计算机视觉第一次作业

16340219 王亮岛

测试环境为windows环境。

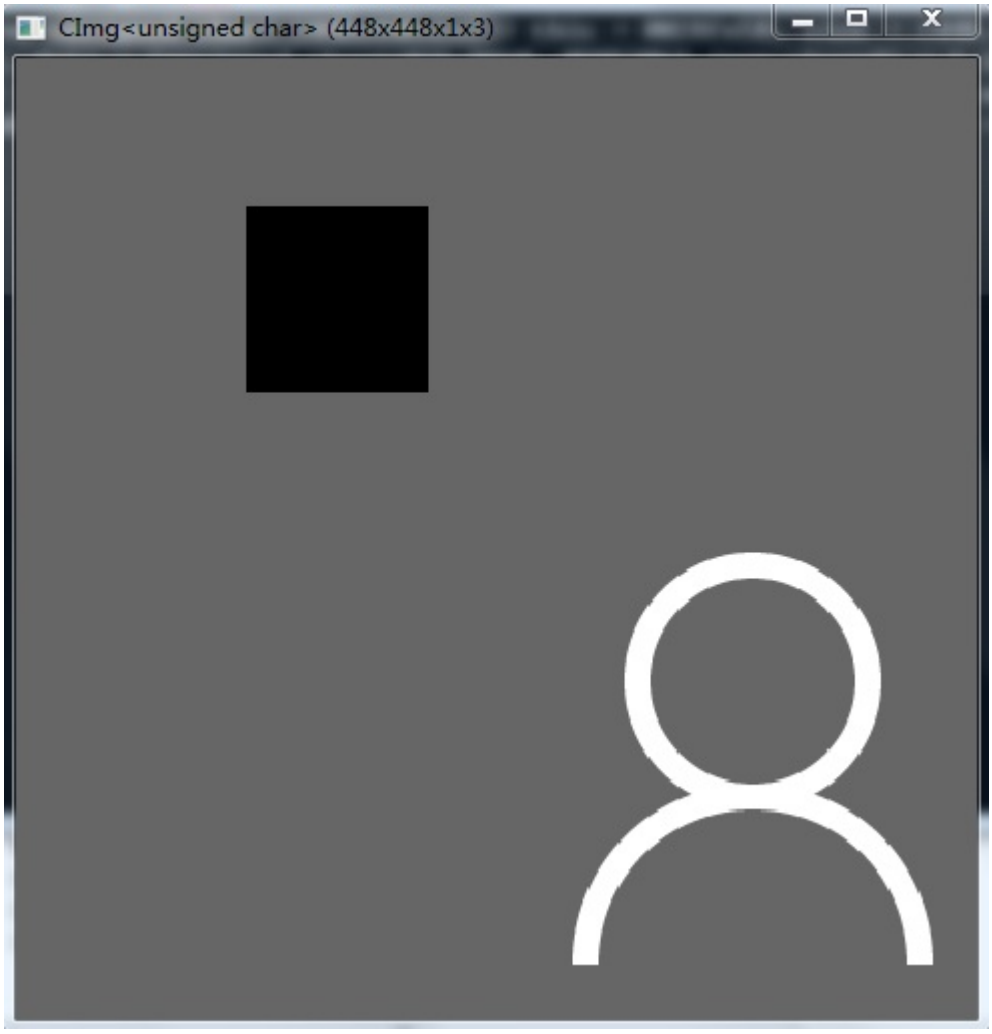
测试数据如下： 画圆的参数：圆心为（50,50），半径分别为30,3，颜色分别为蓝色，黄色。

画直线参数：起点为（0,0），长度为100，角度为35度。

测试代码为test.cpp文件

测试结果以及对结果分析如下：

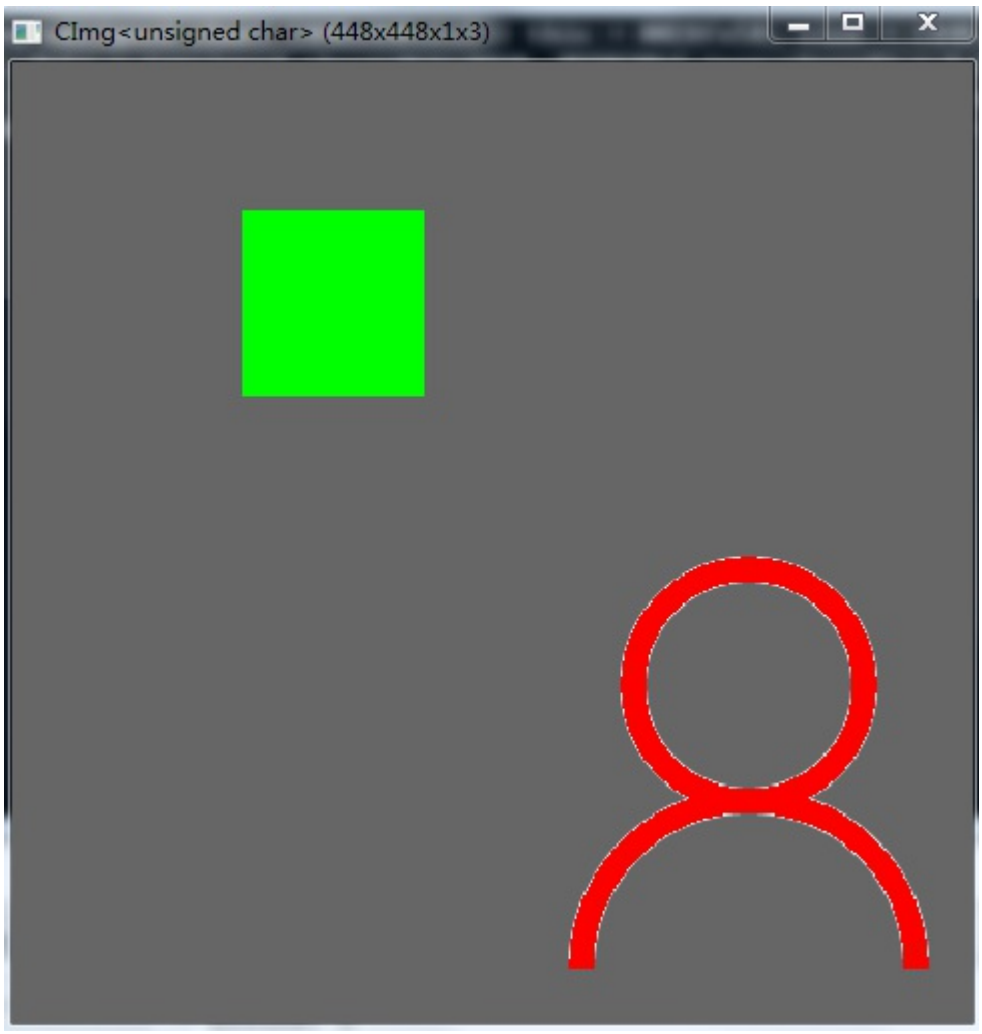
1. 显示原图像



将原图中黑色部分变绿，白色部分变红

```
cimg_forXY(SrcImg, x, y){//白色变红
                if(SrcImg(x, y, 0) == 255 && SrcImg(x, y, 1) == 255 && SrcImg(x, y, 2) == 255){
                    SrcImg(x, y, 0) = 255;
                    SrcImg(x, y, 1) = 0;
                    SrcImg(x, y, 2) = 0;
                }
            }
            cimg_forXY(SrcImg, x, y){//黑色变绿
                if(SrcImg(x, y, 0) == 0 && SrcImg(x, y, 1) == 0 && SrcImg(x, y, 2) == 0){
                    SrcImg(x, y, 0) = 0;
                    SrcImg(x, y, 1) = 255;
                    SrcImg(x, y, 2) = 0;
```

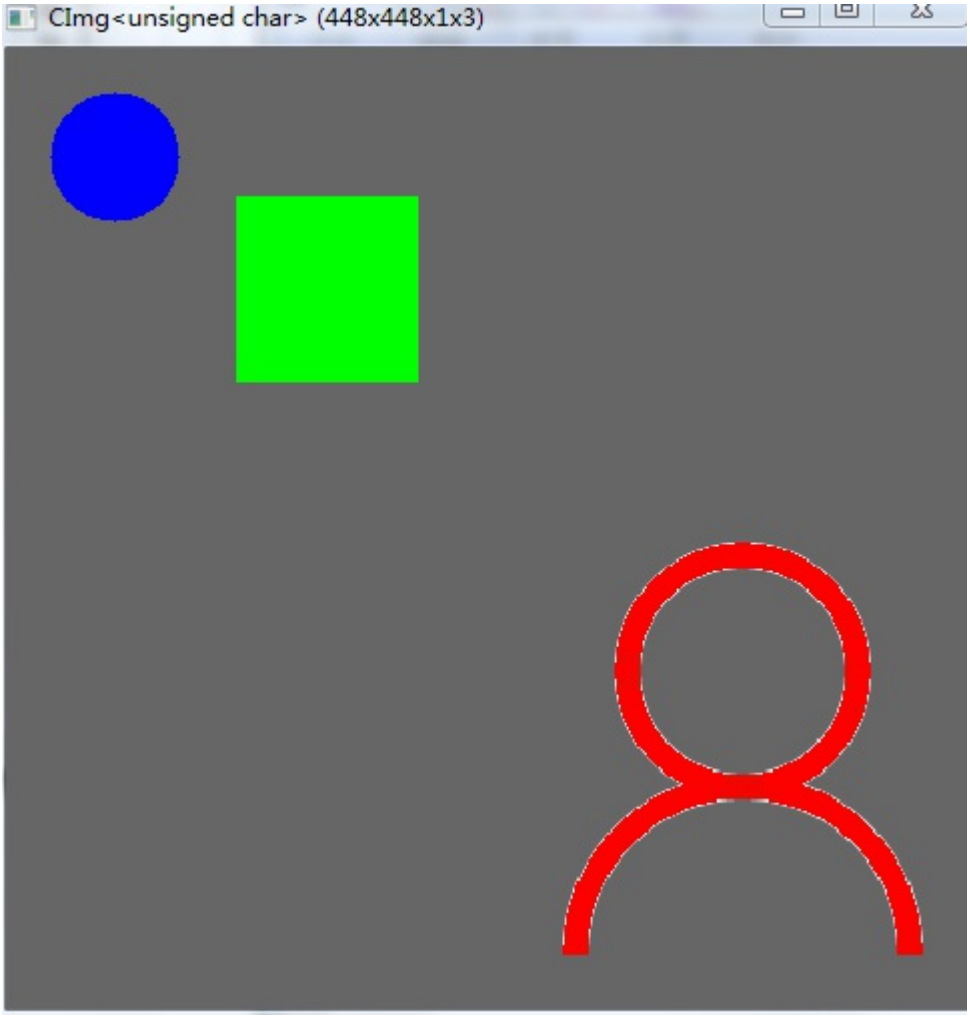
```
        }
    }
```

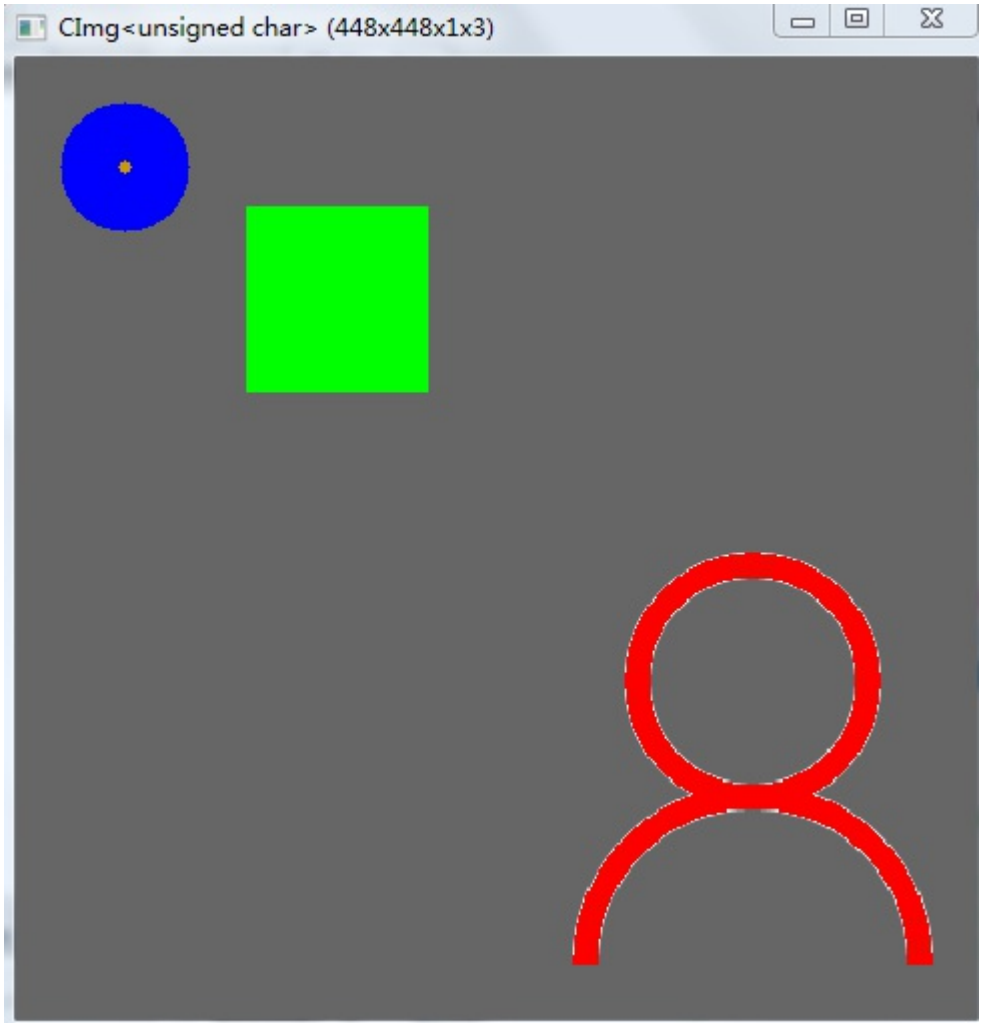


2. 画一个半径为30，圆心在（50,50）处的圆，填充蓝色；画一个半径为3，圆心在（50,50）处的圆，填充黄色；

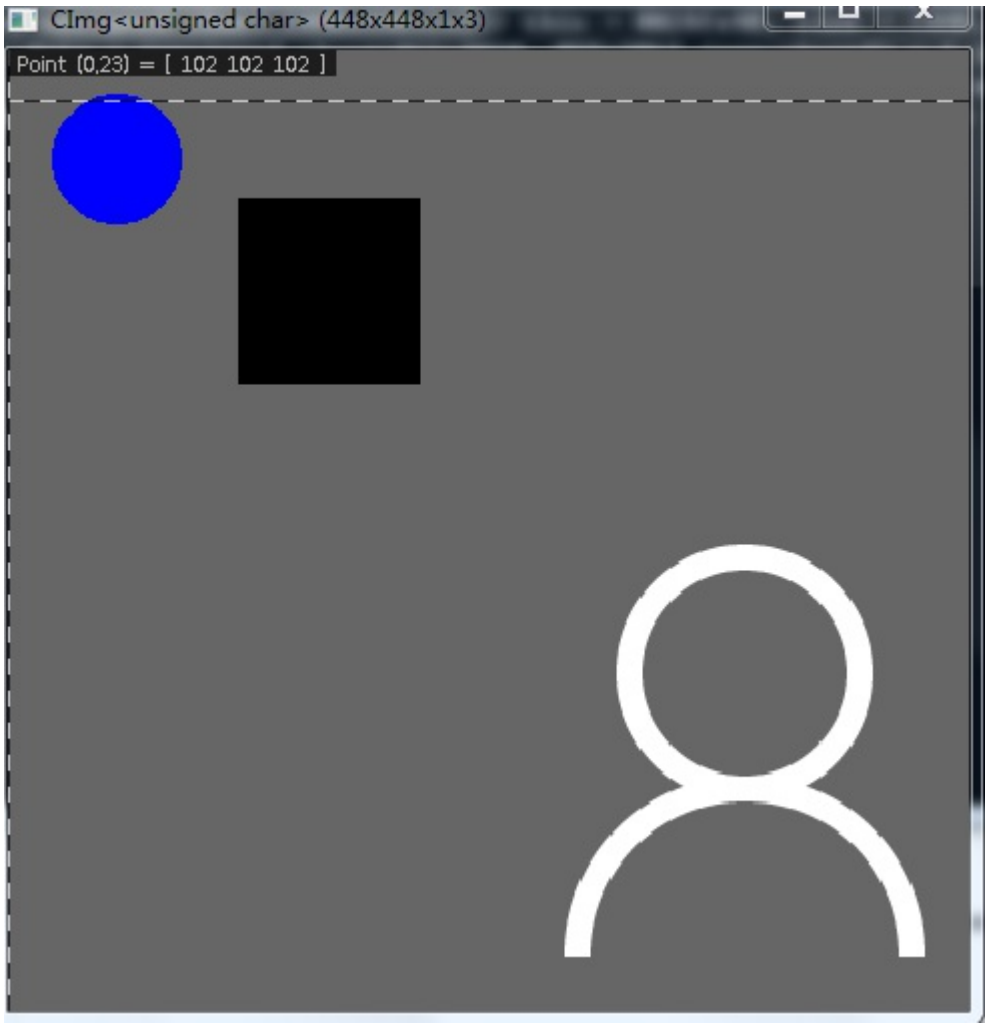
```
cimg_forXY(SrcImg, x, y){
            if(sqrt(pow((x-x0), 2) + pow((y-y0), 2)) <= radius){
                SrcImg(x, y, 0) = R;
                SrcImg(x, y, 1) = G;
                SrcImg(x, y, 2) = B;
            }
        }
```

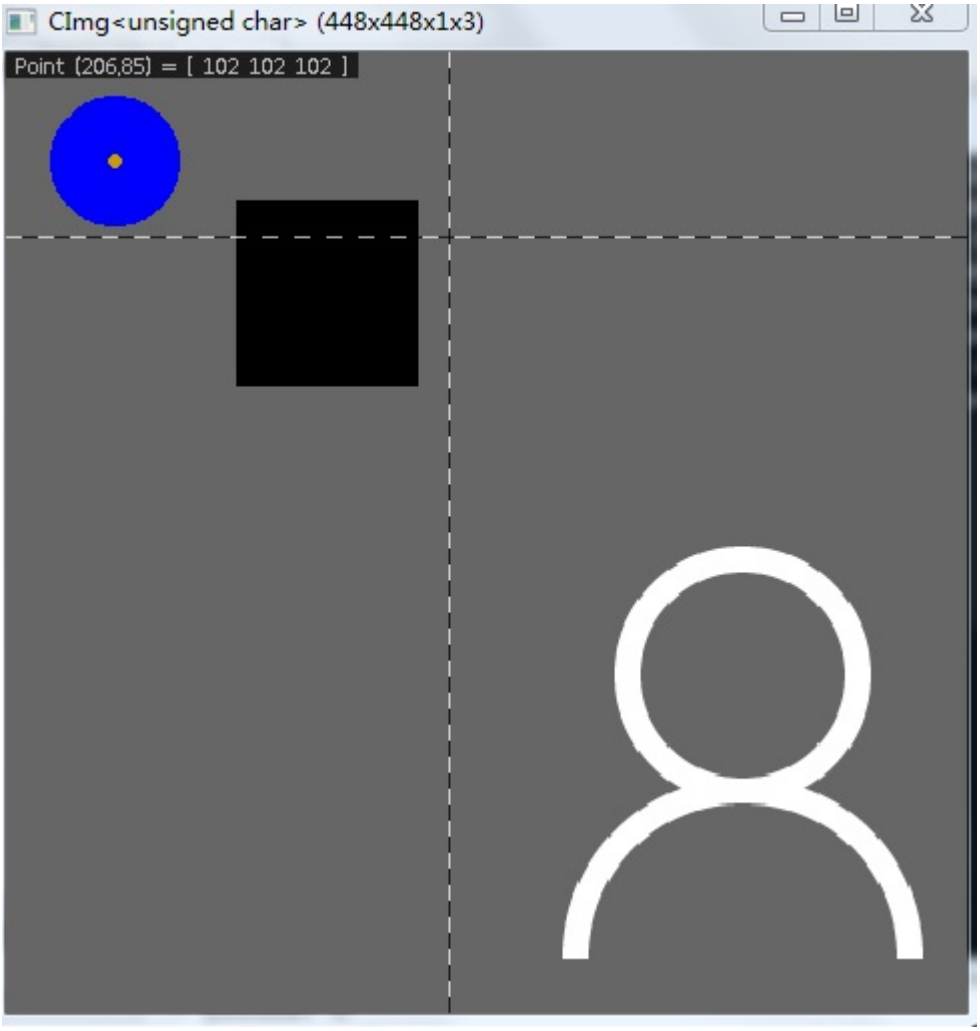
这里我把画圆的函数总结在一起，无论是画什么圆，只要提供圆心位置，半径，填充颜色参数即可。


CImg<unsigned char> (448x448x1x3)



这是用接口函数画出的圆。


CImg<unsigned char> (448x448x1x3)
Point (0,23) = [102 102 102]

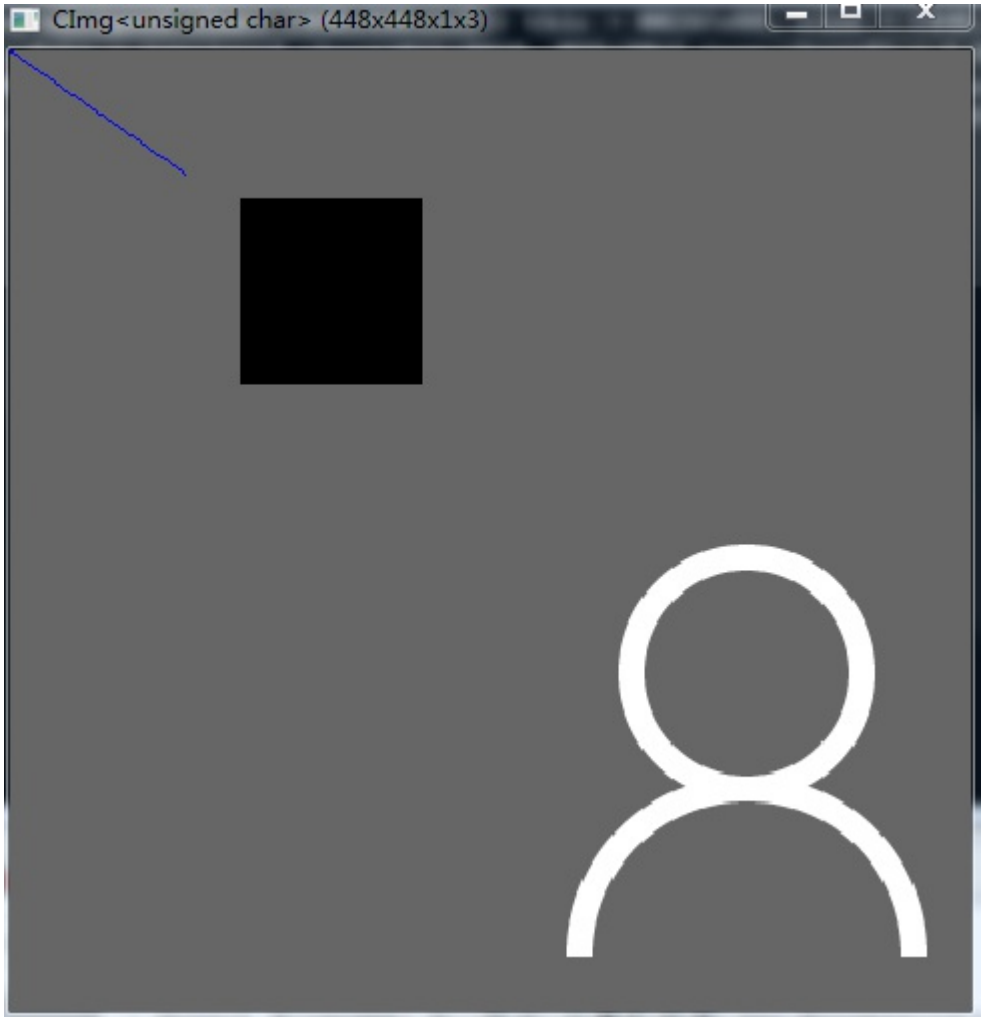


3. 画一条长100，起点为（0,0），角度为35的直线

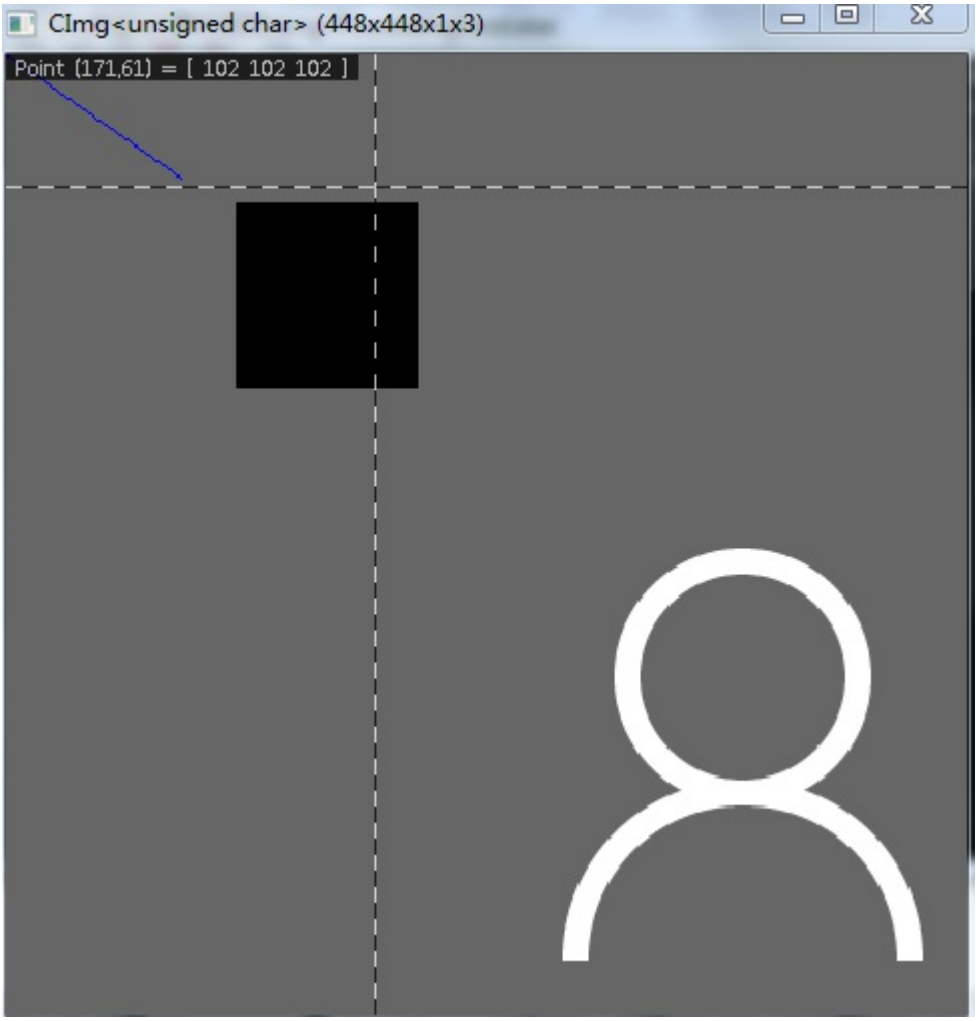
比较函数

```
int cmp(double x, double y){
    if(abs(x-y) <= 0.5){
        return 1;
    }
    return 0;
}
```

```
int x0 = 100*cos(35*pi/180);
            int y0 = 100*sin(35*pi/180);
            cimg_forXY(SrcImg, x, y){
                if(x == 0 && y == 0){
                    SrcImg(x, y, 0) = 0;
                    SrcImg(x, y, 1) = 0;
                    SrcImg(x, y, 2) = 255;
                }
                else if(cmp((double)y, (double)x*tan(35*pi/180)) && (double)x <= x0 && (double)y <=
                    SrcImg(x, y, 0) = 0;
                    SrcImg(x, y, 1) = 0;
                    SrcImg(x, y, 2) = 255;
                }
            }
```



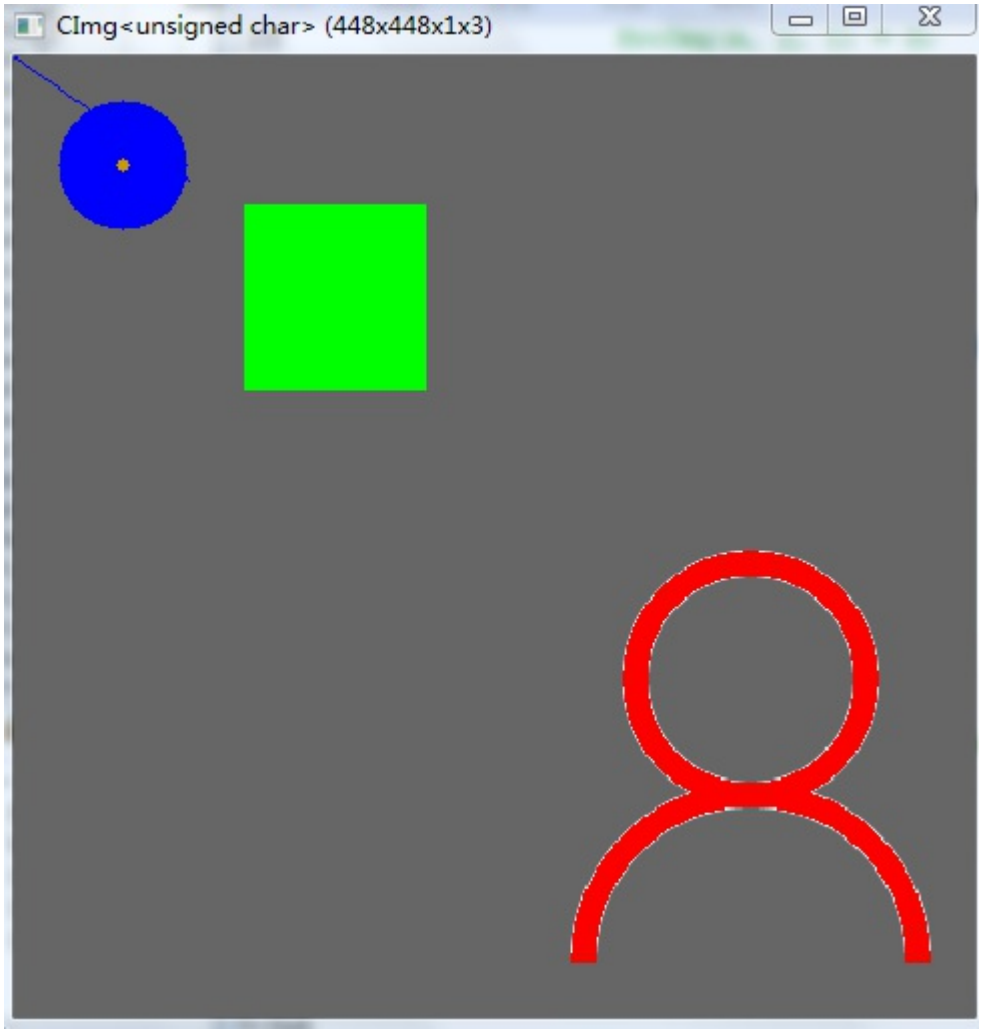
下面使用接口函数实现的：



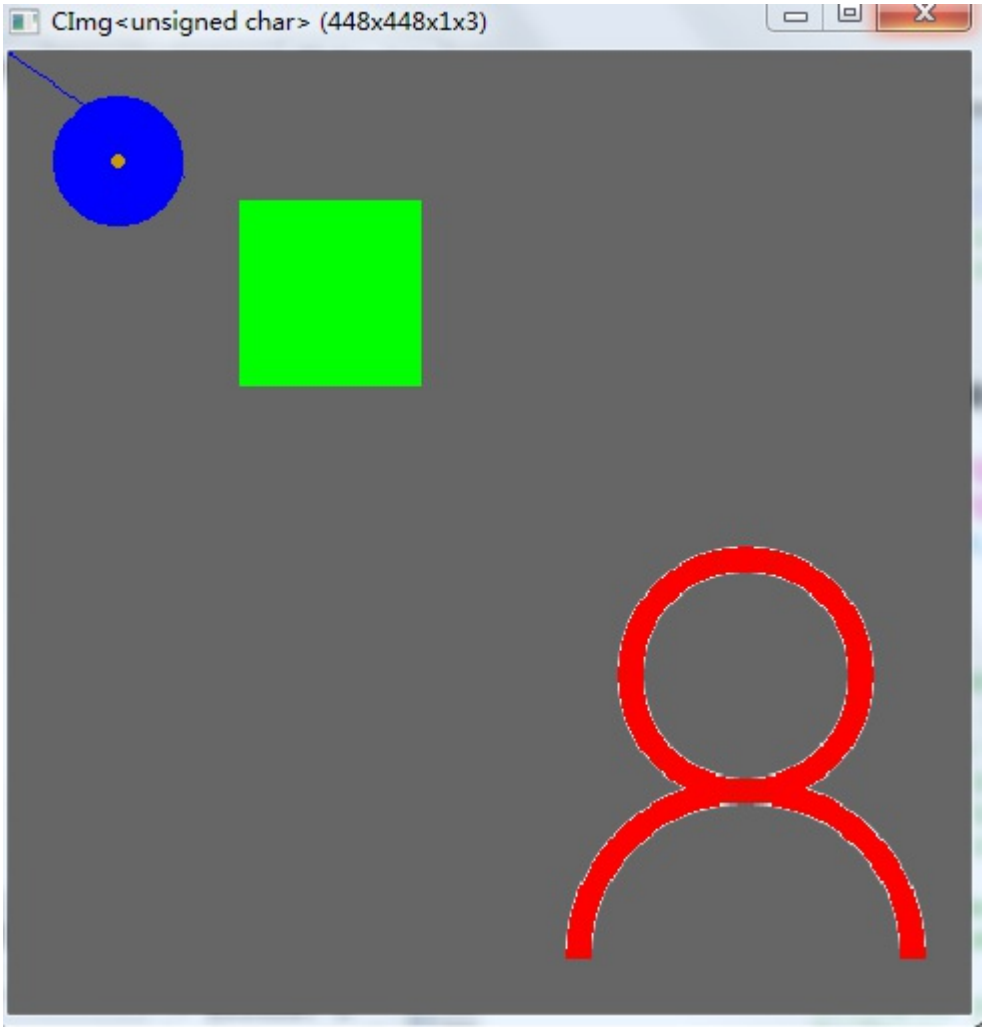
保存为2.bmp

```
CImg<unsigned char> temp = test1.getSrcImg();
temp.save("2.bmp");
```

这是不用接口函数，而是自己逐个像素构建的圆



这是用接口函数构建的图形



对比2,3步的差异

当所画的圆的半径越小，逐步用像素构建的圆会趋向于正方形且轮廓粗糙，就如同黄色的圆所示，而这一缺陷在用接口函数时得到了修复，虽然看起来仍然像菱形，但其轮廓更光滑。而对于直线的构建而言，随着比较函数中两个参数间的误差取值减小，画出来的图像会是一些离散的点而不是一条直线，当误差取值在0.5左右时基本和接口函数所画的直线一致。个人觉得这是由于取样个数过少，因为图像448*448，即每一行每一列都有448个像素值，而当一些像素值超出比较函数规定的误差范围时就会使该点的像素值得不到染色，但是如果误差范围选择太大，又会导致一些原本不在直线上的点被染色使整条直线看起来弯曲了。

思考

对于第三步中画黄色的圆形区域效果不好的原因，个人认为是由于染色的时候是对整个像素块染色，而只要像素块有一部分在圆形区域内就被判定为可以染色，因此被染色的会是一整个正方形的像素块进而导致圆形区域看起来像正方形。