

Contents

1	前言	2
2	相关解答	2
2.1	前置知识了解	2
2.1.1	CIE xy 色度图	2
2.1.2	色品坐标	3
2.1.3	判断坐标是否在要求的色域三角形内	3
2.2	计算结果	3
2.3	结论	6
2.3.1	两种设备的比较	6
2.3.2	两种标准的比较	6
2.3.3	其他	6
2.4	程序代码	6
2.4.1	源代码	6
2.4.2	简单解释	10
2.4.3	缺陷	11
2.5	参考资料	11

计算机图形学第一次作业

彭峰 05121214

2015-4-7

§1 前言

这是任选计算机图形学的第一次编程作业，要求给出简单解题思路、计算结果、结论、程序代码。相关的图片部分放于网上，作业放于 github 上。

§2 相关解答

§2.1 前置知识了解

这次作业查阅了一定量资料，由于未知的原因 CNKI 一直登不上（真是见鬼，那么晚都没登上），所以资料主要来源于 Google。下面简单写写。

§2.1.1 CIE xy 色度图

因为人类眼睛有响应不同波长范围的三种类型的颜色传感器，所有可视颜色的完整绘图是三维的。但是颜色的概念可以分为两部分：明度和色度。例如，白色是明亮的颜色，而灰色被认为是不太亮的白色。换句话说，白色和灰色的色度是一样的，而明度不同。CIE XYZ 色彩空间故意设计得 Y 参数是颜色的明度或亮度的测量。颜色的色度接着通过两个导出参数 x 和 y 来指定，它们是所有三个三色刺激值 X、Y 和 Z 的函数所规范化的三个值中的两个：

$$x = \frac{X}{X + Y + Z} \quad (1)$$

$$y = \frac{Y}{X + Y + Z} \quad (2)$$

$$z = \frac{Z}{X + Y + Z} = 1 - x - y \quad (3)$$

导出的色彩空间用 x, y, Y 来指定，它叫做 CIE xyY 色彩空间并在实践中广泛用于指定颜色。 X 和 Z 三色刺激值可以从色度值 x 和 y 与 Y 三色刺激值计算回来：

$$X = \frac{Y}{y}x \quad (4)$$

$$Z = \frac{Y}{y}(1 - x - y) \quad (5)$$

§2.1.2 色品坐标

CIE 三原色光红原色光 $R=700.00\text{nm}$ 绿原色光 $G=546.1\text{nm}$ 蓝原色光 $B=435.8\text{nm}$ 色品坐标

三原色各自在 $R+G+B$ 总量中的相对比例（记为 r, g, b ）CIE RGB 空间可以被用来以常规方式定义色度：色度坐标是 r 和 g ：

$$r = \frac{R}{R + G + B} \quad (6)$$

$$g = \frac{G}{R + G + B} \quad (7)$$

§2.1.3 判断坐标是否在要求的色域三角形内

主要是判断坐标是否在色域三角形内，转为判断一个点是否在指定的三角形内。算法的方法较多，主要有下面几种：

1. 通过计算三个小三角形的面积是否和三角形的面积相等来判断
2. 通过向量叉乘的方法判断
3. 通过向量点乘的方法判断

这里不做过多叙述。

§2.2 计算结果

Listing 1: Answers

```
1 |
2 | YOU SHOULD INPUT COLOR_X , COLOR_Y , COLOR_Z
3 |
4 | THE OUTPUT IS :
```

```

5
6 97.98 100.00 95.75
7 #####Color1#####
8 ##                               Status                               ##
9 ## Device:Galaxy                  1                               ##
10 ## Device:iPad2                   1                               ##
11 ## Standard:AdobeRGB              1                               ##
12 ## Standard:sRGB                   1                               ##
13 ##                               ##
14 ## When Status is 1 , it means This color can be ##
15 ## displayed in Devices/Standard ,if Status is 0 , ##
16 ## it means can not be.                               ##
17 #####
18
19
20
21 if you want to test more Colors' CIE-XYZ ,Please input
22 Color_X Color_Y Color_Z
23 47.35 85.00 12.40
24 #####Color2#####
25 ##                               Status                               ##
26 ## Device:Galaxy                  1                               ##
27 ## Device:iPad2                   1                               ##
28 ## Standard:AdobeRGB              1                               ##
29 ## Standard:sRGB                   0                               ##
30 ##                               ##
31 ## When Status is 1 , it means This color can be ##
32 ## displayed in Devices/Standard ,if Status is 0 , ##
33 ## it means can not be.                               ##
34 #####
35
36
37
38 if you want to test more Colors' CIE-XYZ ,Please input
39 Color_X Color_Y Color_Z

```

```

40 131.72    211.00    344.70
41 #####Color3#####
42 ##                               Status                               ##
43 ## Device:Galaxy                  1                               ##
44 ## Device:iPad2                   0                               ##
45 ## Standard:AdobeRGB              1                               ##
46 ## Standard:sRGB                  0                               ##
47 ##                               ##
48 ## When Status is 1 , it means This color can be ##
49 ## displayed in Devices/Standard ,if Status is 0 , ##
50 ## it means can not be. ##
51 #####
52
53
54
55 if you want to test more Colors' CIE-XYZ ,Please input
56 Color_X Color_Y Color_Z
57 19.18    72.00    11.68
58 #####Color4#####
59 ##                               Status                               ##
60 ## Device:Galaxy                  1                               ##
61 ## Device:iPad2                   0                               ##
62 ## Standard:AdobeRGB              0                               ##
63 ## Standard:sRGB                  0                               ##
64 ##                               ##
65 ## When Status is 1 , it means This color can be ##
66 ## displayed in Devices/Standard ,if Status is 0 , ##
67 ## it means can not be. ##
68 #####
69
70
71
72 if you want to test more Colors' CIE-XYZ ,Please input
73 Color_X Color_Y Color_Z

```

§2.3 结论

§2.3.1 两种设备的比较

- 这里只对色域部分进行简单比较，多点测试时表明 Galaxy 的可能更广
- Galaxy 四点测试全部通过，iPad 通过两点。

§2.3.2 两种标准的比较

- 已知 AdobeRGB 的色域比 sRGB 的色域更广
- AdobeRGB 通过三点，sRGB 通过两点

§2.3.3 其他

- 本次由于没有使用 MATLAB 作图，所以无法判断色域包含情况和色域位置。
- 无法详细比较色域的重叠情况，无法详细输出设备覆盖 AdobeRGB 和 sRGB 的范围。
- 好像 Galaxy 屏幕比较牛逼，当时是 AMOLED 屏？接着 iPad2 还在 IPS 那里混？

§2.4 程序代码

§2.4.1 源代码

Listing 2: Sources

```
1 |
2 | #include <stdio.h>
3 | #include <stdlib.h>
4 | #include <math.h>
5 |
6 | #define ABS_FLOAT_0 0.00001
7 |
8 |
9 | int main()
10 | {
```

```

11     typedef struct point{
12         float x;
13         float y;
14     }point;
15     typedef struct triangle{
16         point A;
17         point B;
18         point C;
19     }triangle;
20
21     float Side_length (point M , point N ){
22         float length ;
23         length = fabs( sqrt((M.x - N.x)*(M.x - N.x) + (M.y - N.y)*(
24             M.y - N.y)) );
25         return length ;
26     }
27
28     int Is_triangle(triangle one){ //判断
29
30         float AB ,AC ,BC ;
31         AB = Side_length(one.A , one.B );
32         AC = Side_length(one.A , one.C );
33         BC = Side_length(one.B , one.C );
34
35         if ( ( (AB*AC*BC)> 0 )&&( (AB+BC) > AC )&&( (AB+AC) > BC )
36             &&( (AC+BC) > AB ) )
37             return 1;
38         else
39             return 0;
40     }
41
42     float Get_Triangle_Square (point A , point B ,point C ){
43         float AB ,AC ,BC ;
44         float square , semi_perimeter ;
45         AB = Side_length(A , B );

```

```

44     AC = Side_length(A , C );
45     BC = Side_length(B , C );
46     semi_perimeter =( AB + AC + BC )/2 ;
47     square = sqrt(semi_perimeter * (semi_perimeter - AB ) * (
48         semi_perimeter - BC ) * (semi_perimeter - AC ) );
49     return square;
50 }
51
52 int Is_display(point CIE_point , point A , point B ,point C){
53     float SABC , SADB , SADC , SBDC ;
54     float SumSquare ;
55     SABC = Get_Triangle_Square(A ,B ,C);
56     SADB = Get_Triangle_Square(A , CIE_point ,B);
57     SADC = Get_Triangle_Square(A , CIE_point , C);
58     SBDC = Get_Triangle_Square(B ,CIE_point , C );
59     SumSquare = SADC + SADB + SBDC ;
60     if((-ABS_FLOAT_0 < ( SABC - SumSquare )) && ((SABC -
61         SumSquare) < ABS_FLOAT_0 )){
62         return 1;
63     }
64     else{
65         return 0 ;
66     }
67 }
68
69 float color_x , color_y , color_z ;
70 float CIE_x , CIE_y;
71 point CIE_point;
72 triangle Galaxy , iPad , AdobeRGB , sRGB ;
73
74 int status_G = -1 , status_i = -1 , status_A = -1 , status_s =
75     -1 ;
76
77 int count = 0;
78
79 (Galaxy.A).x = 0.6627 ; (Galaxy.A).y = 0.3365 ;
80 (Galaxy.B).x = 0.1750 ; (Galaxy.B).y = 0.7315 ;

```



```

76      (Galaxy.C).x = 0.1440 ; (Galaxy.C).y = 0.0431 ;
77      (iPad.A).x   = 0.6476 ; (iPad.A).y   = 0.3293 ;
78      (iPad.B).x   = 0.3062 ; (iPad.B).y   = 0.6109 ;
79      (iPad.C).x   = 0.1525 ; (iPad.C).y   = 0.0454 ;
80      (AdobeRGB.A).x = 0.64; (AdobeRGB.A).y = 0.33 ;
81      (AdobeRGB.B).x = 0.21; (AdobeRGB.B).y = 0.71 ;
82      (AdobeRGB.C).x = 0.15; (AdobeRGB.C).y = 0.16 ;
83      (sRGB.A).x     = 0.64; (sRGB.A).y     = 0.33 ;
84      (sRGB.B).x     = 0.30; (sRGB.B).y     = 0.60 ;
85      (sRGB.C).x     = 0.15; (sRGB.C).y     = 0.06 ;
86
87
88
89      while (scanf(" %f %f %f " , &color_x,&color_y,&color_z) != EOF
90      ){
91
92          CIE_x = color_x/(color_x + color_y + color_z);
93          CIE_y = color_y/(color_x + color_y + color_z);
94
95          CIE_point.y = CIE_y;
96
97          status_G = -1 , status_i = -1 , status_A = -1 , status_s =
98              -1 ;
99          status_G = Is_display (CIE_point , Galaxy.A , Galaxy.B ,
100              Galaxy.C );
101          status_i = Is_display (CIE_point , iPad.A , iPad.B , iPad.C
102              );
103          status_A = Is_display (CIE_point , AdobeRGB.A , AdobeRGB.B ,
104              AdobeRGB.C );
105          status_s = Is_display (CIE_point , sRGB.A , sRGB.B , sRGB.C
106              );
107
108          count++;
109          printf("#####Color%d
110              #####\n",count );

```

```

104         printf("##                               Status
           ##\n");
105         printf("##   Device:Galaxy                %d
           ##\n",status_G );
106         printf("##   Device:iPad2                %d
           ##\n",status_i );
107         printf("##   Standard:AdobeRGB          %d
           ##\n",status_A );
108         printf("##   Standard:sRGB              %d
           ##\n",status_s );
109         printf("##
           ##\n");
110         printf("## When Status is 1  , it means This color can be
           ##\n");
111         printf("## displayed in Devices/Standard ,if Status is 0 ,
           ##\n");
112         printf("## it means can not be.
           ##\n");
113         printf("
           #####\n")
           ;
114         printf("\n");
115         printf("\n");
116         printf("\n");
117         printf("if you want to test more Colors' CIE-XYZ ,Please
           input \n");
118         printf("Color_X   Color_Y Color_Z   \n");
119     }
120     return 0;
121 }

```

§2.4.2 简单解释

判断三角形存在函数

`Is_triangle()`，不过这次只是写上去，并没有使用。求三角形面积函数 `Get_Triangle_Square()`。判断要求的点是否在三角形内的函数 `Is_display()` 使用三角形面积判断的方式。两种设备和两个标准的相关值被固化在程序内部，输出没有更人性化，直接将返回值返回了。

§2.4.3 缺陷

1. 扩展性不好，只是习惯性的可以输入多个数值。没有可以输入多个色域的函数。
2. 输出没有更人性化，没有提示输入，输出只使用了 0 和 1 表示。
3. 偷懒没有使用 MATLAB，额，其实已经好久没有用过了，还占地方。

§2.5 参考资料

二维平面上判断点是否在三角形内

<http://www.cnblogs.com/TenosDoIt/p/4024413.html>

CIE 的 Presentation

https://www.academia.edu/8313696/_CIE 1

WiKi <http://zh.wikipedia.org/wiki/CIE1931>

题目的图片

<http://fengidea.qiniudn.com/graphics-20150407-002.jpg>