计算机视觉 Ex1 实验文档

梁俊华 数据科学与计算机学院 16340129

一、实验要求

Ex1：图像读取和显示以及像素操作

1. 读入 1.bmp 文件，并用 CImg.display()显示.

2. 把 1.bmp 文件的白色区域编程红色，黑色区域变成绿色.

3. 在图上绘制一个圆形区域，圆心坐标(50,50)，半径为 30，填充颜色为蓝色.

4. 在图上绘制一个图形区域，圆心坐标(50,50)，半径为 3，填充颜色为黄色.

5. 在图上绘制一条长为 100 的直线段，起点坐标为(0,0)，方向角为 35 度，直线的颜色为蓝色.

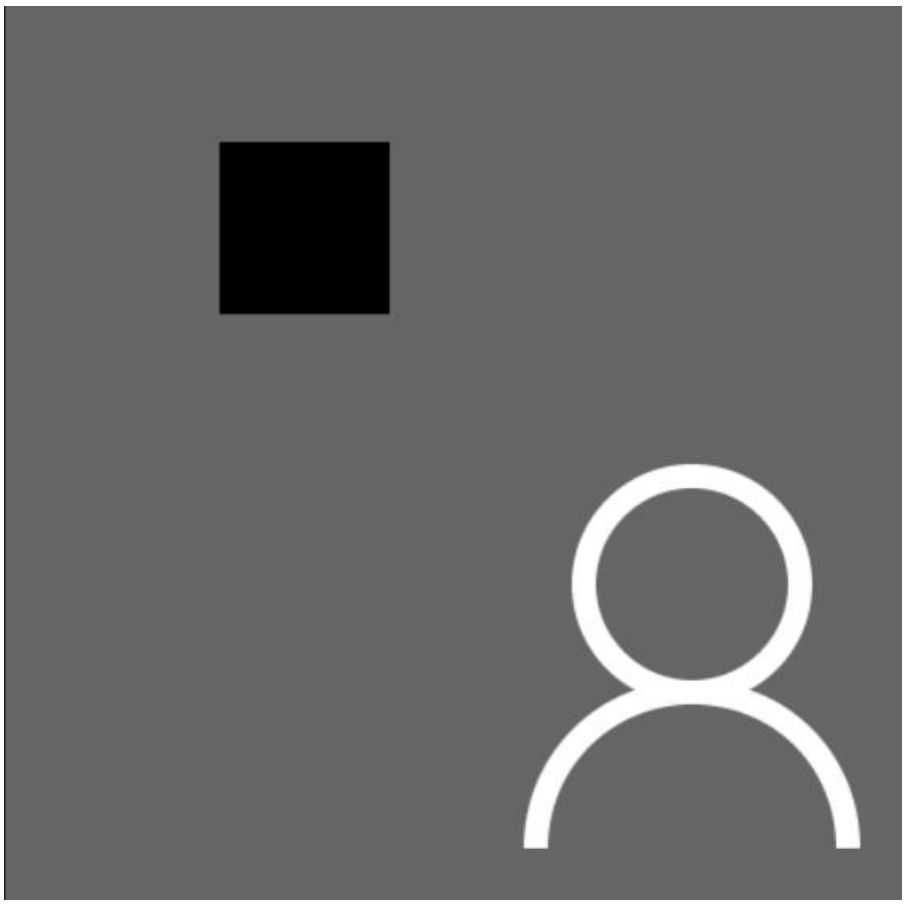
6. 把上面的操作结果保存为 2.bmp

二、测试环境

实验的测试环境在 windows10 操作系统，用 DevC++进行实验。使用 DevC++进行编译时，需要在工具->编译选项中添加相关的条件语句，才能够正常使用 CImg 库，如下图所示：

三、测试数据

实验 Ex1 的测试数据为老师实验中所给的图 1.bmp，如下所示：



四、测试说明

在本次实验中，实验的六个部分被分别封装在 MyImage 类的六个方法里面，其中第三到第五步的方法要求分别使用 CImg 库的绘图方法和不使用 CImg 库的绘图方法，因此这三步均分别设置了两种不同的方法，分别为未使用 CImg 库绘图方法的 processThree()、processFour()、processFive()，和使用了 CImg 库的绘图方法 draw_circle() 和 draw_line() 的方法 processThreeCImg()、processFourCImg()、processFiveCImg().

在测试的过程中，如果需要测试使用了 CImg 库绘图方法的函数，

则在执行后输入 0，图像会保存至 3.bmp；如果需要测试未使用 CImg 库方法的函数，则在执行后输入 1，图像会保存至 2.bmp。其中两张图片均是执行完所有实验要求后的结果，因此都会显示出 1 个蓝黄的同心圆和一条穿过圆的直线。

如果希望测试某一个单独的方法，如 processTwo()，则可以注释掉其他的几个方法，单独测试，若要保存出相应的结果图像来观察，应注意不要把 processSix()注释掉。

实验代码中相关类的说明，在代码中均有注释，在此处不再说明。

五、实验原理

步骤一只需要直接调用 CImg 库的方法 display()即可.

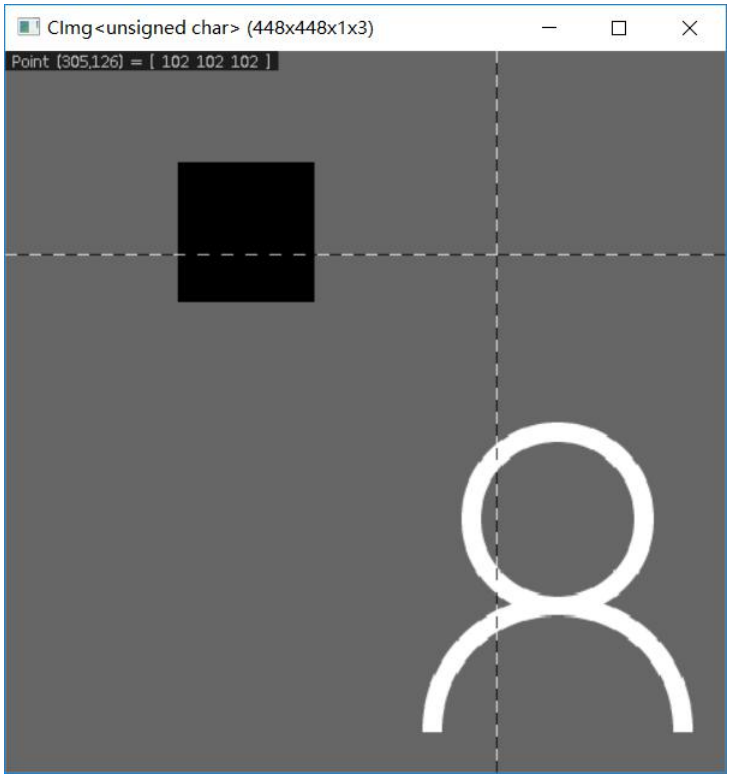
步骤二的实验原理，是检测图片白色区域和黑色区域的 RGB 值分别为（255，255，255）和（0，0，0），然后通过对其中像素点的 RGB 值修改为（255，0，0）和（0，255，0）即可编程红色和绿色.

步骤三和步骤四的实验原理相同，都是对 bmp 图像的每一个像素点，根据给定的圆心 (a,b) 和半径 r，确定圆的方程为 $(x-a)^2+(y-b)^2=r^2$，然后计算某一像素点 (x_0,y_0) 到圆心 (a,b) 的距离 $d=\sqrt{(x_0-a)^2+(y_0-b)^2}$，当 $d\le r$ 时修改相应的 RGB 值即可画出圆.

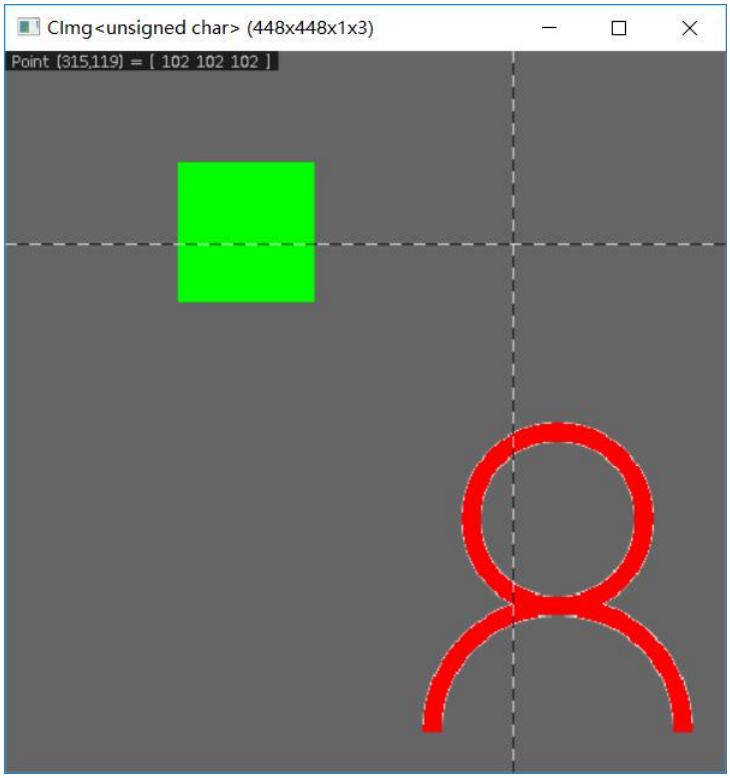
步骤五的实验原理是根据直线的斜截式方程 $y=y_0+k(x-x_0)$，对于每一个像素点的横坐标 x_k，若满足 $y_k=y_0+k(x_k-x_0)$，则修改相应的 RGB 值，即可画出直线.

六、实验结果

步骤一的实验结果如下：

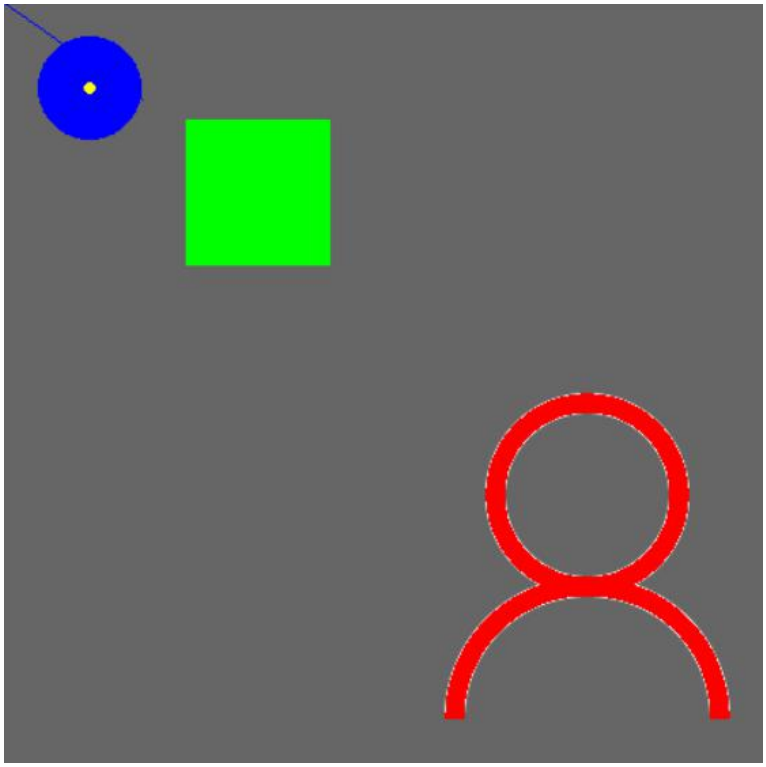


步骤二的实验结果如下：

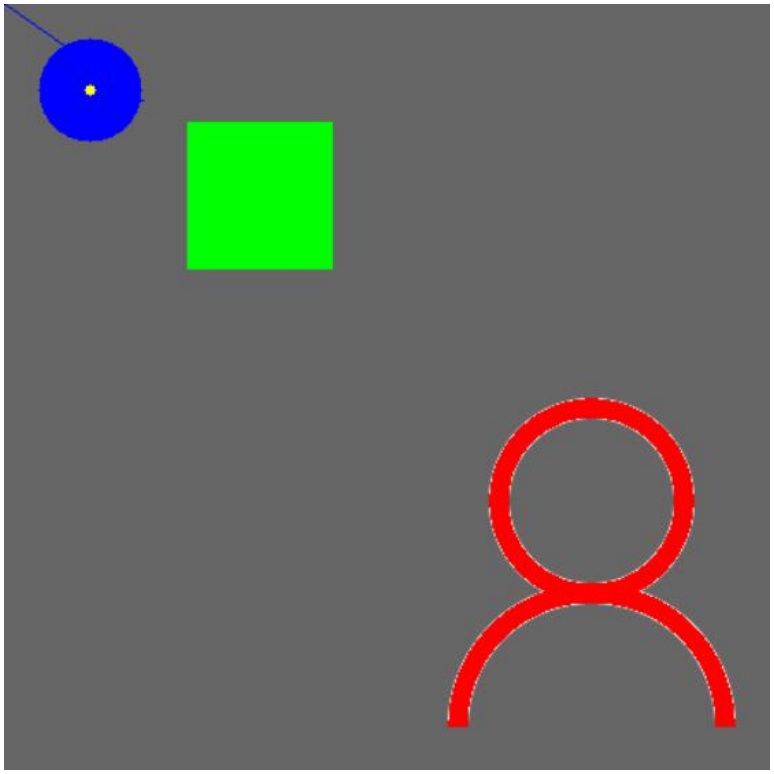


步骤三、四、五的实验结果如下所示：

①使用了 CImg 绘图方法的实验结果：



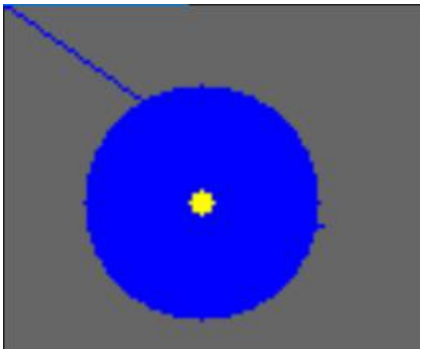
②未使用 CImg 绘图方法的实验结果如下所示：



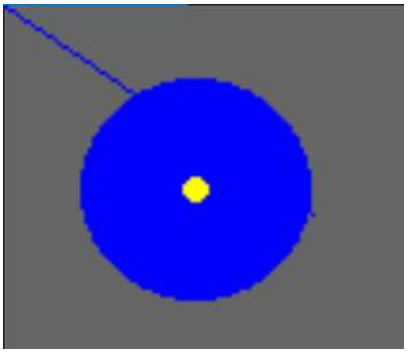
七、实验分析

从上述实验可以看出，步骤一成功显示出和原图相同的图像，而步骤二相应的区域也变成了要求的颜色，但是步骤二中红色和灰色的边缘似乎还有一些白色的存在，可能是逐像素处理的过程中没有处理到这些像素，从而使得这些颜色得以残留。

对于步骤三到步骤五的过程，可以看出比较明显的区别。其中步骤四画出的圆形中，使用了 CImg 绘图方法的结果显然要优于未使用 CImg 库的方法，圆看上去更加的光滑，而未使用的部分看上去则别叫粗糙，显得不是很圆而是一个带棱角的星。但是像步骤三当画的圆稍微大一点的时候，这种差异就显得比较小了，放大对比图如下：



未使用 CImg 绘图方法



使用了 CImg 绘图方法

步骤五画直线的部分，明显觉得直线并不是很直，这有可能是算法的关系，在我自己使用直线方程直接画图的算法中，会因为数据类型之间的一些舍入关系，而导致某些像素点有所差异，所以使得直线看起来比较粗糙，但是用了 CImg 库的画直线的方法，感觉并没有多大的改善。

八、实验思考

Question：为什么第四步绘制的圆形区域形状效果不好？

Answer：我觉得可能是绘制的图形比较小，导致处理的过程中处理的像素点也比较少，所以看起来像素方块的性质就能够凸显出来，因此看上去圆没有那么光滑.

九、实验心得

本实验是计算机视觉第一次作业，应该算是接触到 C++处理图像的一些基本方法和 CImg 库的使用，也对像素和算法有了一些更加深入的理解，初步接触到了在实际处理过程中的一些应用。