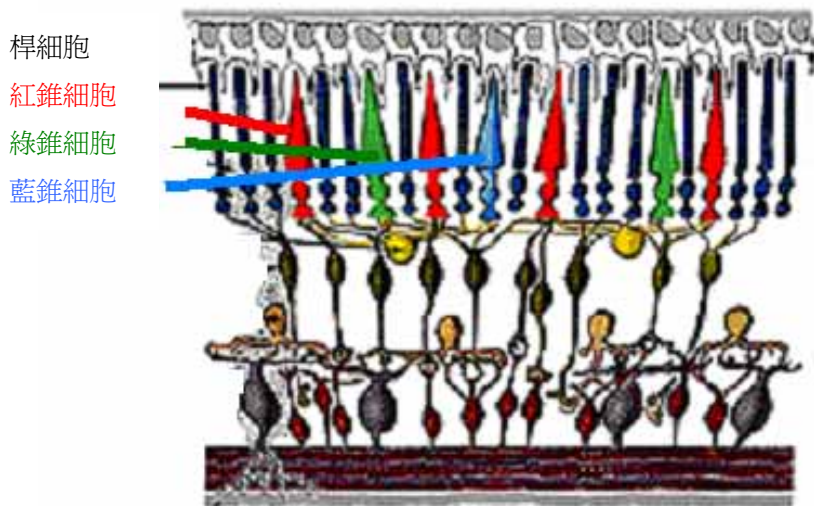


圖 4

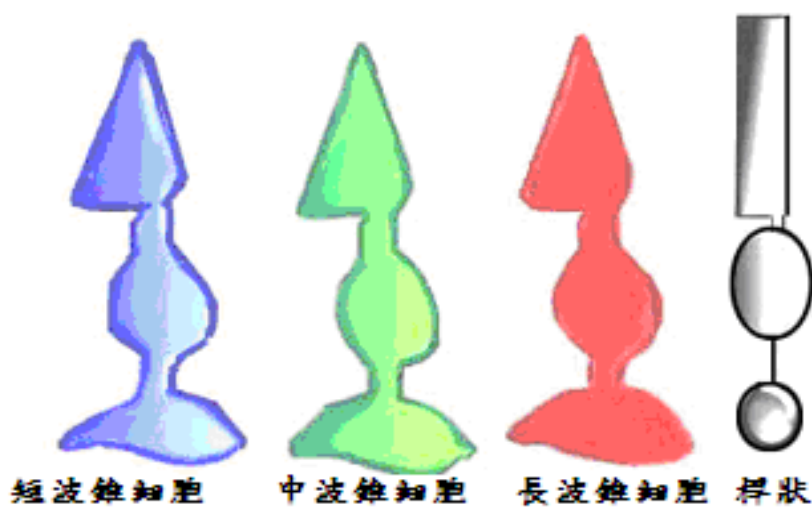


圖 5

視網膜的色素層：色素層內的黑色素(黑素)及脈絡膜內的黑色素可防止光線在眼球內折射，這對敏銳的視力有無比的重要性。這些色素與相機內的黑色塗料具有相同的功能。假

若視網膜中沒有色素的存在，那麼射入進來的光線就會在眼球內朝各個方向反射，如此一來，視網膜上就不會有黑白分明的光點，我們也就看不到精確的影像了。色素層尚貯存有大量的維生素 A。這維生素 A 可透過埋在色素層內的桿細胞和錐細胞之外段細胞而來回交換，而維生素 A 是光敏感色素的一重要前身，所以維生素 A 的交換對於調節光感受器的敏感度是非常重要的。

## 二、電腦程式以視覺化的方式，來分析、呈現視網膜中黑、白、紅、綠、藍等感光細胞的作用。

人體中控制顏色的主要是紅色錐細胞(長波)、綠色錐細胞(中波)、藍色錐細胞(短波)及桿細胞(黑白)，波長大至由 425nm~675nm，約有 250nm 的差距，選 256 是因為最接近的 16 進位數字 (0000EE)，方便程式設計，電腦計算) 由此設計訂出：

- 紅色(Red，簡寫為 R)錐細胞(長波) 值的範圍為 0~255
- 綠色(Green，簡寫為 G)錐細胞(中波) 值的範圍為 (0~255)\*256
- 藍色(Blue，簡寫為 B)錐細胞(短波) 值的範圍為 (0~255)\*256\*256
- 桿細胞 R=G=B 時為灰階，R=G=B=0 為黑色

顏色可由以下公式得出

$Red(0\sim 255) + Green(0\sim 255)*256 + Blue(0\sim 255)*256*256$  只要輸入 256 進位

在程式中，可以利用下列指令畫出該畫素的顏色：

`Image1->Canvas->Pixels[x][y];`

其中 [x][y]值與 Red、Green、Blue 的關係如下：

[x][y] 值除以 256\*256 的商得到 Blue，

由[x][y] 值除以 256\*256 的餘式再除以 256 的商得到 Green，

再由[x][y] 值除以 256 的餘式 Red。

$256*256*256=1677216$  即一般所謂的十六萬色全彩

由此得知各色的比例，再以電腦繪出。詳細程式碼請參考附錄 1。

### 〔一〕基本視覺色彩障礙的檢驗：

將已設計好之圖樣以雷射印表機（或精密之專業印刷機）用無反光白紙印出，並用眼睛觀察受試者對顏色分辨的能力。

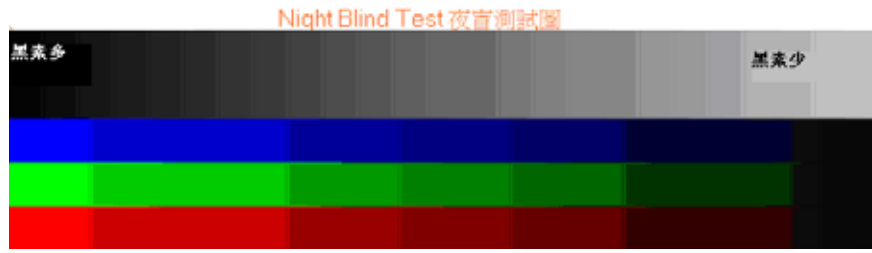
由文獻資料可知視覺分為兩部分：黑白與彩色，分別討論如下：

#### 1. 測試桿狀細胞對黑色白色的敏感性及黑素：(圖 6)

桿狀細胞感光能力不夠時：灰階部份會偏暗，黑白辨識不佳。

黑素不足時：彩色及灰階部份會混沌，色彩辨識不佳。

圖 6



2. 視覺色彩障礙測試的原理及方法：

視覺色彩障礙可能是因為紅綠藍錐狀細胞感光能力缺損、功能的優劣、視覺中樞、視覺神經異常所引起的，其原因有可能是遺傳性、後天病變或意外傷害造成的。根據 Trichromatic Theory 三色理論 (RGB)，以自行設計的軟體繪製出圖 7，以降低顏色的飽和度，測試是否有色弱視覺色彩障礙。若沒辦法在兩旁空白欄為中寫出長條和短條的個數，代表某色錐狀細胞感光能力較差。例如：無法辨識出圖 9 中第一行有四條短條，一條長條，則可能是紅綠色彩障礙。無法辨識第二行有兩條短條，兩條長條，則可能是黃藍色彩障礙，實際以一位經醫師鑑定為色弱患者來測試，證實無法分辨出圖 9 中第二行為兩長條，色彩測試圖調查表(附錄 7)。



圖 7

圖 8 為正常人、紅色色弱、紅色色盲錐狀細胞解析度與紅色波長之關係，正常人可達 1~2nm 解析度，紅色色弱者只達 5~15 nm 解析度，紅色色盲者則完全無法感應紅色，圖 8-1 是根據圖 8 資料圖所設計測試圖，經實測可以判別出色弱類，證實色弱是由錐狀細胞解析度不足引起。視覺色彩障礙的種類相當多，患者的症狀不盡相同，說明如表 1、2，進一步的檢測涉及醫療行為需由專業眼科醫師執行。

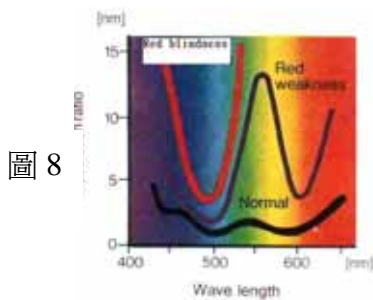


圖 8

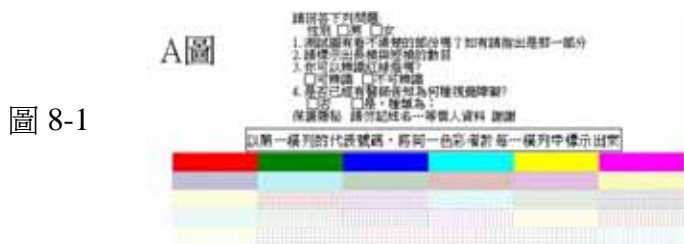


圖 8-1

表 1

| Table1 常見的視覺色彩障礙 |       |
|------------------|-------|
| 視網膜錐形細胞缺乏        | 色盲    |
| 視網膜錐形細胞能力不足      | 色弱    |
| 視網膜黃斑區損害         | 黃藍色缺陷 |
| 視神經異常            | 紅綠色缺陷 |
| 大腦視覺中樞損害         | 色盲    |

表 2

| Table2 Various Color Vision Deficiencies Based on Data in Hunt(1991) |              |              |
|--|--------------|--------------|
| Type   | Male(%)      | Female(%)    |
| <b>Protanopia</b>  | <b>1.0</b>   | <b>0.02</b>  |
| <b>Deutanopia</b>  | <b>1.1</b>   | <b>0.01</b>  |
| <b>Trianopia</b>   | <b>0.002</b> | <b>0.001</b> |
| <b>Cone monochromatism</b>   | <b>~ 0</b>   | <b>~ 0</b>   |
| <b>Rod monochromatism</b>  | <b>0.003</b> | <b>0.002</b> |
| <b>Protanomaly</b>   | <b>1.0</b>   | <b>0.02</b>  |
| <b>Deuteranomaly</b>   | <b>4.9</b>   | <b>0.38</b>  |
| <b>Trianomaly</b>  | <b>~ 0</b>   | <b>~ 0</b>   |
| <b>Total</b>   | <b>8.0</b>   | <b>0.4</b>   |

(二) 感光細胞在視網膜內分布的情況及檢視方法

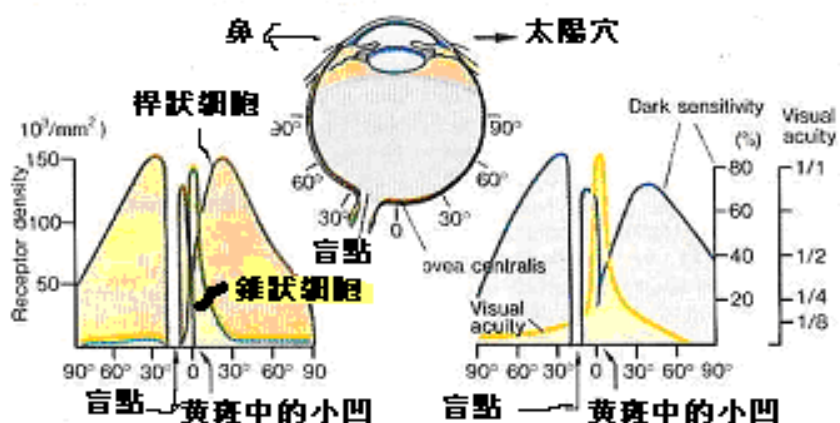


圖 9

由圖 9 中得知感光細胞中的錐狀細胞(紅、綠、藍)多集中在黃斑位置上(即 0 度附近)，桿狀細胞則分布較廣，可達 90 度，但大部分集中在 60 度之內，即角度的重要性是不置可否的。因此以此設計出此圖 10，所以人們可看見的黑白影像範圍比彩色影像的範圍廣的多。

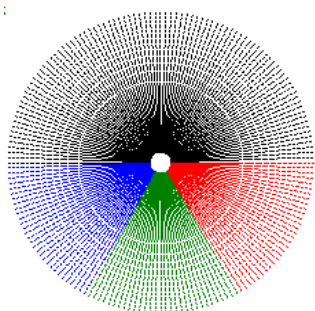


圖 10

錐狀細胞異常時，無法清晰看出圖 11 的細線與色彩，若是視網膜聚焦不正確時，



僅無法清晰看出細線但仍可辨識色彩。

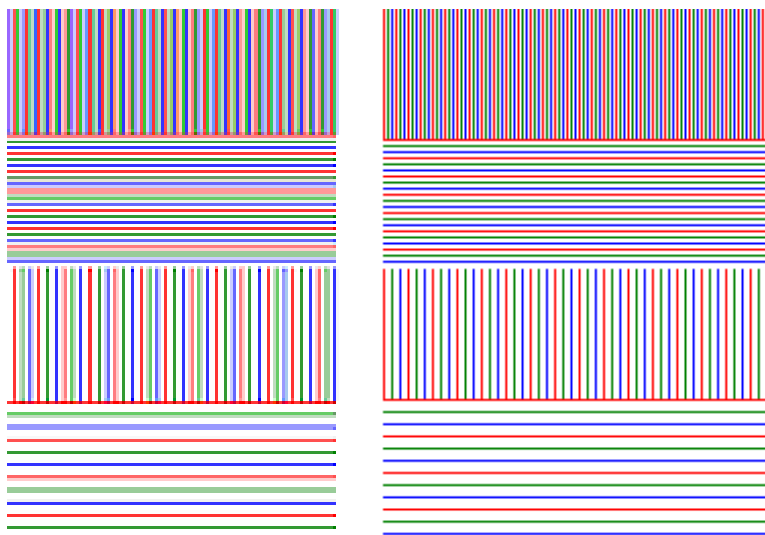
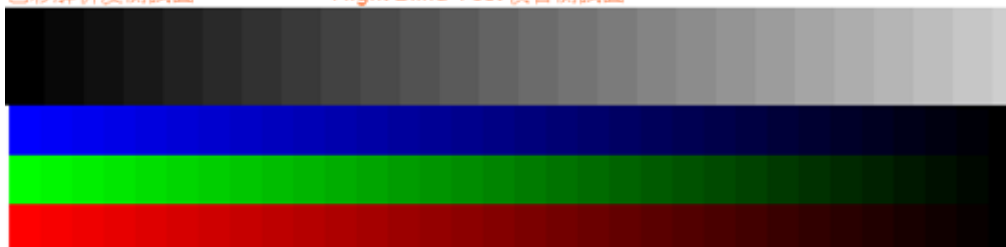


圖 11

〔三〕將已設計好之圖樣〔圖 12〕以雷射印表機（或精密之專業印刷機）以透明投影片印製。有圖樣透明投影片覆蓋於電腦螢幕上或其他的印表機印製成品，相互比對檢視差異。



圖 12



〔四〕以數位攝影機先拍攝無反光白紙之圖樣〔圖 13〕

以同一螢幕或印表機與透明投影片，以不同印表機或螢幕與透明投影片，相互比對檢視

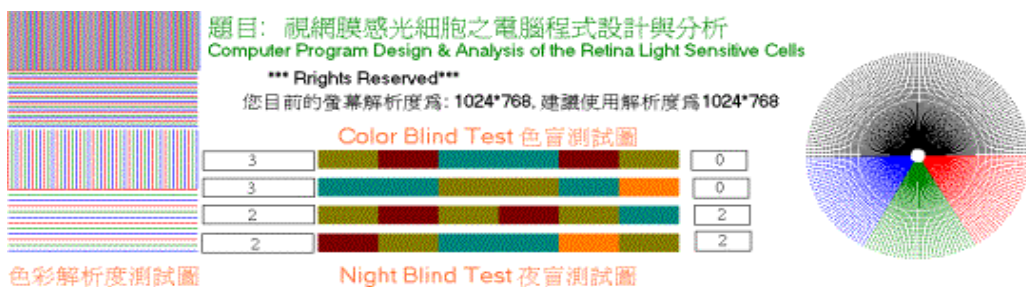
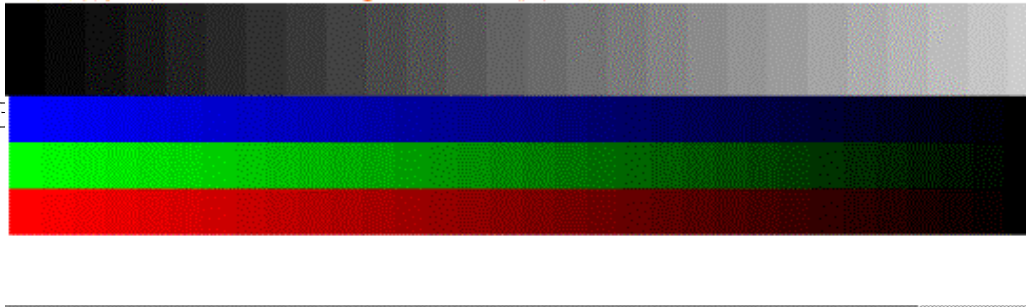


圖 13

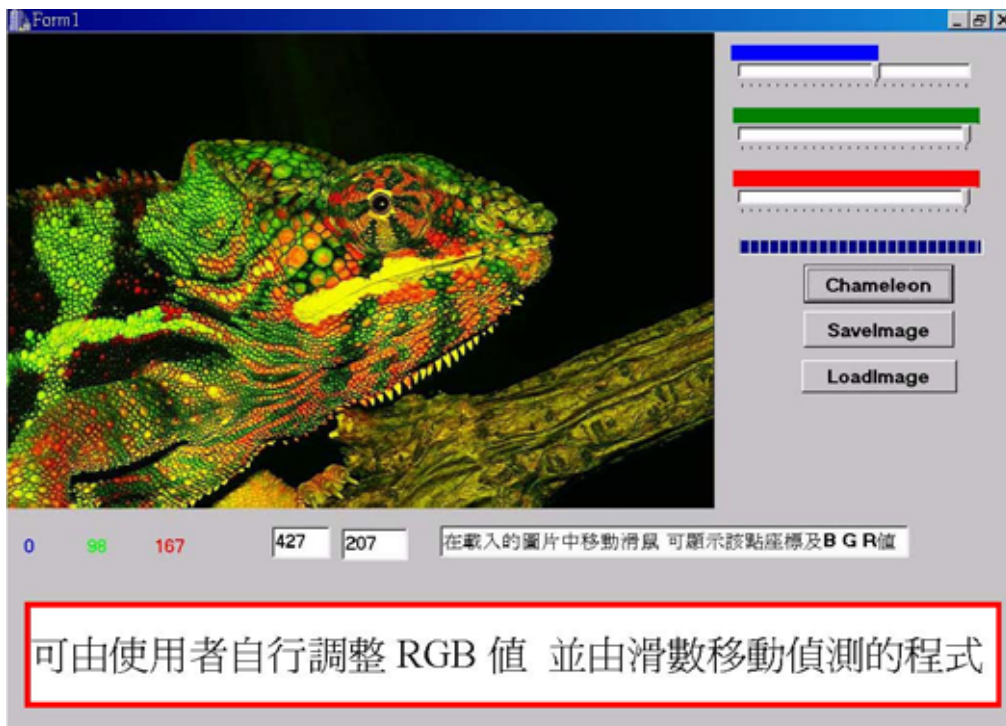


〔三〕

缺乏紅色錐狀細胞為紅色色盲，所以如果把一張圖片中紅色部分移除，可能會是完全缺乏紅色錐狀細胞的紅色色盲看到的影像。以此作為設計程式的概念，以下為試圖模擬各種缺乏紅、綠或藍色色盲看到的影像。程式碼詳見附錄二。

程式功能：

猜猜看！我是少了什麼顏色的龍？



〔圖 14〕

1. 顏色調整拉桿

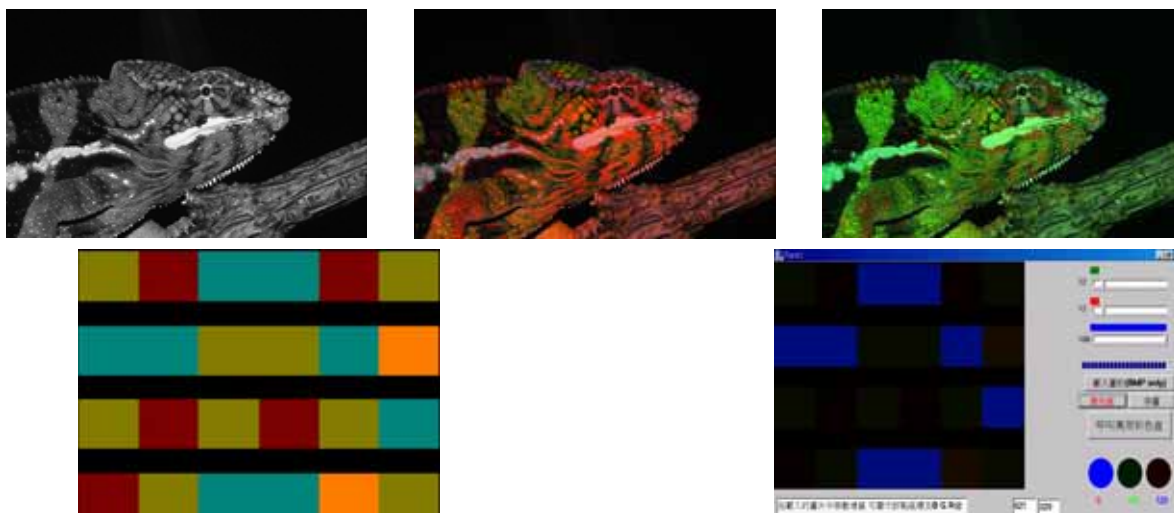
可調整紅綠藍三色的比例，模擬出各種可能。

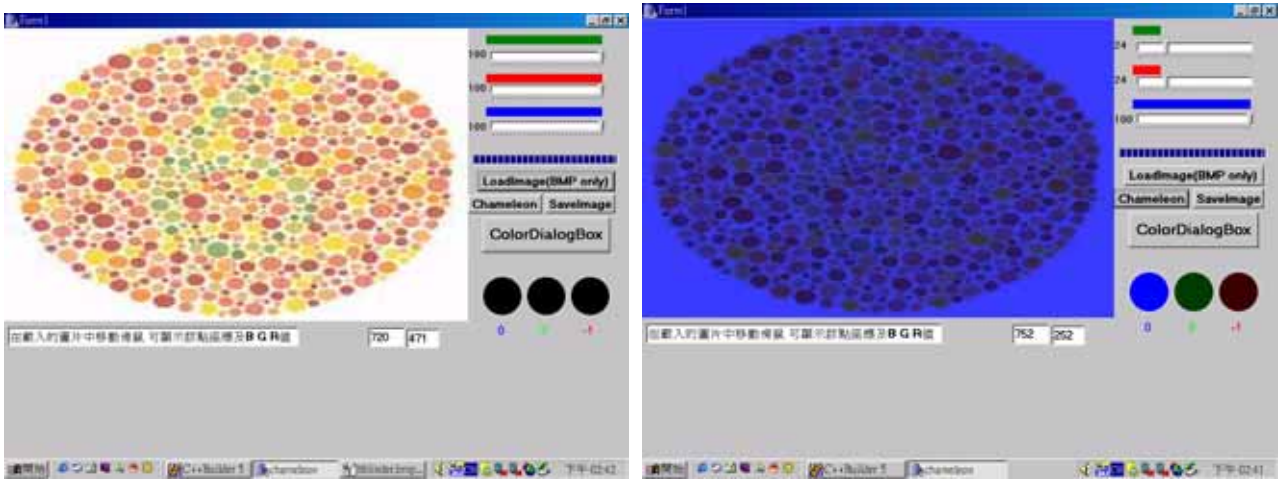
2. 載入圖片 BMP 檔功能（按下 Loadimage）

除了原本提供測試的圖片，亦可自己輸入任何想測試的圖片（限 BMP 檔）。

3. 滑鼠在測試畫面移動時顯示RGB及座標值

當滑鼠在畫面中移動時，程式中的欄位即會顯示出該點的RGB及座標值，此可方便使用者知道此點（看到的顏色），實際上應是何色。





以石原氏圖表 Ishihara Plate 測試的結果 [圖 15]

(六) 色彩比對測試 (Color Matching Experiment) [圖 16]: (程式碼詳見附錄三)

先設定欲測試之 R、G、B 之值，產生欲測試之色帶，隱藏 R、G、B 之值後，由測試者根據欲測試之色帶，自行調整 R、G、B 之值，能調整愈接近測試之色帶者，表示色彩分辨能力佳，無法調整接近測試之色帶者，亦可由其調整結果判讀是何種色彩分辨能力有障礙。本程式除了可測試視覺色彩障礙外，更可進一步訓練調色能力。



[圖 16]

(七) 盲人導航(無人駕駛) [圖 17] (程式碼詳見附錄四)

運用以上的分析，撰寫簡易盲人導航程式，並建立實際運用於生活的模式。將本研究對顏色判別程式設計的成果，應用在協助盲人得知何處有障礙物上。將雷射光照設在地面上，當雷射光碰到障礙物就會曲折，不再呈現一直線，利用此原理，即可根據本研究對顏色判別程式設計的成果，設計出可偵測有障礙物及發出警報聲的應用程式。只要拍攝一張照片如圖 11，將照片掃描存進程式中，令程式偵測紅色的線，並轉換成白色的線（按按鍵BlindGuide），如圖 17，讀取相鄰緊密的兩個點中y軸是否超過五個單位，(若是直線，不易超過此範圍)，若超過，則表示碰到障礙物，所以不是直線，程式此時就會發出警報聲圖 14 之應用程式，未來希望可發展成一類似PDA大小之機器，機頭前處有雷射燈頭可照

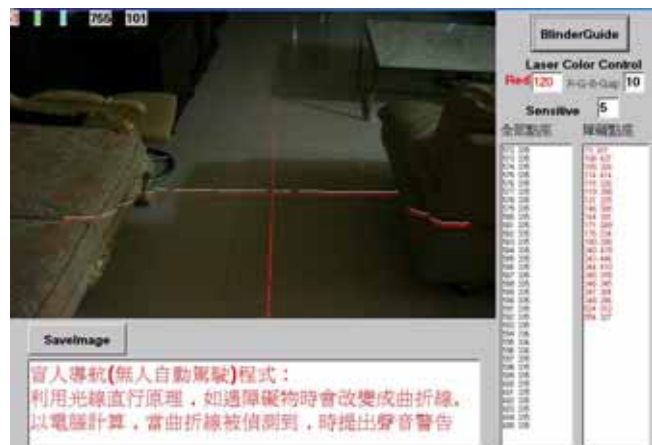
射，機器可將畫面一一拍攝，存入機器中透過此應用程式運算。如此，當盲人在走動時，機器即可將畫面回傳，運算出有障礙物時，可立即發出警戒聲。

\*\*\*\*\*應用本研究的分析，關鍵程式說明\*\*\*\*\*

```

if ((R>StrToInt(Edit1->Text)) & (R>G+StrToInt(Edit2->Text)) & (R>B+StrToInt(Edit2->Text)))
//選出影像中紅、藍、綠值符合要求的區域
{ if((ay-Y)>StrToInt(Edit3->Text)|(Y-ay)>StrToInt(Edit3->Text))
//變化過多處，發出警告聲音
{ Image1->Canvas->Pixels[X][Y-5]=clYellow;
//標示為警示區
sndPlaySound("ding.wav", SND_SYNC);
//另一側變化過多處，發出警告聲音

```



〔圖 17〕

#### (八) 逆向工程 3D量測 (原理與盲人導航同) 〔圖 18〕 (程式碼詳見附錄五)

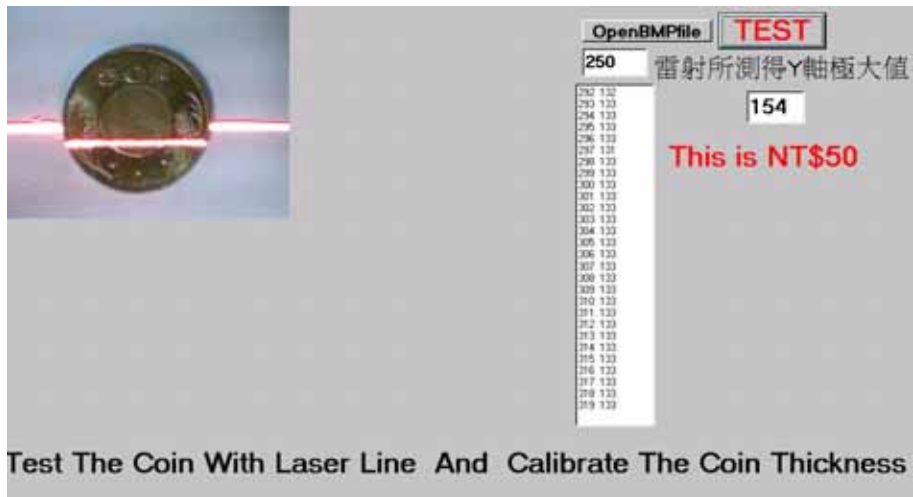
將雷射光照設在硬幣上，當雷射光碰到不同厚度的硬幣就會產生不同的曲折，比較計算所產生的不同曲折就可辨識不同種類的硬幣，同理可對未知物體測試之長、寬、高，此即逆向工程 3D 量測的應用。

\*\*\*\*\*應用本研究的分析，關鍵程式說明\*\*\*\*\*

```

// 選出影像中紅、藍、綠值符合要求的區域，並比較出極大值
Image1->Canvas->Pixels[X][Y-2]=clRed;
Memo1->Lines->Add(FloatToStr(X)+" "+FloatToStr(Y));
    if(M<Y)      { M=Y; break; }
    ax=X; ay=Y; break;    } }
//根據測出的極大值，標示出硬幣種類
Edit4->Text=IntToStr(M);          if(StrToInt(Edit4->Text)==154)
    {Label1->Caption="This is NT$50"; }
if(StrToInt(Edit4->Text)==152)
    {Label1->Caption="This is NT$1";}
if(StrToInt(Edit4->Text)==151)
    {Label1->Caption="This is NT$5"; }
if(StrToInt(Edit4->Text)==153)
    {Label1->Caption="This is NT$10"; } }

```



〔圖 18〕

(九) 醫學影像處理及量測〔圖 19〕(程式碼詳見附錄六)

一般醫用放射線與雷射相同為單高頻率，極為適合直接數位影像處理，即便是以化學藥劑處理之醫用放射影像底片，亦可先轉換為數位影像，再進行數位測量。臨床應用以測量顏面骨骼生長曲線為例，先以目視找出 Sella center of Hypophyseal Forssa (假想腦垂體的中心點，一般以此點為基準點)，再以程式設計找出數位影像中顏面骨骼的輪廓，如此便可計算出欲測量點與基準點的距離及角度，醫師可用來判斷骨骼生長情況或整形手術時骨骼須如何修補。

\*\*\*\*應用本研究的分析，關鍵程式說明\*\*\*\*

//醫用放射影像底片數位處理，使不受化學藥劑等外界因素影響

R=(Image1->Canvas->Pixels[X][Y]%256;

G=(Image1->Canvas->Pixels[X][Y]%65536)/256;

B=(Image1->Canvas->Pixels[X][Y]/65536);

A=(R+G+B)/3;

Image2->Canvas->Pixels[X][Y]=A\*256\*256+A\*256+A; }

A=(R+G+B)/3;

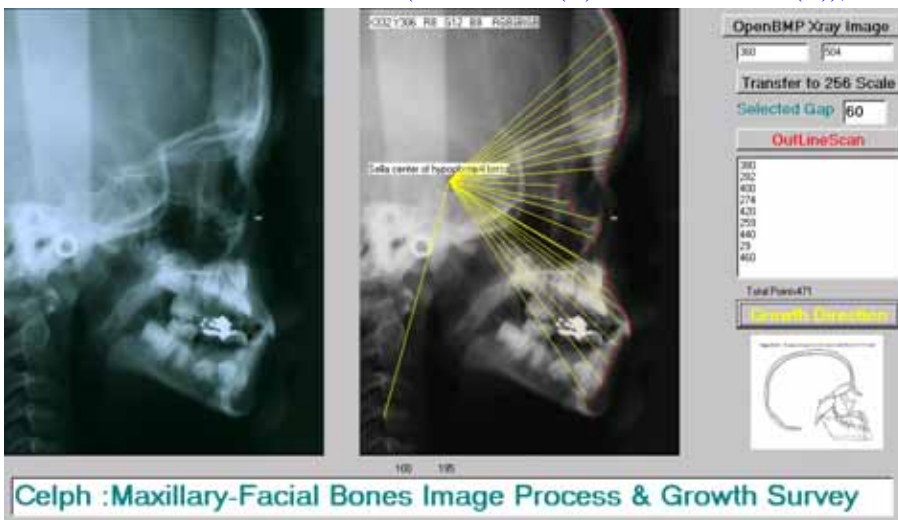
//掃描出欲測量或欲診斷之區域

if (A>gap)

{ Image2->Canvas->Pixels[X][Y]=clRed;

Memo1->Lines->Add("X"+IntToStr(X)+" Y"+IntToStr(Y));

break; } }



〔圖 19〕

## 陸、討論

由於用相機拍攝各種不同畫素相同的圖片後(圖 10)，發現相同的圖片用不同的畫素拍出，顏色有差異(圖 11)；代表不同眼睛(錐狀、桿狀細胞感光能力不同)的人，所看到的顏色有差異。

本研究實驗結果可驗證當錐狀、桿狀細胞感光能力不足時，所看見的影像顏色較不清楚。桿狀細胞感光能力不足時，灰階部份會偏暗，黑白辨識不佳。黑素不足時，彩色及灰階部份會混沌，色彩辨識不佳。也符合圖 12、13 設計的錐狀細胞分布密度，會造成顏色解析度的差異，若密度不足時，無法清晰看出圖 13 的彩色細線。

圖 8-1 是根據圖 8 資料圖所設計測試圖，經實測可以判別出色弱類，證實色弱是由錐狀細胞解析度不足引起，與色盲原因不同。

根據實驗(五)所測出的錐狀細胞感光能力，調整目前視覺色彩障礙眼鏡抑制黃光的多寡，可進一步推廣及應用視覺色彩障礙眼鏡，打造出專屬個人的矯正視覺色彩障礙眼鏡。經與台北馬偕醫院資深眼科主治醫師討論，因為此屬於醫療行為，進一步須經由專業的眼科醫師另擬定實驗計畫。同時，也可將生活中各種圖片先用此程式測試，了解體驗色彩障礙者的世界。

絕大部份的視覺色彩障礙屬於先天基因遺傳，目前醫學上是無法治療的，未來希望能以現代數位科技配合臨床醫學，提供一種可行的視覺輔助方式，更希望借此裝置，可直接將影像訊息傳至視覺中樞，如此即可取代螢幕裝置了。

至於視網膜產生的電位差傳至大腦皮質，可產生視覺腦波  $\alpha$ 、 $\beta$  波，但實際上，腦波與人體、週邊的雜訊相比，非常微弱，差距可達一千倍，不易偵測，以往利用嚴格的訓練、集中注意力、明暗或強烈對比顏色等產生足以偵測到的視覺  $\alpha$ 、 $\beta$  腦波，此項技術已應用於全身癱瘓者意識表達之研究，且經過不斷努力研究，目前現階段判讀正確率達九成二，平均四、五秒就可以辨識一個指令，已十分具實用價值。

## 柒、結論

根據本研究，藉由設計電腦程式，經硬體顯示，可以數字表示眼睛的視網膜感光細胞的特性，透過本研究所設計的圖表〔圖 9〕及軟體，可以更進一步的測出是屬於何種視覺色彩障礙。

根據本研究也可將色彩簡單地應用於生活中。如：若照片在電腦螢幕上出現與相機內不同的顏色時，可以用本研究先前列印的透明片貼在螢幕上做比對、調整，藉此有一套精準的顏色可做比對，列印或顯示出來的圖片就可避免嚴重色差。

以本研究為基礎，可發展機器人視覺 (machine vision)，如盲人導航(無人駕駛)、逆向工

程 3D 量測、醫學影像處理及量測等，展示不用眼睛“看”是可能的，而且還有可能“看”的更好。

以本研究為基礎，更可嘗試研究兩棲類變色龍及許多軟體動物，如何隨生活環境中改變顏色；亦可發展機器人視覺，在精密電子機械製造業上有極為重要廣泛的應用；在生物微機電方面，以矽晶片視網模代替數位攝影機拍攝提供信號，轉換為電流，刺激視神經或大腦視覺中樞，再整合太陽能電池發電，成為視網膜晶片。

## 捌、未來展望

1894 人們開始使用留聲機（phonograph）聽音樂時，有誰會料到 2004 時，會是用 iPod？同樣的人們使用的視覺設備也從黑白電視、彩色電視 進入虛擬實境，再來是什麼呢？（i Vision？）

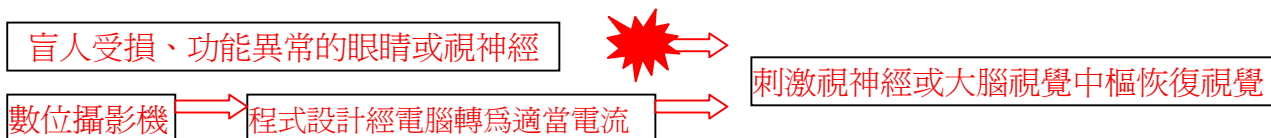
1857 留聲機 phonograph

⇒ 2004 i Pod

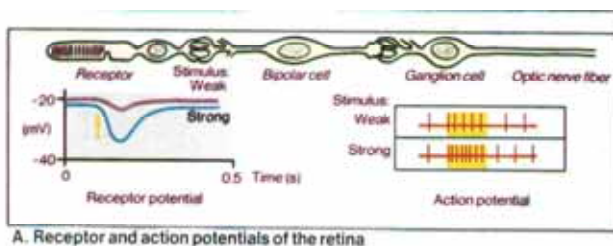
黑白電視 彩色電視 虛擬實境

⇒ 再來是什麼？（i Vision？）

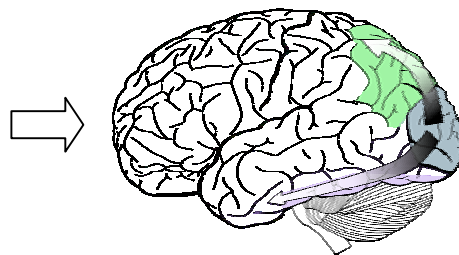
未來終極研究目標則期使用生物微機電方式將圖 21 數位攝影機所獲得信號，以類圖 20 方式傳送，希望能使盲人恢復視覺或發展輔助盲人視覺儀器，若能成功，將對盲人有很大的貢獻，正常人也無須螢幕，可直接將影像訊息傳至視覺中樞。



〔圖 20〕



〔圖 21〕



## 玖、參考資料及其他

### 一、參考資料

1. 楊宗哲 (民 95)。基礎物理，(第 110, 112 頁)。台北市：全華科技。
2. 洪維恩 (民 95)。C++ 教學手冊。台北市：博碩文化。
3. 色盲者如何看世界？ (民 95)。民國 96 年 2 月 26。取自：<http://blog.lezplay.net/?p=1208>
4. Despopoulos/Silbernagl(2003). Color Atlas of Physiology. Georg Thieme Verlag.
5. Chamelon Image, AquaPets, 45。
6. Mostafa(2006). How I (color blind person) see the world?, Retrieved March 1, 2007, from <http://spellcoder.com/blogs/mostafa/archive/2006/08/20/317.aspx>.
7. University of Utah The Organization of the Retina and Visual system. Retrieved December 12, 2006, from <http://webvision.med.utah.edu/>
8. Human Color Vision 劉秀鳳
9. Leeson & Leeson Histology
10. Leeson & lesson A Brief Atlas of Histology
11. [Life]==色盲者如何看世界?  
<http://blog.lezplay.net/?p=1208>  
【單篇文章，無作者】  
彰化 大學生物系  
<http://www.bio.ncue.edu.tw/general/ch45/eye.htm>
12. 【單篇文章，無作者】  
奇摩知識 <http://tw.knowledge.yahoo.com/question/?qid=1005021506105>
13. [www.vischeck.com](http://www.vischeck.com) supported by google
14. Color vision - Wikipedia, the free encyclopedia.htm
15. “整合視覺腦波操控智慧型看護系統” 徐國鎧 中央大學電機系教授



# 附錄一

設計電腦程式以視覺化的方式，來分析、說明：視網膜中黑、白、紅、綠、藍等感光細胞的作用。

說明(以下列舉兩個電腦程式語言稍做說明)

例 1. **Image1->Canvas->Font->Color=clGreen;**

代表在一個物件中(Image1)的畫布上(Canvas)，寫上字型(Font)的顏色(Color=clGreen)。

例 2. **for(int x=64;x<121;x+=9)**

```
{ //int s=(x)*(x+1)/250;  
Image1->Canvas->Pen->Color=clRed;  
Image1->Canvas->MoveTo(1,x);  
Image1->Canvas->LineTo(190,x);
```

代表在一個物件中(Image1)的畫布上(Canvas)，將紅色(Color=clRed;)畫筆(Pen)畫在 X 座標 =64、64+9、64+9+9、...、121(for(int x=64;x<121;x+=9))的位置上(MoveTo),並長度要延至 y 座標 190(LineTo(190,x))

```
////////////////////////////////////  
c++ Builder Program of ColorScale  
////////////////////////////////////  
#include <vcl.h>  
#pragma hdrstop  
#include <printers.hpp>  
#include "Unit1.h"  
#include "math.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::FormCreate(TObject *Sender)  
{  
Image1->Canvas->Font->Color=clGreen;  
Image1->Canvas->Font->Size=22;  
Image1->Canvas->TextOutA(200,0, "視網膜感光細胞之電腦程式設計與分析 ");  
Image1->Canvas->Font->Size=14;  
Image1->Canvas->TextOutA(200,30, "Computer Program Design & Analysis of the Retina Light Sensitive Cells ");  
Image1->Canvas->Font->Color=clBlack;  
Image1->Canvas->Font->Size=16;  
Image1->Canvas->TextOutA(260,60 "***All Rights Reserved***");  
Image1->Canvas->TextOutA(230,85, " 您目前的螢幕解析度為: "+  
IntToStr(Screen->Width)+"*"+IntToStr(Screen->Height)+"", 建議使用解析度為 1024*768");  
  
Image1->Canvas->Font->Color=65*256*256+125*256+255;  
Image1->Canvas->Font->Size=22;
```

```
Image1->Canvas->TextOutA(320,120, "Color Blind Test 色盲測試圖 ");
Image1->Canvas->TextOutA(1,270, "色彩解析度測試圖
```

```
Night Blind Test 夜盲測試圖 ");
```

```
//resolution test
```

```
for(int x=1;x<190;x+=6)
```

```
{
```

```
//int s=(x)*(x+1)/250;
```

```
Image1->Canvas->Pen->Color=clRed;
```

```
Image1->Canvas->MoveTo(x,0);
```

```
Image1->Canvas->LineTo(x,64);
```

```
Image1->Canvas->Pen->Color=clGreen;
```

```
Image1->Canvas->MoveTo(x+2,0);
```

```
Image1->Canvas->LineTo(x+2,64);
```

```
Image1->Canvas->Pen->Color=clBlue;
```

```
Image1->Canvas->MoveTo(x+4,0);
```

```
Image1->Canvas->LineTo(x+4,64);
```

```
}
```

```
for(int x=64;x<121;x+=9)
```

```
{
```

```
//int s=(x)*(x+1)/250;
```

```
Image1->Canvas->Pen->Color=clRed;
```

```
Image1->Canvas->MoveTo(1,x);
```

```
Image1->Canvas->LineTo(190,x);
```

```
Image1->Canvas->Pen->Color=clGreen;
```

```
Image1->Canvas->MoveTo(1,x+3);
```

```
Image1->Canvas->LineTo(190,x+3);
```

```
Image1->Canvas->Pen->Color=clBlue;
```

```
Image1->Canvas->MoveTo(1,x+6);
```

```
Image1->Canvas->LineTo(190,x+6);
```

```
}
```

```
for(int x=1;x<190;x+=12)
```

```
{
```

```
Image1->Canvas->Pen->Color=clRed;
```

```
Image1->Canvas->MoveTo(x,128);
```

```
Image1->Canvas->LineTo(x,192);
```

```
Image1->Canvas->Pen->Color=clGreen;
```

```
Image1->Canvas->MoveTo(x+4,128);
```

```
Image1->Canvas->LineTo(x+4,192);
```

```
Image1->Canvas->Pen->Color=clBlue;
```

```
Image1->Canvas->MoveTo(x+8,128);
```

```
Image1->Canvas->LineTo(x+8,192);
```

```
}
```

```
for(int x=192;x<248;x+=18)
```

```
{
```

```
//int s=(x)*(x+1)/250;
```

```
Image1->Canvas->Pen->Color=clRed;
```

```
Image1->Canvas->MoveTo(1,x);
```

```
Image1->Canvas->LineTo(190,x);
```

```
Image1->Canvas->Pen->Color=clGreen;
```

```
Image1->Canvas->MoveTo(1,x+6);
```

```
Image1->Canvas->LineTo(190,x+6);
```

```
Image1->Canvas->Pen->Color=clBlue;
```

```
Image1->Canvas->MoveTo(1,x+12);
```

```
Image1->Canvas->LineTo(190,x+12);
```

```
}
```

```

//Color blind test
//red
for(int x=220;x<310;x++)
{
    for(int y=150;y<170;y++)
    {
        Image1->Canvas->Pixels[x][y]=clRed;
    }
    for(int x=490;x<580;x+=2)
    {
        for(int y=170;y<190;y+=2)
        {
            Image1->Canvas->Pixels[x][y]=clRed;
        }
    }
    for(int x=310;x<400;x+=3)
    {
        for(int y=190;y<210;y+=3)
        {
            Image1->Canvas->Pixels[x][y]=clRed;
        }
    }
    for(int x=670;x<760;x+=4)
    {
        for(int y=210;y<230;y+=4)
        {
            Image1->Canvas->Pixels[x][y]=clRed;
        }
    }
    for(int x=490;x<580;x+=5)
    {
        for(int y=230;y<250;y+=5)
        {
            Image1->Canvas->Pixels[x][y]=clRed;
        }
    }
}

//green
for(int x=310;x<400;x++)
{
    for(int y=150;y<170;y++)
    {
        Image1->Canvas->Pixels[x][y]=clGreen;
    }
}
for(int x=400;x<490;x+=2)
{
    for(int y=170;y<190;y+=2)
    {
        Image1->Canvas->Pixels[x][y]=clGreen;
    }
}
for(int x=580;x<670;x+=3)
{
    for(int y=190;y<210;y+=3)
    {
        Image1->Canvas->Pixels[x][y]=clGreen;
    }
}
for(int x=310;x<400;x+=4)
{
    for(int y=210;y<230;y+=4)
    {
        Image1->Canvas->Pixels[x][y]=clGreen;
    }
}
for(int x=580;x<670;x+=5)
{
    for(int y=230;y<250;y+=5)
    {
        Image1->Canvas->Pixels[x][y]=clGreen;    } }

//blue
for(int x=400;x<490;x++)
{
    for(int y=150;y<170;y++)
    {
        Image1->Canvas->Pixels[x][y]=clBlue;    } }
for(int x=220;x<310;x+=2)
{
    for(int y=170;y<190;y+=2)
    {
        Image1->Canvas->Pixels[x][y]=clBlue;    } }
for(int x=670;x<760;x+=3)
{
    for(int y=190;y<210;y+=3)
    {
        Image1->Canvas->Pixels[x][y]=clBlue;    } }
for(int x=400;x<490;x+=4)
{
    for(int y=210;y<230;y+=4)
    {
        Image1->Canvas->Pixels[x][y]=clBlue;
    }
}
for(int x=310;x<400;x+=5)
{
    for(int y=230;y<250;y+=5)
    {
        Image1->Canvas->Pixels[x][y]=clBlue;
    }
}

```

```

    } }
//no red
for(int x=490;x<580;x++)
{   for(int y=150;y<170;y++)
    {   Image1->Canvas->Pixels[x][y]=255*256*256+255*256;    } }
    for(int x=310;x<400;x+=2)
    {   for(int y=170;y<190;y+=2)
        {   Image1->Canvas->Pixels[x][y]=255*256*256+255*256;    } }
    for(int x=490;x<580;x+=3)
    {   for(int y=190;y<210;y+=3)
        {   Image1->Canvas->Pixels[x][y]=255*256*256+255*256;    }}
    for(int x=220;x<310;x+=4)
    {   for(int y=210;y<230;y+=4)
        {   Image1->Canvas->Pixels[x][y]=255*256*256+255*256;    }}
    for(int x=670;x<760;x+=5)
    {   for(int y=230;y<250;y+=5)
        {   Image1->Canvas->Pixels[x][y]=255*256*256+255*256;    } }

//no blue
for(int x=580;x<670;x++)
{   for(int y=150;y<170;y++)
    {   Image1->Canvas->Pixels[x][y]=255*256+255;    } }
    for(int x=670;x<760;x+=2)
    {   for(int y=170;y<190;y+=2)
        {   Image1->Canvas->Pixels[x][y]=255*256+255;    } }
    for(int x=220;x<310;x+=3)
    {   for(int y=190;y<210;y+=3)
        {   Image1->Canvas->Pixels[x][y]=255*256+255;    } }
    for(int x=580;x<670;x+=4)
    {
        for(int y=210;y<230;y+=4)
        {   Image1->Canvas->Pixels[x][y]=255*256+255;    } }
        for(int x=220;x<310;x+=5)
        {   for(int y=230;y<250;y+=5)
            {   Image1->Canvas->Pixels[x][y]=255*256+255;    } }
    }

//no green
for(int x=670;x<760;x++)
{   for(int y=150;y<170;y++)
    {   Image1->Canvas->Pixels[x][y]=255*256*256+255;    } }
    for(int x=580;x<670;x+=2)
    {
        for(int y=170;y<190;y+=2)
        {   Image1->Canvas->Pixels[x][y]=255*256*256+255; } }
        for(int x=400;x<490;x+=3)
        {   for(int y=190;y<210;y+=3)
            {   Image1->Canvas->Pixels[x][y]=255*256*256+255; } }
        for(int x=490;x<580;x+=4)
        {   for(int y=210;y<230;y+=4)
            {   Image1->Canvas->Pixels[x][y]=255*256*256+255; } }
        for(int x=400;x<490;x+=5)
        {   for(int y=230;y<250;y+=5)
            {   Image1->Canvas->Pixels[x][y]=255*256*256+255; } }
    }

//cone & rod density test
for(int r=1;r<150;r++)
{ for(int a=90;a<180;a++)
    { double s=r*(r+1)/200*cos(3.1415926/180*a*2);
      double t=r*(r+1)/200*sin(3.1415926/180*a*2);
      Image1->Canvas->Pixels[908+s][156+t]=clBlack/r; }
    for(int a=60;a<90;a++)
    { double s=r*(r+1)/200*cos(3.1415926/180*a*2);

```

```

double t=r*(r+1)/200*sin(3.1415926/180*a*2);
Image1->Canvas->Pixels[908+s][156+t]=clBlue; }
for(int a=0;a<30;a++)
{ double s=r*(r+1)/200*cos(3.1415926/180*a*2);
double t=r*(r+1)/200*sin(3.1415926/180*a*2);
Image1->Canvas->Pixels[908+s][156+t]=clRed; }
for(int a=30;a<60;a++)
{ double s=r*(r+1)/200*cos(3.1415926/180*a*2);
double t=r*(r+1)/200*sin(3.1415926/180*a*2);
Image1->Canvas->Pixels[908+s][156+t]=clGreen; } }
//256 gray scale
for(int x=0;x<Image1->Width;x++)
{ int g=Image1->Width/256+1;//avoid (0~256)/256=0
Image1->Canvas->Pen->Color=x/g*256*256+x/g*256+x/g;
Image1->Canvas->MoveTo(x,300);
Image1->Canvas->LineTo(x,400); }
//color fade
for(int x=4;x<1024;x+=4)
{ Image1->Canvas->Pen->Color=256*256*256-x/4*256*256;
Image1->Canvas->MoveTo(x,400);
Image1->Canvas->LineTo(x,450);
Image1->Canvas->MoveTo(x+1,400);
Image1->Canvas->LineTo(x+1,450);
Image1->Canvas->MoveTo(x+2,400);
Image1->Canvas->LineTo(x+2,450);
Image1->Canvas->MoveTo(x+3,400);
Image1->Canvas->LineTo(x+3,450);
Image1->Canvas->Pen->Color=256*256-x/4*256;
Image1->Canvas->MoveTo(x,450);
Image1->Canvas->LineTo(x,500);
Image1->Canvas->MoveTo(x+1,450);
Image1->Canvas->LineTo(x+1,500);
Image1->Canvas->MoveTo(x+2,450);
Image1->Canvas->LineTo(x+2,500);
Image1->Canvas->MoveTo(x+3,450);
Image1->Canvas->LineTo(x+3,500);
Image1->Canvas->Pen->Color=256-x/4;
Image1->Canvas->MoveTo(x,500);
Image1->Canvas->LineTo(x,550);
Image1->Canvas->MoveTo(x+1,500);
Image1->Canvas->LineTo(x+1,550);
Image1->Canvas->MoveTo(x+2,500);
Image1->Canvas->LineTo(x+2,550);
Image1->Canvas->MoveTo(x+3,500);
Image1->Canvas->LineTo(x+3,550); } }
//-----
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{ //PrintDialog1->Execute();
PrinterSetupDialog1->Execute();
TRect ImgRect= Rect(0, 0, Image1->Width, Image1->Height);
//float r=ComboBox1->ItemIndex;
int r=StrToInt (Edit1->Text);
TRect rate= Rect(0,0,Image1->Width*(r+1),Image1->Height*(r+1));
Printer()->BeginDoc();
Printer()->Canvas->CopyRect(rate,Image1->Canvas,ImgRect);
Printer()->EndDoc();}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{ SavePictureDialog1->Execute();
Image1->Picture->SaveToFile(SavePictureDialog1->FileName);}

```

```
//-----
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{ColorDialog1->Execute();
BitBtn3->Font->Color=ColorDialog1->Color;
Image1->Canvas->Pen->Color=ColorDialog1->Color;
for(int x=4;x<1024;x+=3)
    { Image1->Canvas->MoveTo(x,600);
      Image1->Canvas->LineTo(x,550); }
//-----
```

## 附錄 2 Chameleon 變色龍 C++ 主程式碼

```
int X,Y,R,G,B,A;
for (X=0;X<Image1->Width;X++)
    { for (Y=0;Y<Image1->Height;Y++)
      { R=Image1->Canvas->Pixels[X][Y]%256;
        G=(Image1->Canvas->Pixels[X][Y]%65536)/256;
        B=Image1->Canvas->Pixels[X][Y]/65536;
          A=(R+G+B)/3;
        //no Red
          Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-R;
        //no Green
          //Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-G*256;
        //noBlue
          //Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-B*256*256;
        //no GreenRed
          //Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-G*256-R;
        //no BlueRed
          //Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-B*256*256-R;
          //noGreenBlue
        //Image1->Canvas->Pixels[X][Y]=Image1->Canvas->Pixels[X][Y]-B*256*256-G*256;
        //no R G B = Gray Scale
          //Image1->Canvas->Pixels[X][Y]=A*256*256+A*256+A;
        }
    }
}
```

## 附錄 3 色彩比對測試 (Color Matching Experiment) C++ 主程式碼

```
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{}
//-----
void __fastcall TForm1::Edit1Change(TObject *Sender)
{Shape1->Brush->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);
Shape1->Pen->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{Edit1->Visible=true;Edit2->Visible=true; Edit3->Visible=true; }
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{Edit1->Visible=false;Edit2->Visible=false; Edit3->Visible=false;}
//-----
void __fastcall TForm1::Edit2Change(TObject *Sender)
```

```

{Shape1->Brush->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);
Shape1->Pen->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);}
//-----
void __fastcall TForm1::Edit3Change(TObject *Sender)
{Shape1->Brush->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);
Shape1->Pen->Color=StrToInt(Edit1->Text)+256*StrToInt(Edit2->Text)+256*256*StrToInt(Edit3->Text);}
//-----
void __fastcall TForm1::ScrollBar1Change(TObject *Sender)
{Shape2->Brush->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Shape2->Pen->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Edit4->Text=IntToStr(ScrollBar1->Position);}
//-----
void __fastcall TForm1::ScrollBar2Change(TObject *Sender)
{Shape2->Brush->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Shape2->Pen->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Edit5->Text=IntToStr(ScrollBar2->Position);}
//-----
void __fastcall TForm1::ScrollBar3Change(TObject *Sender)
{Shape2->Brush->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Shape2->Pen->Color=StrToInt(ScrollBar1->Position)+256*StrToInt(ScrollBar2->Position)+256*256*StrToInt(ScrollBar3->Position);
Edit6->Text=IntToStr(ScrollBar3->Position);}
//-----

```

## 附錄 4 盲人導航（無人駕駛） C++ 主程式碼

```

{Memo1->Clear();
Memo2->Clear();
int X,Y,R,G,B,ax,ay;
for (X=0;X<Image1->Width;X++)
{for (Y=Image1->Height;Y>0;Y--)
{ R=Image1->Canvas->Pixels[X][Y]%256;
G=(Image1->Canvas->Pixels[X][Y]%65536)/256;
B=Image1->Canvas->Pixels[X][Y]/65536;
if ((R>StrToInt(Edit1->Text)) & (R>G+StrToInt(Edit2->Text)) & (R>B+StrToInt(Edit2->Text)))
{ Image1->Canvas->Pixels[X][Y-5]=clWhite;
if((ay-Y)>StrToInt(Edit3->Text)|(Y-ay)>StrToInt(Edit3->Text)) //發出警告聲音
{ Image1->Canvas->Pixels[X][Y-5]=clYellow;
sndPlaySound("ding.wav", SND_SYNC);
Memo2->Lines->Add(FloatToStr(X)+" "+FloatToStr(Y)); }
Memo1->Lines->Add(FloatToStr(X)+" "+FloatToStr(Y));
ax=X;
ay=Y;
break; } } }

```

## 附錄 5 逆向工程 3D 測量 C++ 主程式碼

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"

```

```

TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
Memo1->Clear();
Image1->Picture->Assign(NULL);
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
int X,Y,R,G,B,ax,ay=0,M=0;
Edit4->Text=IntToStr(0);

for (X=0;X<Image1->Width;X++)
    { for (Y=0;Y<Image1->Height;Y++)
        { R=Image1->Canvas->Pixels[X][Y]%256;
G=(Image1->Canvas->Pixels[X][Y]%65536)/256;
B=Image1->Canvas->Pixels[X][Y]/65536;

if (R>StrToInt(Edit3->Text))
    { Image1->Canvas->Pixels[X][Y-2]=clRed;
        Memo1->Lines->Add(FloatToStr(X)+" " +FloatToStr(Y));
        if(M<Y)
            { M=Y;
                break; }
        ax=X;
        ay=Y;
        break;
    } }
Edit4->Text=IntToStr(M);
    if(StrToInt(Edit4->Text)==154)
        {Label1->Caption="This is NT$50"; }
    if(StrToInt(Edit4->Text)==152)
        {Label1->Caption="This is NT$1";}
    if(StrToInt(Edit4->Text)==151)
        {Label1->Caption="This is NT$5"; }
    if(StrToInt(Edit4->Text)==153)
        {Label1->Caption="This is NT$10"; } }
//-----

void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
OpenPictureDialog1->Execute();
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);

```



```

Edit4->Text=IntToStr(0);
Label1->Caption=" ";}
//-----
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{Image1->Canvas->TextOutA(10,10,"R"+IntToStr(Image1->Canvas->Pixels[X][Y]%256)+"G"+IntToStr(Image1->Ca
nvas->Pixels[X][Y]%65536/256)+"B"+IntToStr(Image1->Canvas->Pixels[X][Y]/65536)+" ");}
//-----

```

## 附錄 6 醫學影像處理及測量 C++ 主程式碼

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{}
//-----
int sx,sy;
int R,G,B,A;
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{BitBtn4->Enabled=False;
OpenPictureDialog1->Execute();
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
//Image1->Canvas->TextOutA(10,10,Edit1->Text+Edit2->Text+Edit3->Text+Edit4->Text+Edit5->Text);
Edit4->Text=IntToStr(Image1->Picture->Width);
Edit5->Text=IntToStr(Image1->Picture->Height);
Image1->Height=Image1->Picture->Height;
Image1->Width=Image1->Picture->Width;
Image2->Canvas->TextOutA(10,10," Celph for MaxilFacial Bones");}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{int A,X,Y,R,G,B,R1,B1,G1,A1;
Memo1->Clear();
Image1->Picture->Assign(NULL);
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
int gap=StrToInt(Edit1->Text);
// Right to left scan
for (int Y=0;Y<504;Y++)
{
    for (int X=355;X>0;X--)

```

```

    { R=Image2->Canvas->Pixels[X][Y]%256;
      G=(Image2->Canvas->Pixels[X][Y]%65536)/256;
      B=(Image2->Canvas->Pixels[X][Y]/65536);
      A=(R+G+B)/3;
      if ( A>gap )
      {
        Image2->Canvas->Pixels[X][Y]=clRed;
        Memo1->Lines->Add("X"+IntToStr(X)+" Y"+IntToStr(Y));
        break;      }  }}
Label4->Caption=" Total Points"+IntToStr(Memo1->Lines->Count);}
//-----
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{int X,Y,R,G,B,A;
for (X=0;X<Image1->Width;X++)
  {for (Y=0;Y<Image1->Height;Y++)
    {   R=Image1->Canvas->Pixels[X][Y]%256;
        G=(Image1->Canvas->Pixels[X][Y]%65536)/256;
        B=(Image1->Canvas->Pixels[X][Y]/65536);
        A=(R+G+B)/3;
        Image2->Canvas->Pixels[X][Y]=A*256*256+A*256+A;   }
      Image2->Height=Image1->Height;
Image2->Width=Image1->Width;}}
//-----
void __fastcall TForm1::Image2MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{Image2->Canvas->TextOutA(10,10,"X"+IntToStr(X)+" Y"+IntToStr(Y)+"
R"+IntToStr(Image2->Canvas->Pixels[X][Y]%256)+"
G"+IntToStr((Image2->Canvas->Pixels[X][Y]%65536)/256)+"
B"+IntToStr(Image2->Canvas->Pixels[X][Y]/65536)+" RGB");}
//-----
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
{Image1->Picture->Assign(NULL);
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
Image2->Canvas->Pen->Color=clYellow;
int gap=StrToInt(Edit1->Text);
// Right to left scan
for (int Y=0;Y<504;Y+=20)
  {   for (int X=355;X>0;X--)
      { R=Image2->Canvas->Pixels[X][Y]%256;
        G=(Image2->Canvas->Pixels[X][Y]%65536)/256;
        B=(Image2->Canvas->Pixels[X][Y]/65536);
        A=(R+G+B)/3;
        if ( A>gap )
        { Image2->Canvas->Pixels[X][Y]=clRed;
          Memo1->Lines->Add(IntToStr(X));
        }
      }
    }
}

```

```

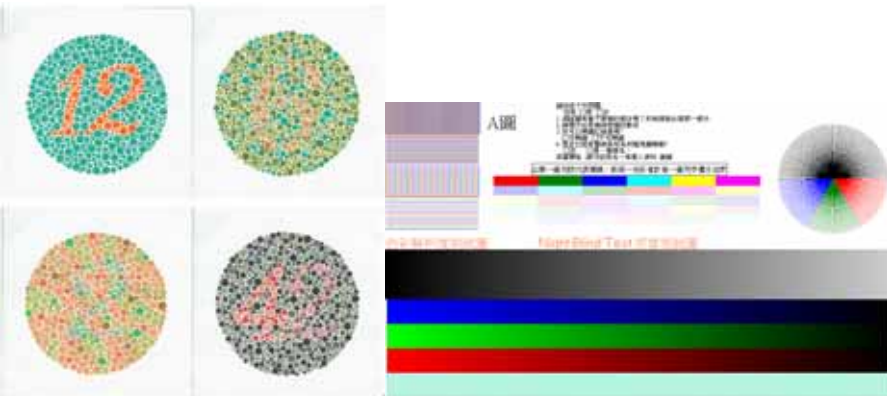
Memo1->Lines->Add(IntToStr(Y));
Image2->Canvas->MoveTo(StrToInt(Label2->Caption),StrToInt(Label3->Caption));
Image2->Canvas->LineTo(X-2,Y);
break;          }   }}}
//-----
void __fastcall TForm1::Image2MouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{Image2->Canvas->Pen->Color=clRed;
Image2->Canvas->MoveTo(X+3,Y);
Image2->Canvas->LineTo(X-3,Y);
Image2->Canvas->MoveTo(X,Y+3);
Image2->Canvas->LineTo(X,Y-3);
Label2->Caption=IntToStr(X);
Label3->Caption=IntToStr(Y);
sx=X;
sy=Y;
Image2->Canvas->TextOutA(10,Y-20,"Sella center of hypophyseal fossa");
BitBtn4->Enabled=True;}
//-----

```

## 附錄7色彩測試圖調查表

**視覺色彩異常測試步驟:**

1. 是否能指出石原氏圖譜中的數字。
2. 如能指出圖譜中的全部數字，則無須進一步測試。
3. 如無法指出圖譜中的全部數字，請進一步測試 A、B 附圖。
4. 測試 A、B 附圖，依照 A、B 附表中的說明填入。
5. 請務必保護個人隱私，切勿記錄姓名...等個人資料，謝謝。
6. 請參考 A、B 附表樣本。




The form contains two main sections: 'A 表' (Table A) and 'B 圖' (Figure B). Each section includes a color calibration chart and a table for recording test results. The tables have columns for different color categories and rows for individual test items.