

实验环境: Ubuntu 16.04 LTS

可执行文件: image.out

代码文件: image.cpp

一、实验部分

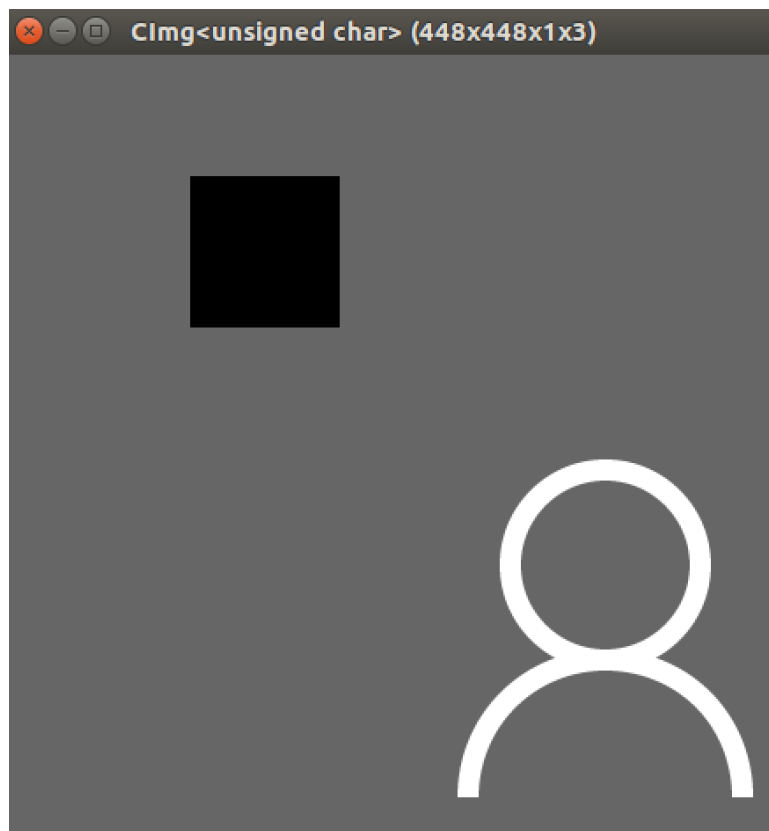
1. 读取 bmp 文件并使用 CImg.display()显示

代码如下:

```
CImg<unsigned char> image;  
//load and display the origin bmp file  
image.load_bmp("1.bmp");  
image.display();
```

通过调用 load_bmp 来 load 函数，并通过使用 display()来呈现函数。

实验结果:



该步骤成功

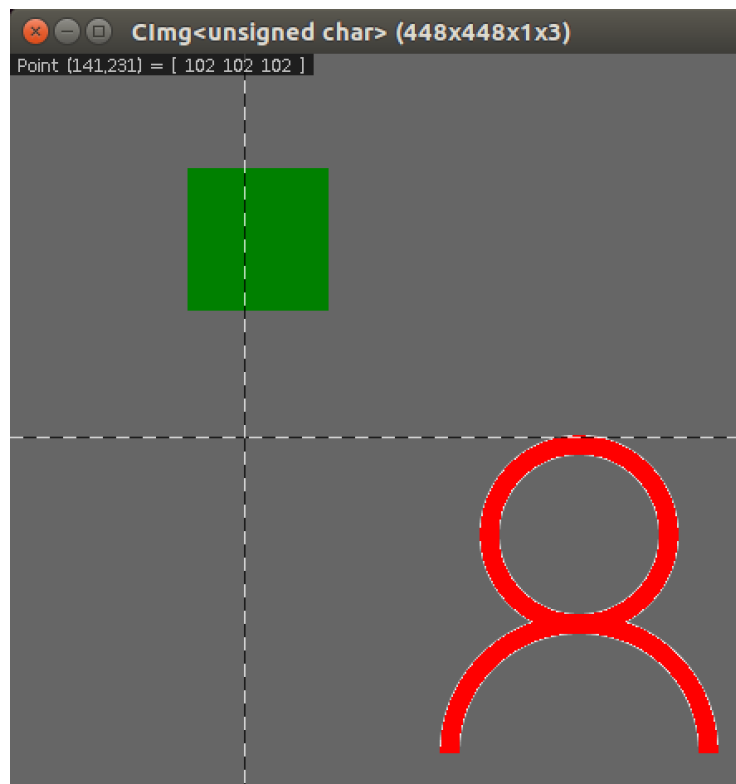
2. 把 1.bmp 文件的白色区域变成红色，黑色区域变成绿色

代码如下:

```
//copy each pixel from the origin pic to the new pic
cimg_forXY(image, x, y) {
    if(image(x,y,0) == image(x,y,1) && image(x,y,1) == image(x,y,2)
        && image(x,y,1) > 102) {
        //find the white pixel and change it to red
        step2(x, y, 0) = 255;
        step2(x, y, 1) = 0;
        step2(x, y, 2) = 0;
    } else if(image(x,y,0) == 0 && image(x,y,1) == 0 && image(x,y,2) == 0) {
        //modify black space to green
        step2(x, y, 0) = 0;
        step2(x, y, 1) = 128;
        step2(x, y, 2) = 0;
    } else {
        //copy the rest of the pixels
        step2(x, y, 0) = image(x, y, 0);
        step2(x, y, 1) = image(x, y, 1);
        step2(x, y, 2) = image(x, y, 2);
    }
}

step2.display();
```

通过使用 `cimg_forXY` 来遍历图像的每一个像素点，由于白色的 RGB 为 (255, 255, 255)，所以分别获取每一个像素点的 RGB 三通道的值，若为白色则替换为红色。在试验中，由于有部分边界处的点的 RGB 值非 (255, 255, 255) 会导致留下一些看似为白色但其实不是白色的点产生的“锯齿”（如下图所示）。



为显的得比较美观，故设置为 R，G，B 的值相同且大于 102 的像素点都更改为红色。同理，处理所有 RGB 为(0,0,0)的点。

实验结果：



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。

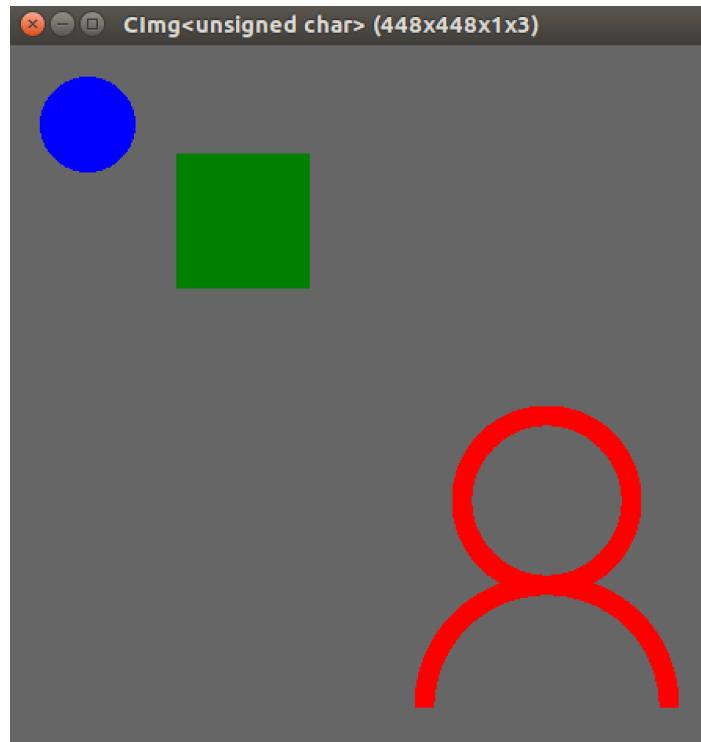
代码如下：

```
CImg<unsigned char> step3(step2);  
step3.draw_circle(50, 50, 30, blue);  
//draw a circle with radius 30 and located at (50,50)  
step3.display();
```

在这一步使用的是 draw_circle()函数在 (50,50) 的位置以 30 为半径画一个蓝色的圆。

实验结果：

如图所示，在图的左上角有一个蓝色的圆形，该步骤按照预期完成。



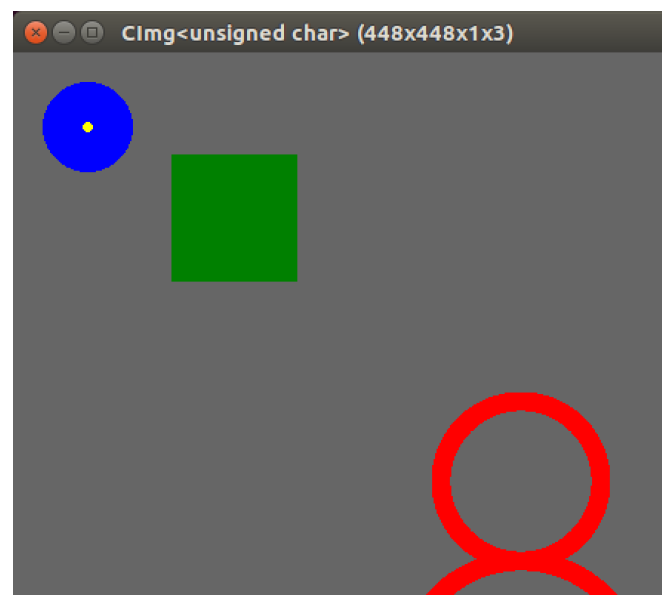
4、在图上绘制一个圆形区域，圆心坐标(50,50)，半径为 3，填充颜色为黄色。

代码如下：

```
CImg<unsigned char> step4(step3);  
//draw another circle with radius 3 and located at (50,50)  
step4.draw_circle(50, 50, 3, yellow);  
step4.display();
```

如上一步的原理相同。

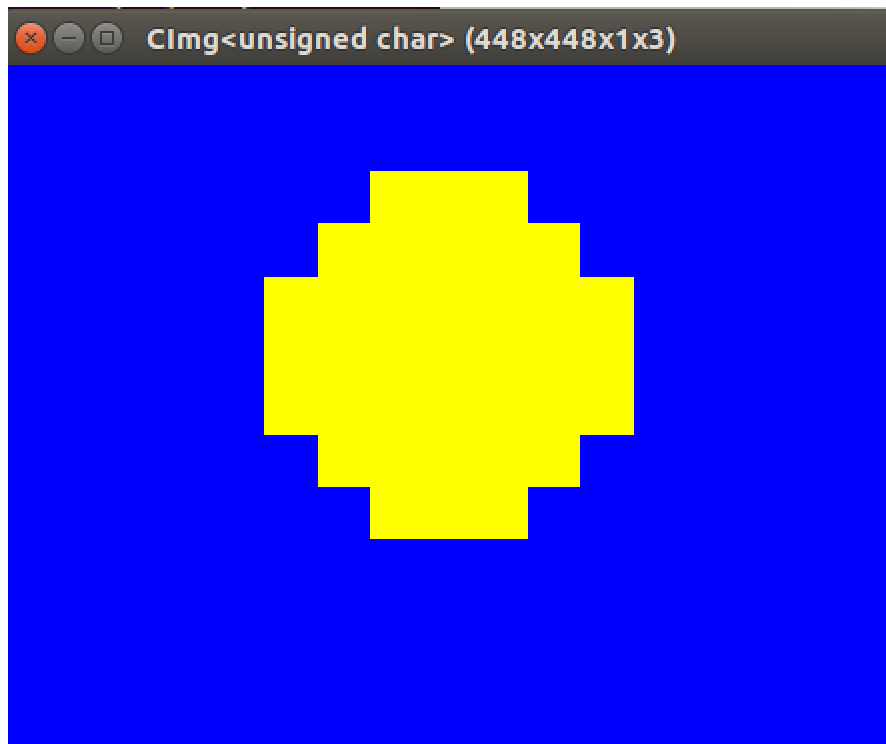
实验结果：



如图可知道该步骤成功。

二、思考部分

1. 为什么第四步绘制的圆形区域形状效果不好。



将黄色的区域放大之后，得到的图行如上图所示，由于计算机是通过一个个的像素进行成像的，由于像素是有大小的，所以在画的图形较小的时候，就会出现上图中那样的误差，导致圆形的显示效果不够好。