



# 《计算机视觉》

## (作业一实验报告)

学 院 名 称 : 数据科学与计算机学院

专业 (班级) : 17 级软工 5 班

学 生 姓 名 : 张淇

学 号 : 17343153

时 间 : 2019 年 9 月 5 日

## 成绩：

# 作业一：图像读取和显示以及像素操作

## 一. 实验目的

1. 将老师课上教的内容付诸于实践，达到学以致用效果。
2. 掌握Cimg库的基本像素操作，为之后的作业打基础。

## 二. 实验内容

对于给定的1.bmp，完成以下操作：

1. 读入 1.bmp 文件，并用 CImg.display() 显示。
2. 把 1.bmp 文件的白色区域变成红色，黑色区域变成绿色。
3. 在图上绘制一个等边三角形区域，其中心坐标(50,50)，边长为 40，填充颜色为蓝色。
4. 在图上绘制一个圆形区域，圆心坐标(50,50)，半径为 15，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 135 度，直线的颜色为绿色。
6. 把上面的操作结果保存为 2.bmp。

需要满足以下要求：

1. 对于上面的第三、四、五步，先不用 CImg 的函数调用，绘制出相应的图形。然后再调用 CImg 相关函数绘制出相应的图形。并在实验文档中对比两者的差异。
2. 把上述功能代码写成类的形式，把上面的第三、四、五步操作封装为该类的操作。

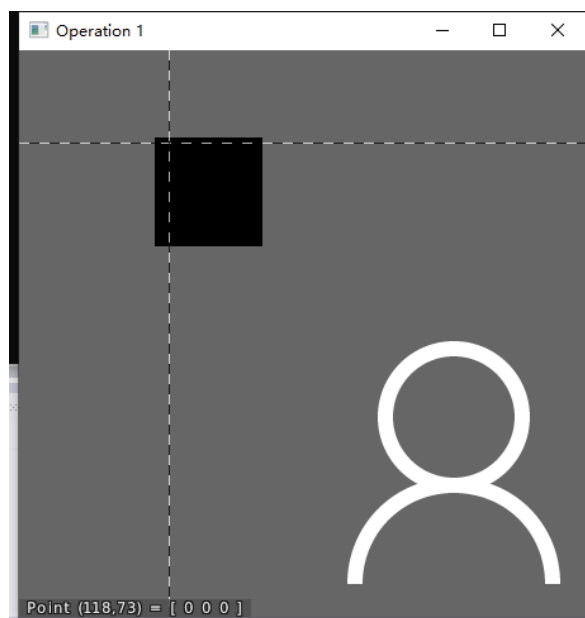
## 三. 实验器材

Win10、VS2017、Cimg.h

## 四. 实验过程与结果

1. 读入 1.bmp 文件，并用 `CImg.display()` 显示。

实验结果如下：

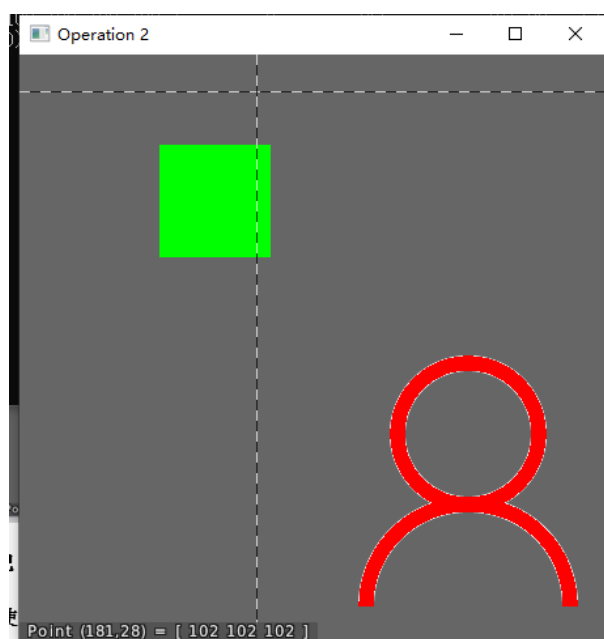


2. 把 1.bmp 文件的白色区域变成红色，黑色区域变成绿色。

使用CImg函数`cimg_forXY()`对图的每一个像素进行遍历、判断即可。

Project中对应函数为`Ans::ans2()`

实验结果如下：

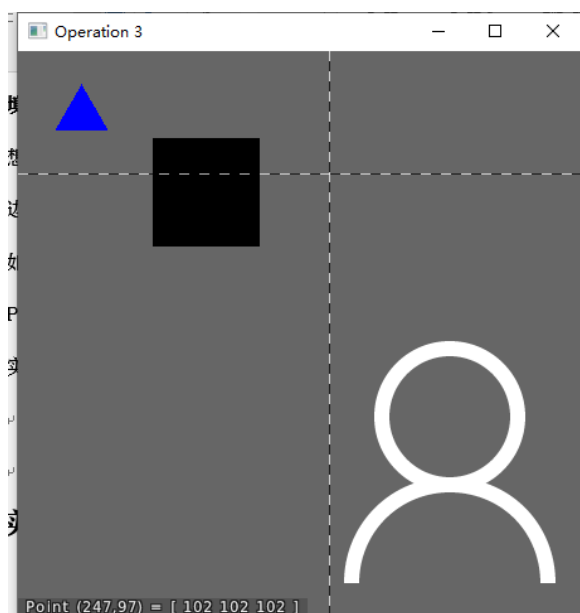


3. 在图上绘制一个等边三角形区域，其中心坐标(50,50)，边长为 40，填充颜色为蓝色。

想要绘制出一个三角形，那么就要确定其边界。所以我先将三角形的三边的方程求出然后再对“包含三角形最小矩形”中对像素进行注意判断，如果该像素是在三边之内的，那么就将其染成蓝色。

Project中对应函数为Ans::ans3()

实验结果如下：

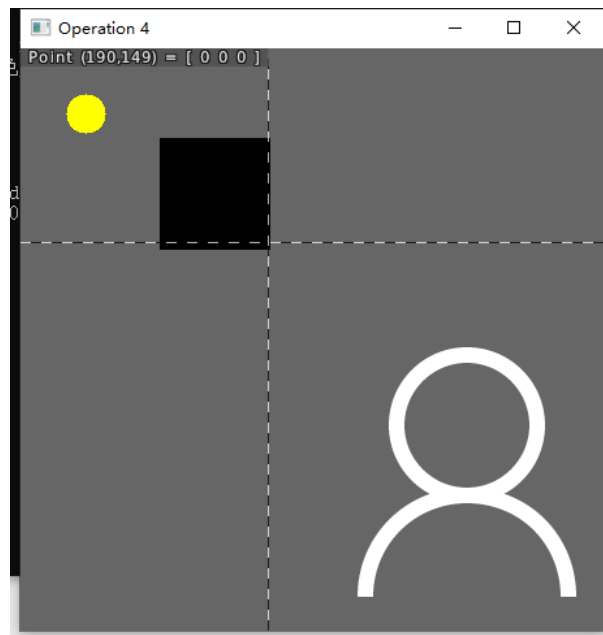


4. 在图上绘制一个圆形区域，圆心坐标(50,50)，半径为 15，填充颜色为黄色。

使用圆的方程 $(x-50)^2+(y-50)^2 \leq 15^2$ ，将坐标x, y符合这个条件的像素染成黄色即可。

Project中对应函数为Ans::ans4()

实验结果如下：

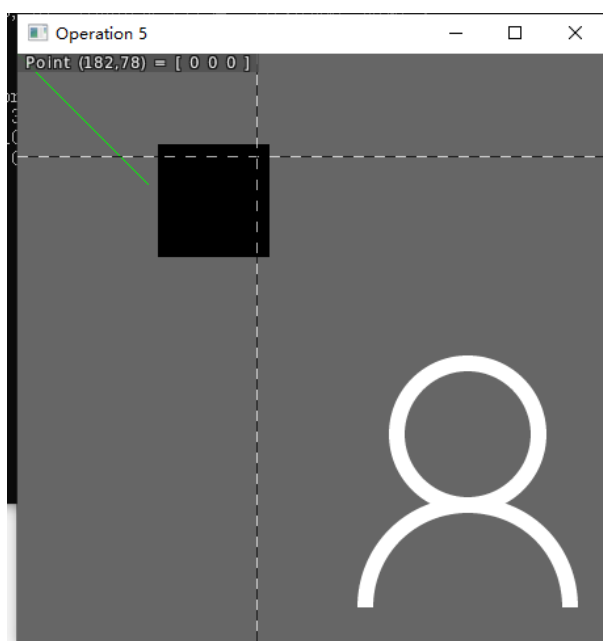


5. 在图上绘制一条长为 100 的直线段， 起点坐标为(0, 0)， 方向角为 135 度， 直线的颜色为绿色。

遍历整张图，将坐标符合( $x \geq 0 \ \&\& \ x < 100 \ \&\& \ y == x$ )的像素染成绿色即可。

Project中对应函数为 `Ans::ans5()`

实验结果如下：

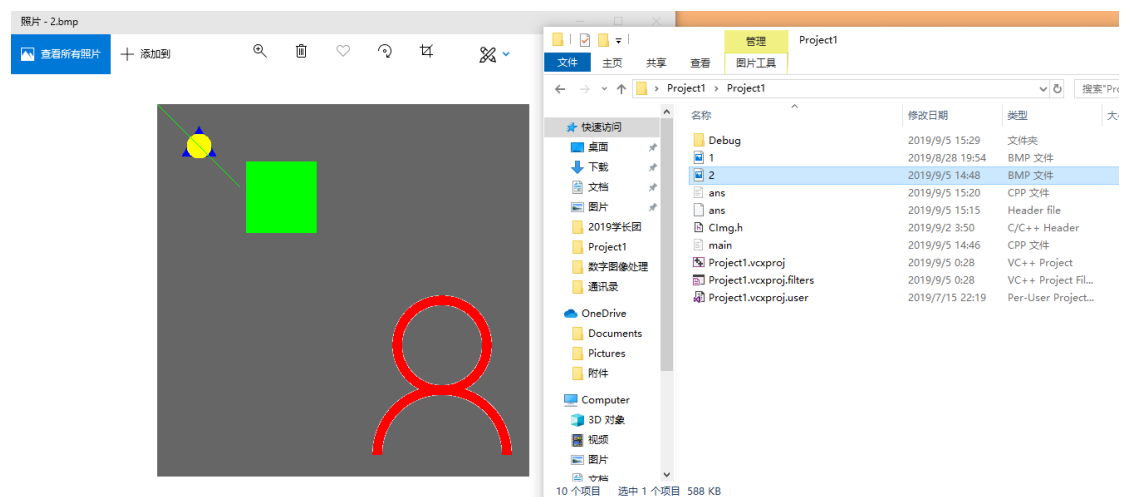


## 6. 把上面的操作结果保存为 2.bmp。

使用CImg的函数save（）即可将上述操作后的图片保存为2.bmp。

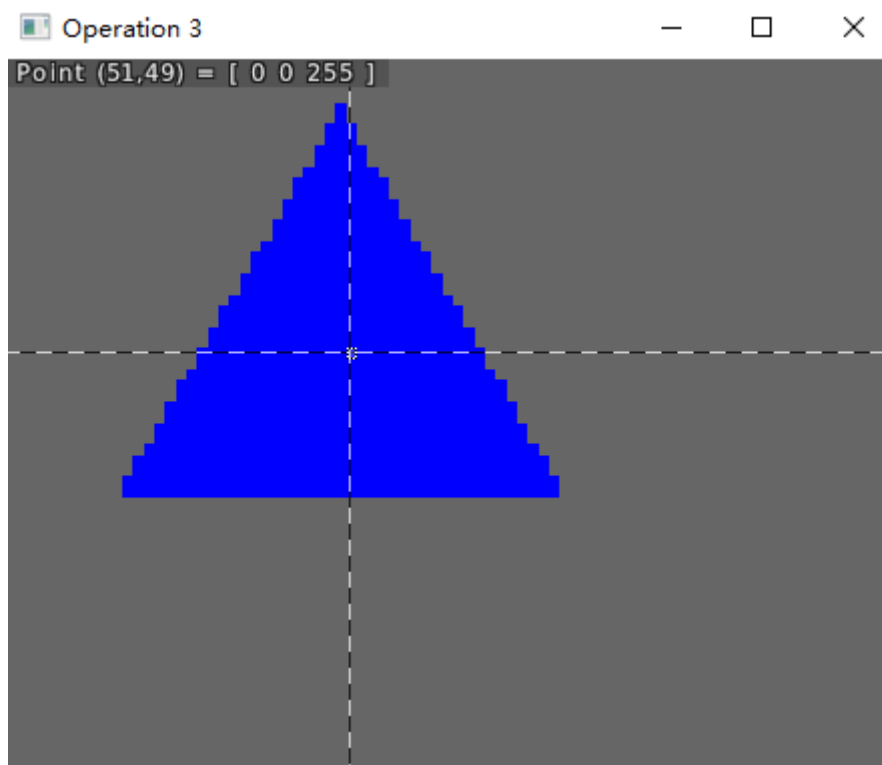
Project中对应函数为Ans::ans6()

实验结果如下：

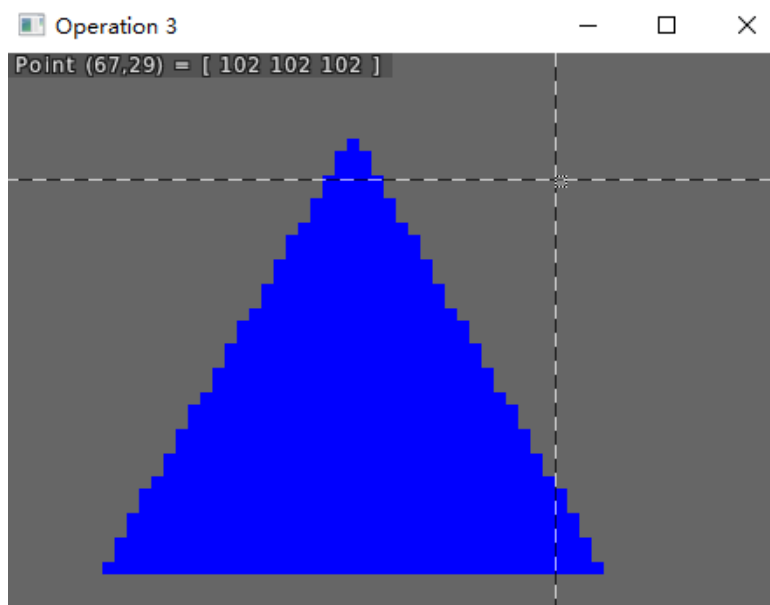


## 【Questions】

对于问题3，这个是根据我自己的操作写出来的图案放大后的版本：

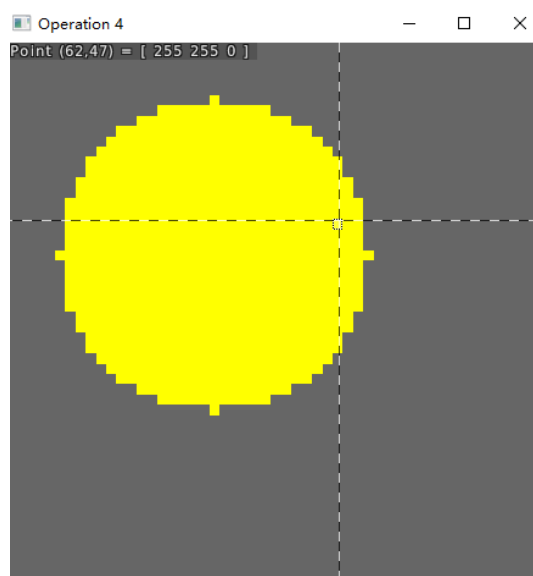


这个是使用CImg的`draw_triangle()`函数得到的放大后的图案：

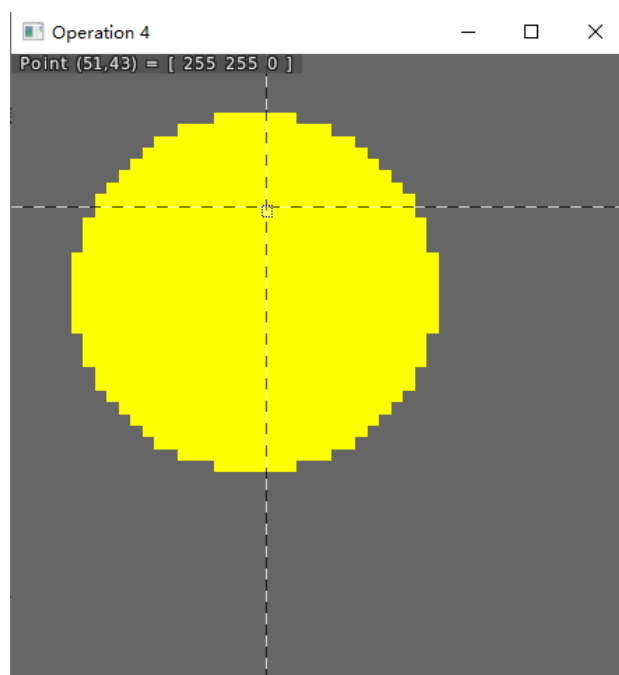


看上去的话其实并没有多大区别，然后我去看了一下CImg这个函数的源码，发现其基本思路也是从三个点确定三条线（边界），最后填充里面的内容。和我的做法大致相同，所以得到的结果也是差别不是很大。

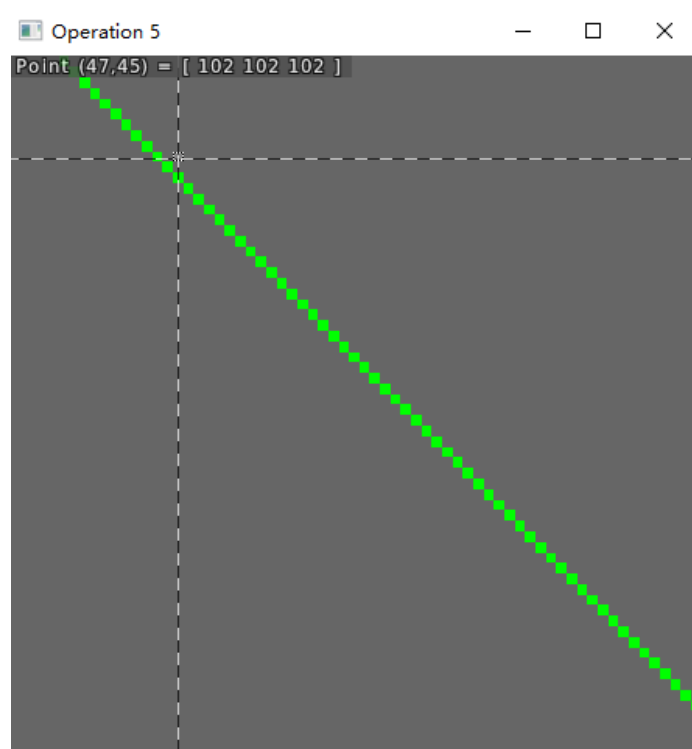
对于问题4，我得到的结果的放大版是这样的（上下左右“多出来一点”是因为我在判断的方程中取了等号）：



反观CImg使用函数`draw_circle()`得到的结果，比我的更像圆一点，原因是他的第一行和第二行的“格子数量”差距较小，所以整体看起来比较好一些。



对于问题5，两种方法得到的结果放大以后都是这样的，所以也没什么可以比较。





## 五. 实验心得

本次实验难度不大，内容不多，能够比较轻松地完成。

不过这很大原因是Cimg比较“简单”，只有一个六万多行的头文件，而且不用配环境，这点相比opencv来说真的是对新人很友好了。

不过我在网上查找相关函数及其用法的时候，发现内容相关的网站、教程很少，所以感觉这个库应该还是比较小众的吧。不过我感觉这个可能会带来一些问题，毕竟没有办法去看完六万多行的代码，就算看完也不能把全部函数都记住，所以之后可能会出现“重复造轮子”的问题…