

中華民國第四十七屆中小學科學展覽會  
作品說明書

---

高中組 生活與應用科學科

第三名

040817

視網膜感光細胞之電腦程式設計分析

學校名稱：臺北市立第一女子高級中學

|               |                     |
|---------------|---------------------|
| 作者：<br>高一 劉康玲 | 指導老師：<br>黃芳蘭<br>陳若潔 |
|---------------|---------------------|

關鍵詞：視網膜 感光細胞 程式設計

# 作品名稱：視網膜感光細胞之電腦程式設計分析

## 壹、摘要

本研究嘗試以自行設計的電腦程式，探討分析眼睛感光細胞的特色。由本研究所設計的圖樣，可以簡易地測量自己是否色盲，還可以比對家中螢幕是否有色差，試著建立模擬色盲患者看見的影像，並撰寫程式應用於盲人導航。

## 貳、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。

## 參、研究目的

本研究嘗試以自行編輯的電腦程式，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。

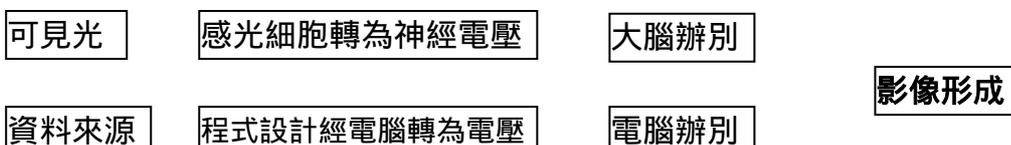
## 肆、研究設備及器材

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

## 伍、研究過程或方法

### 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。



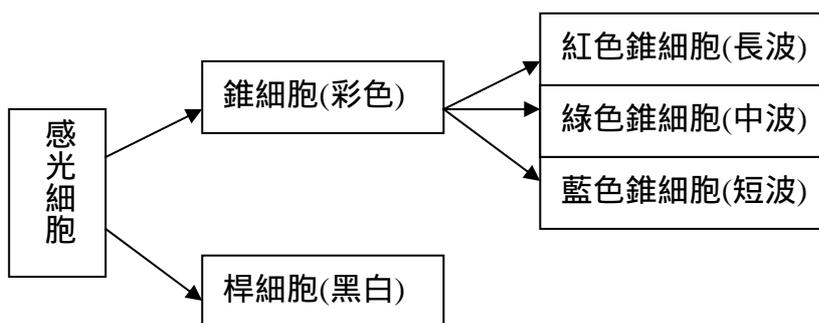
## 二、方法

- (一) 了解視網膜中黑、白、紅、綠、藍等感光細胞的基本結構。
- (二) 撰寫電腦程式，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。分析呈現的項目如下：
  - 1、眼睛色盲及夜盲檢驗。
  - 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）用透明投影片印製，並將印製好的投影片覆蓋於電腦螢幕及印好圖樣的紙張上，相互比對檢視差異
  - 4、用不同畫素數位攝影機拍攝無反光白紙之圖樣（與 3 的圖樣相同），相互比對檢視其差異。
  - 5、撰寫程式，模擬色盲患者所見的影像。
  - 6、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。

## 陸、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（如圖 1）。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（如圖 2、圖 3），連微弱到一百萬億分之一瓦特的光線都能看到。而錐狀細胞則負責對彩色感覺（如圖 2、圖 3）。無論何時，這些感光細胞皆能同時處理 150 萬個訊息。



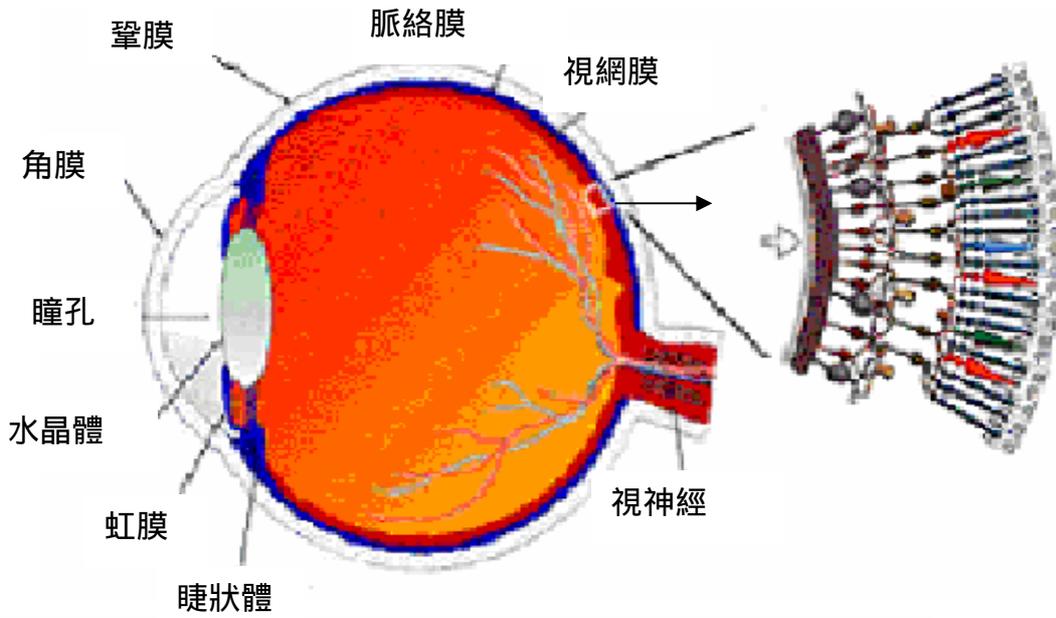


圖 1

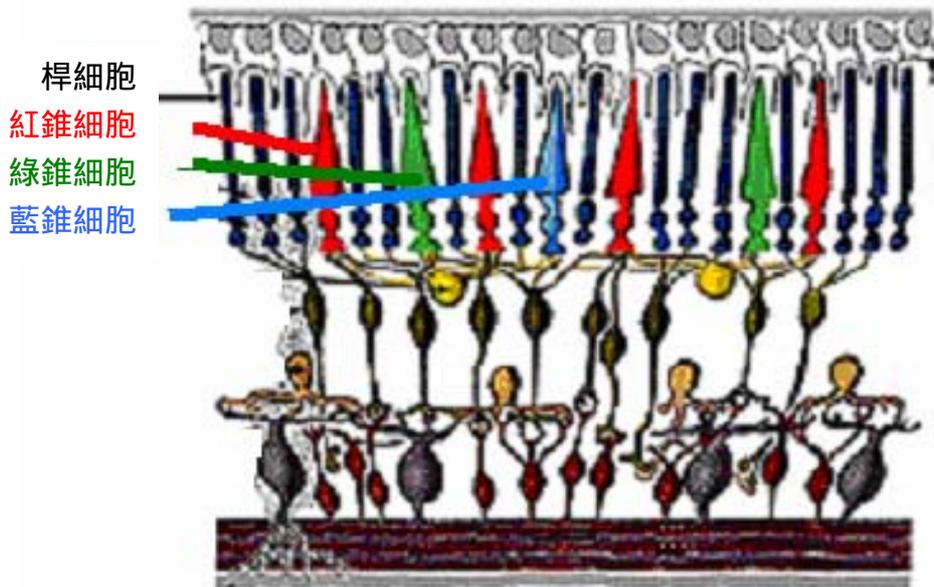


圖 2

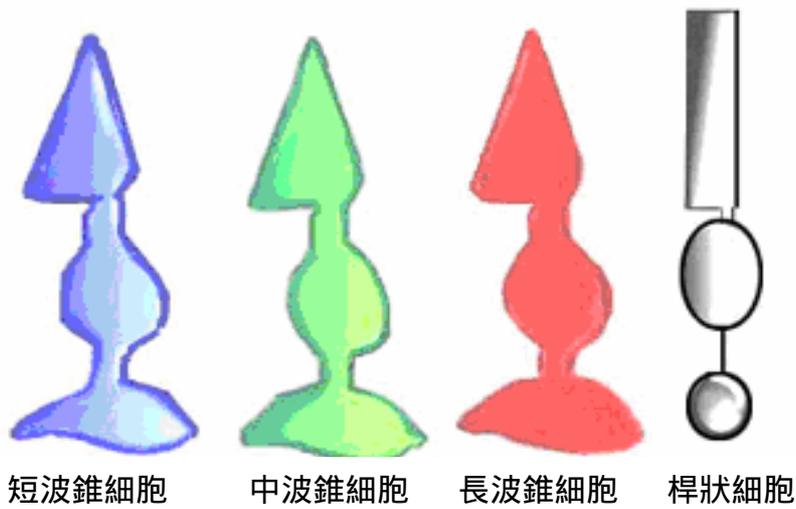


圖 3

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)，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。

$256 * 256 * 256 = 16,777,216$  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一。

(一) 眼睛色盲及夜盲檢驗：

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. 測試桿狀細胞對黑白色敏感性及黑素：(圖 4)

桿狀細胞密度不夠時：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。

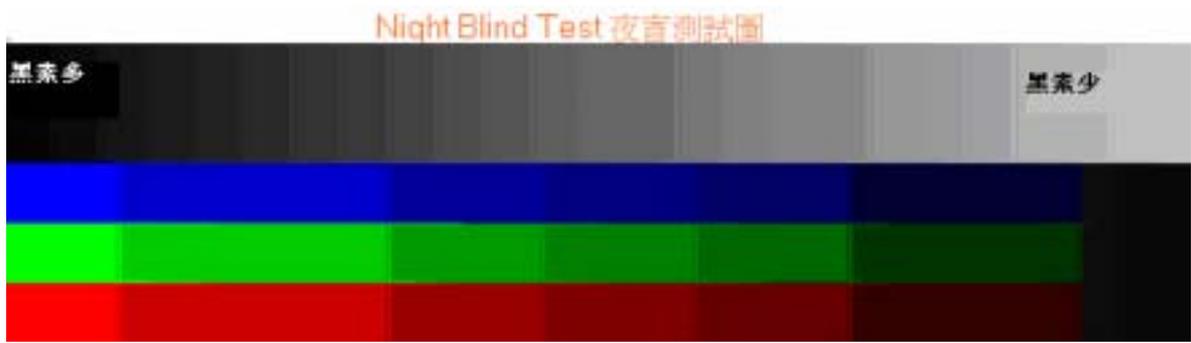


圖 4

2. 色盲測試的原理及方法：

色盲是因為紅藍綠錐狀細胞的多寡所引起的，因此以圖 5 每格顏色疏密程度的不同 (好比紅藍綠錐狀細胞的多寡)，測試是否有色盲，及如果有，是什麼顏色的色盲。

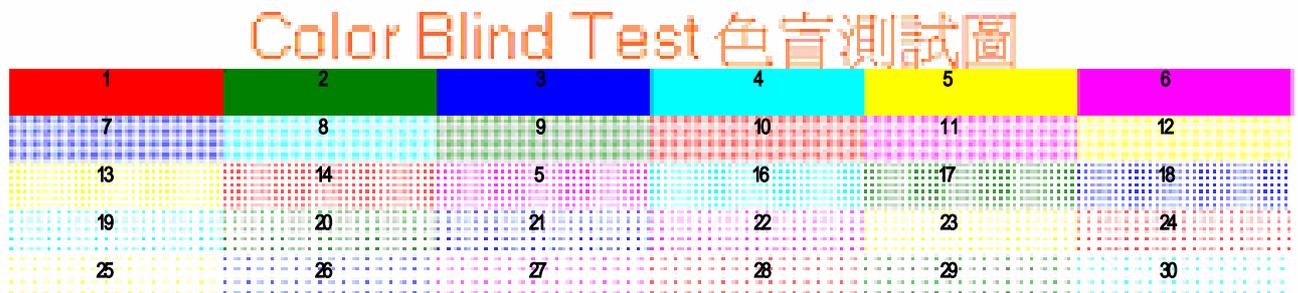


圖 5

若沒辦法找出圖 5 中某色的整欄代號，代表某色錐狀細胞不足，則可進階得知屬於哪一色色盲。例如：如果無法辨識出圖 6 中的 1, 10, 14, 24, 28 代表為紅色色盲。

| 顏色 | 代號代表圖 9 上每格所標示的代號 |
|----|-------------------|
| 紅  | 1, 10, 14, 24, 28 |
| 綠  | 2, 9, 17, 20, 29  |
| 藍  | 3, 7, 18, 21, 26  |
| 藍綠 | 4, 8, 16, 19, 30  |
| 黃  | 5, 12, 13, 23, 25 |

圖 6

(二) 感光細胞在視網膜內分布的情況及檢視方法

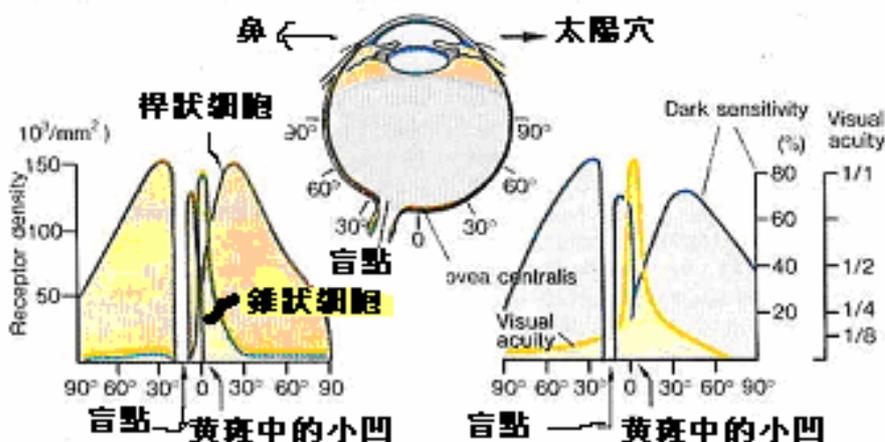


圖 7

由圖 7 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, 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 8 表示本研究中感光細胞在眼睛正面視網膜中的分布密度。

由於錐狀細胞分部的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 9 的彩色細線。這就如同相機解析度不足時無法拍出清晰的圖像。

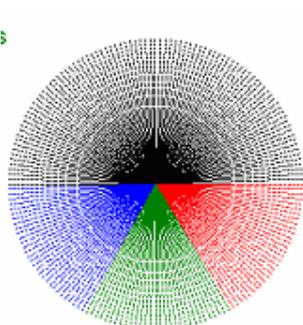


圖 8

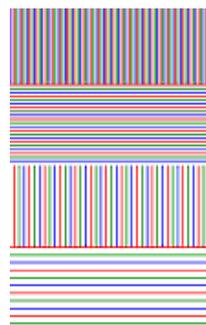
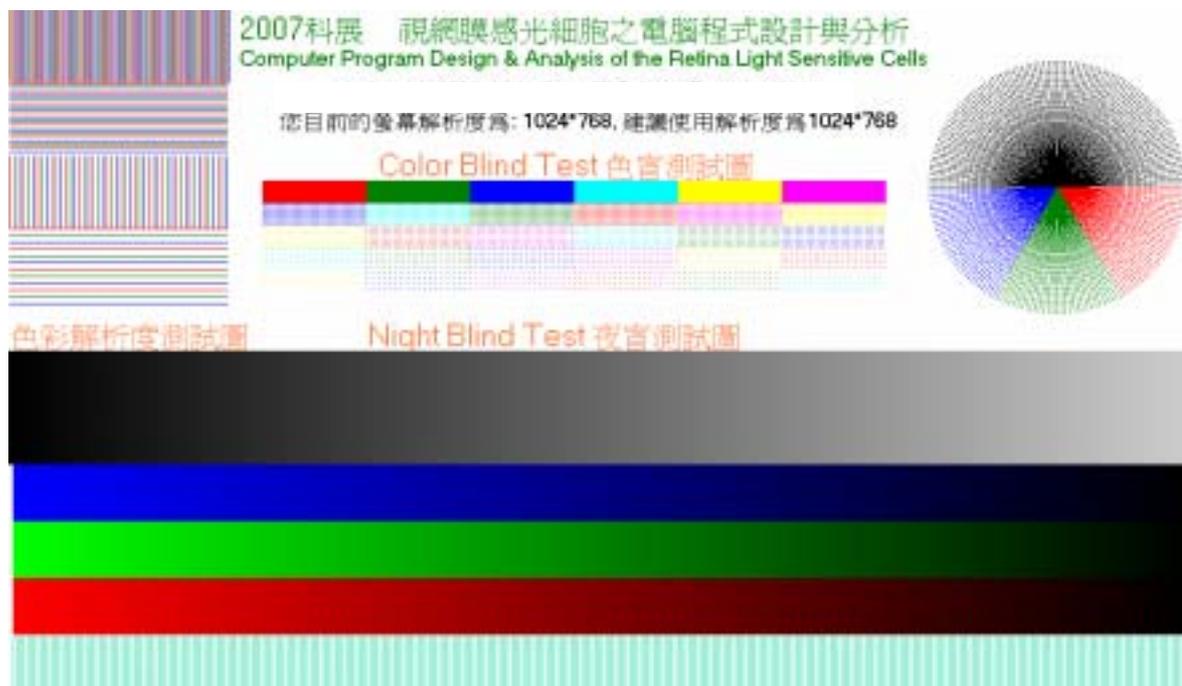


圖 9

(三) 將已設計好之圖樣(圖 10)以雷射印表機(或精密之專業印刷機)用透明投影片印製, 並將印製好的投影片覆蓋於電腦螢幕及印好圖樣的紙張上, 相互比對檢視差異。



- (四) 以數位攝影機先拍攝無反光白紙之圖樣 (圖 11)  
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  
檢視差異。

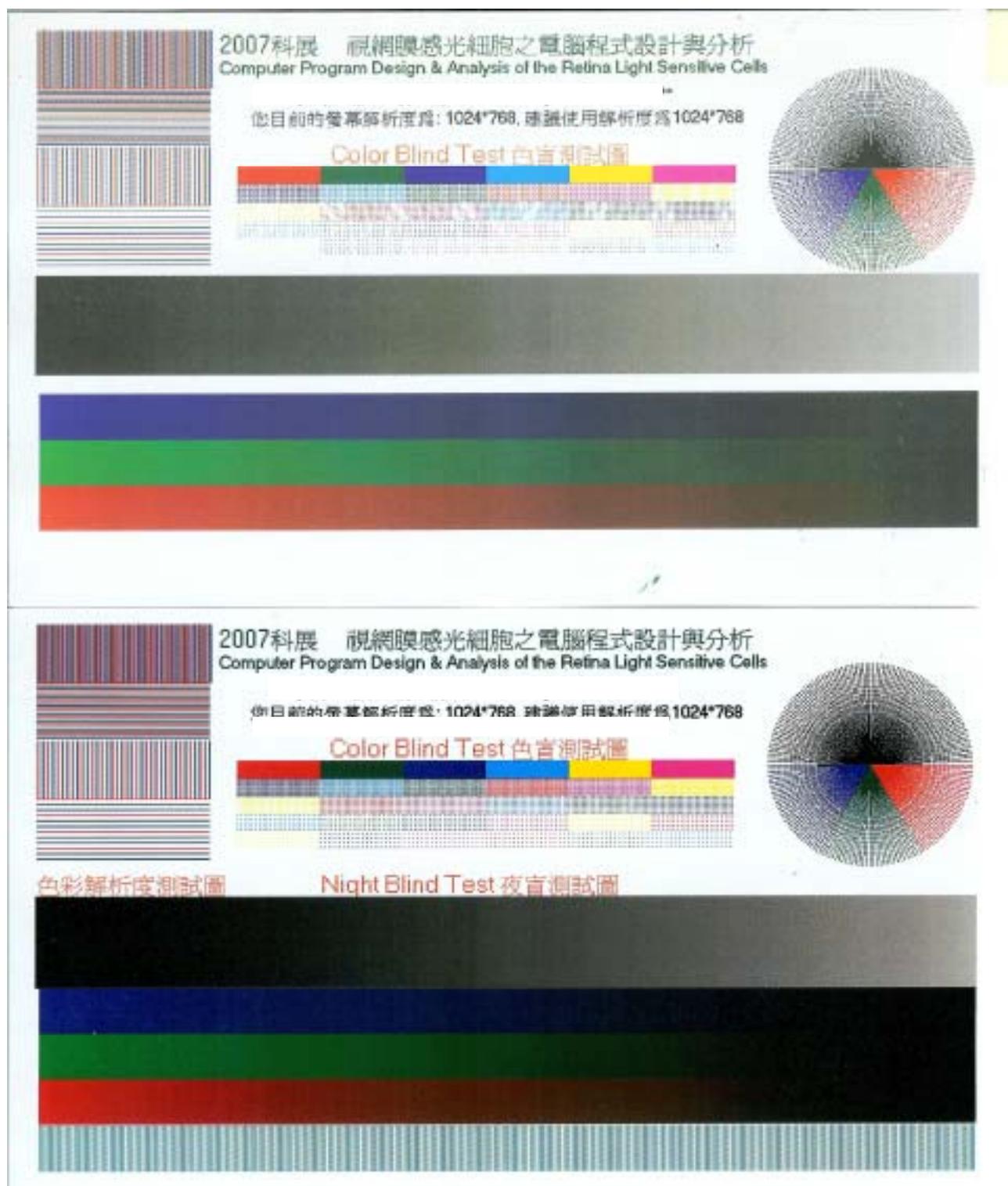
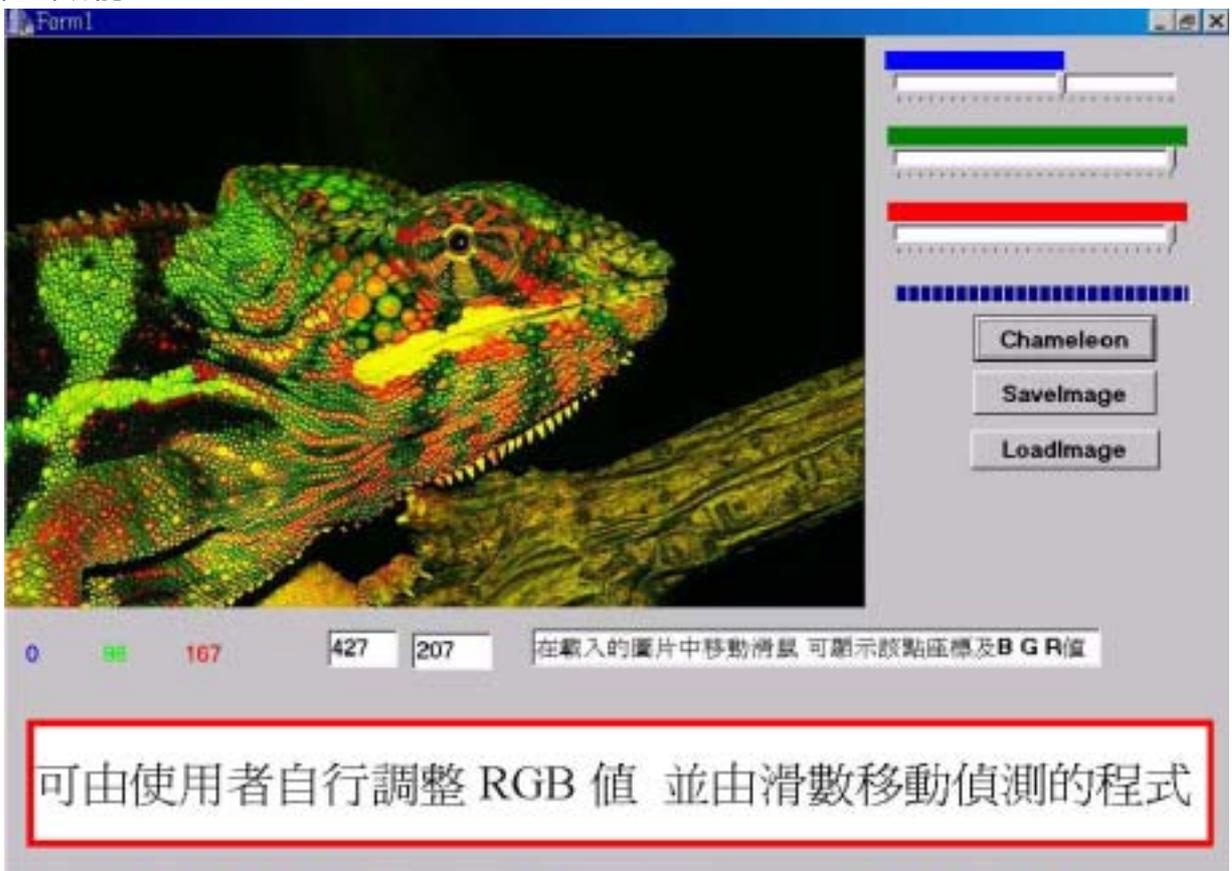


圖 11

(五) 撰寫程式設計，模擬色盲患者所見的影像。

色盲是因為錐狀細胞不足所引起的，當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：



1. 顏色調整拉桿（拉桿值為 0~255 的 1/10）

色盲的人，可先藉由圖五所設計之表格，測出自己哪一種錐狀細胞不足，然後再一邊看著一張圖案範本，一邊看電腦螢幕上同圖案範本的圖示。因已先測出自己哪一種錐狀細胞不足，在此調整該紅、綠、藍拉桿，直至電腦螢幕的圖示經過程式運算後（按下 Chameleon），所顯示出的對受試者來說與圖案範本相同，此即是色盲者看到的影像。

註：若拉桿在一定範圍內，對受試者而言，所看見的圖片相同，則應調到此範圍之最小值。例如完全沒有紅色錐狀細胞的人，調整整條紅色拉桿，對他而言所呈現的圖片都相同，所以，將拉桿調至最小值 0，即符合完全沒有紅色錐狀細胞。

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，此可方便色盲者知道此點（看到的顏色），實際上應是何色。



圖 12 原圖(全彩)

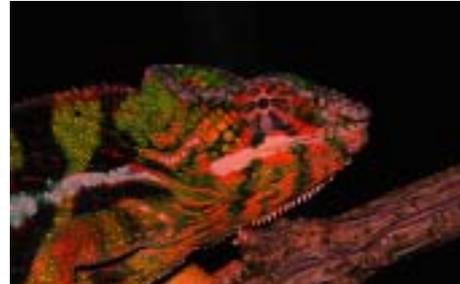


圖 13 模擬少了二分之一的藍色錐狀細胞、綠色錐狀細胞，看到的影像。



圖 14 模擬少了二分之一的藍色錐狀細胞、紅色錐狀細胞，看到的影像。



圖 15 模擬少了二分之一的綠色錐狀細胞、紅色錐狀細胞，看到的影像。



圖 16 模擬少了二分之一的綠色錐狀細胞，看到的影像。



圖 17 模擬少了二分之一的紅色錐狀細胞，看到的影像。

#### (六) 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 18，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 19，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。程式碼詳見附錄三。



圖 18



圖 19

## 柒、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，顏色不太相同有差異(圖 11)；這代表不同眼睛或眼內錐狀、桿狀細胞數目不同的人，所看到的顏色是有差異的。

也就符合圖 4、5 設計預測的顏色疏密程度的不同(好比紅藍綠錐狀細胞的多寡)，若錐狀、桿狀細胞數目不足時，所看見的影像較不清晰。桿狀細胞密度不夠時：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。

又符合圖 7、8 設計預測的感光細胞在眼睛正面視網膜中的分布密度。由於錐狀細胞分部的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 8 的彩色細線。

圖 13 到圖 17 所模擬的結果為完全缺少某一種或二種顏色錐狀細胞的色盲所看到的影像，這與實際色盲患者看到的，應該還有些落差。這裡僅是簡單模擬而已，只有考慮到有沒有什麼顏色的錐狀細胞，未來精確的預估還需考量更多的變因，如錐狀細胞的數量、分布範圍等。

圖 19 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，即可立即發出警戒聲。

## 捌、結論

根據本研究，藉由設計電腦程式，經硬體運算，可以數字精準地顯示出眼睛的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）自行簡易的驗證。因為傳統的色盲測試表，只能大略測試出有色盲的現象，透過本研究所設計的圖表（圖 5），可以更精確的測出患者是屬於何種顏色的色盲，也就是何種錐狀細胞（紅藍綠）的缺失。

根據研究也可將色彩簡單的應用於生活中。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，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（chameleon 變色龍：遇酸性水溶液時會成紅色，遇鹼性水溶液時會成綠色，遇中性水溶液時則不會變色。多軟體動物：當放射肌收縮時，色素細胞脹大，體色變濃；當放射肌舒張時，色素細胞縮小，體色變淡。與色素細胞混雜出現的，還有一種能發出特殊閃光的虹細胞。

以本研究為基礎，亦可發展機器人視覺（machine vision），這在目前精密電子機械製造業上有極為重要廣汎的應用（如盲人導航、無人駕駛、逆向工程、甚至於可更精準的取代目前最 hot 的 Wii 3D 陀螺儀定位，在生物微機電方面（MicroElectronicMechanicSystem，MEMS），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，希望能使盲人恢復視覺或發展可輔助盲人視覺儀器，若能成功將對盲人有很大的貢獻。

## 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. 陳立仁臨床指導醫師。台北馬偕醫院資深眼科主治醫生 ( 專精視網膜眼疾 )。
5. Despopoulos/Silbernagl(2003). Color Atlas of Physiology. Georg Thieme Verlag.
6. Chamelon Image, AquaPets, 45。
7. 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>.
8. University of Utah The Organization of the Retina and Visual system. Retrieved December 12, 2006, from <http://webvision.med.utah.edu/>

### 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)，且長度要延至 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, "2007 科展
    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]=c1Blue;
} }

//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);
    }
}
//-----

```

## 附錄二

### 模擬色盲者看到的影像

```

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;
//no GreenBlue
//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;          } }
```

1. 增加顏色調整拉桿 ( 拉桿值為0~25 . 256的1/10 )

```
Image1->Canvas->Pixels[X][Y]==B*(TrackBar1->Position/25)*256*256+G*(TrackBar2->P
osition/25)*256+R*(TrackBar3->Position/25);
```

2. 增加輸入BMP檔控制鈕

```
OpenPictureDialog1->Execute();
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
```

3. 增加滑鼠在測試畫面移動時顯示RGB及座標值

```
Label3->Caption=Image1->Canvas->Pixels[X][Y]%256;
Label2->Caption=(Image1->Canvas->Pixels[X][Y]%65536)/256;
Label1->Caption=Image1->Canvas->Pixels[X][Y]/65536;
Edit1->Text=IntToStr(X); Edit2->Text=IntToStr(Y);
```

### 附錄三

本研究之生活應用，利用本研究之對顏色判別程式設計成果配合雷射直線行進的原理，設計出可偵測有障礙物及發出警報聲的應用程式。

```
for (X=0;X<Image1->Width;X++)
{for (Y=Image1->Height;Y>0;Y--)
{ R=Image1->Canvas->Pixels[X][Y]%256;
//先分析出該點RGB值
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)))
//判別出為紅色雷射
{ Image1->Canvas->Pixels[X][Y-5]=clWhite;
if((ay-Y)>StrToInt(Edit3->Text)|(Y-ay)>StrToInt(Edit3->Text))
//鄰近兩點y軸差距超過敏感值時
{Image1->Canvas->Pixels[X][Y-5]=clYellow;
sndPlaySound("ding.wav", SND_SYNC); //發出警告聲音
Memo2->Lines->Add(FloatToStr(X)+" " +FloatToStr(Y));          }
Memo1->Lines->Add(FloatToStr(X)+" " +FloatToStr(Y));
ax=X;
ay=Y;
break;    } } } }
```

【評 語】 040817 視網膜感光細胞之電腦程式設計分析

1. 整合各個科目所學，由自身及生活發想，值得鼓勵。
2. 本作品「利用 RGB 合成全彩的影像所用的模型」應進一步研究與  
「RGB 感光細胞→彩色影像視覺」是否相同。
3. 上一點的釐清有助於確認如色盲視覺研究的結果可信程度。
4. 色盲視覺研究應考慮比對色盲者的實際視覺。