EX3 实验报告

15331159 李沁航

实验环境：Ubuntu 16.04 LTS 虚拟机

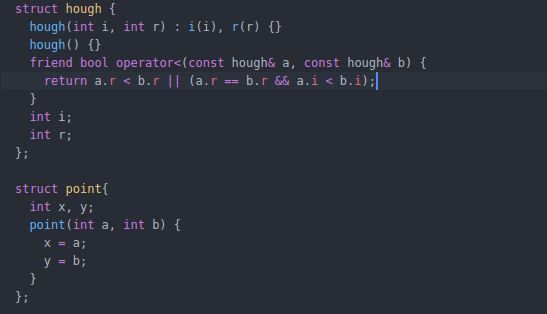
可执行文件：a.out

代码文件：canny.h canny.cpp HoughTransform.h HoughTransform.cpp

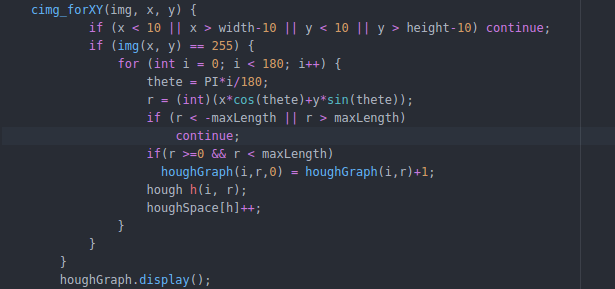
1. 实验部分

3-1 找到图像的四个边缘，提取A4纸的角点，并且输出直线方程。

1. 定义一个结构体hough，用于表示hough space极坐标中的点，以及一个point结构体，表示在二维坐标系中的点。

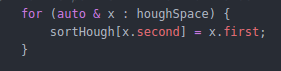


1. 通过使用在上一次作业中canny算子得到的结果，对每一个值为255的像素点，进行hough transform，并存在map<hough, int>类型的变量houghSpace中。

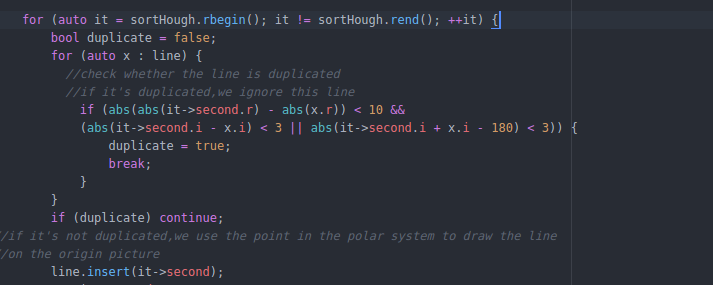


在houghSpace中houghSpace[h]的值就表示的是这个点上的计数值的大小。

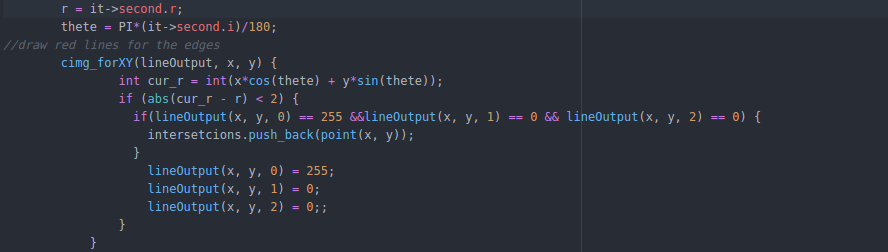
1. 进行排序。



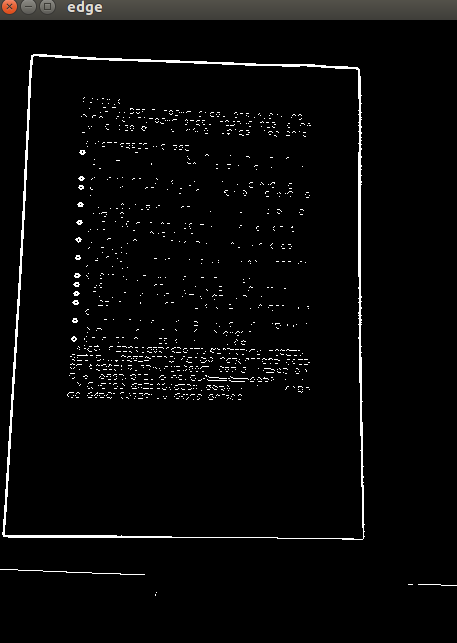
1. 对排序后的结果进行判断，如果判断为在原图像中的同一条直线，则去掉这个结果，否则将其存下来。



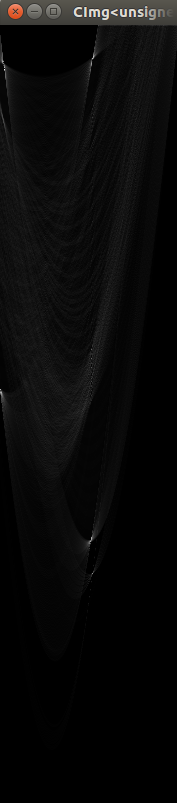
1. 取在极坐标中最“亮”的四个点作为结果，然后对图像中的每一个点，计算它在对应的theta值下映射得到的点，是否与这个四个点中的某一个重合，是的话这个点就作为边界点，然后标记为红色，作为标记的红色边界。并将角点标记出来。



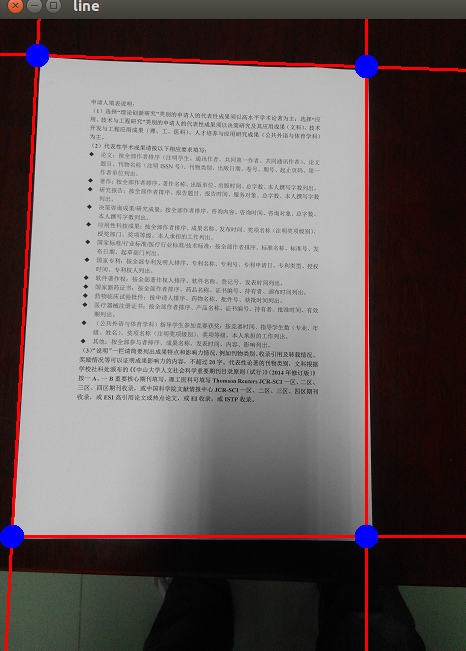
3-1 实验结果。



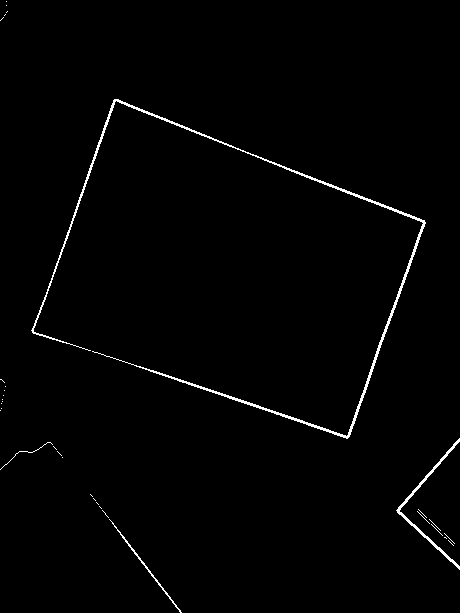
该图片对应的hough space的亮度图：

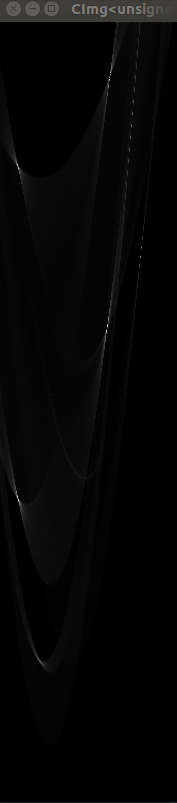
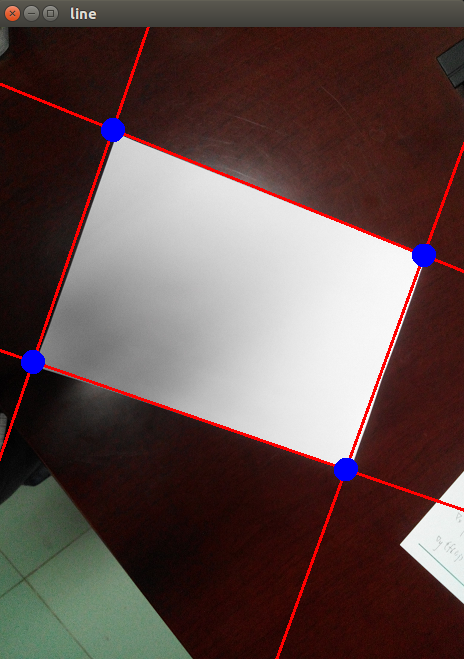


标记完成后的图片为：

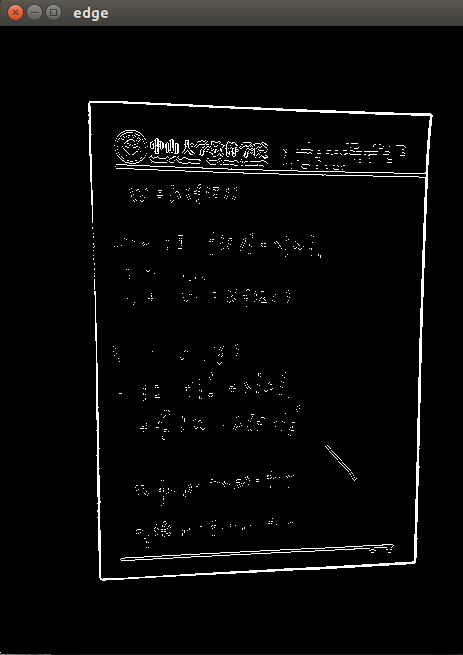


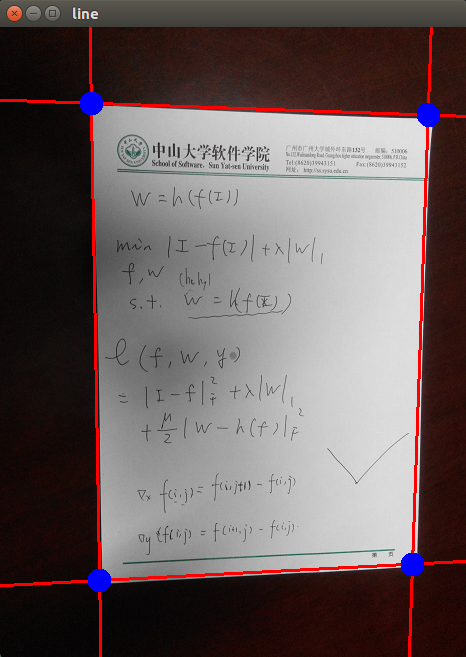
图片二：



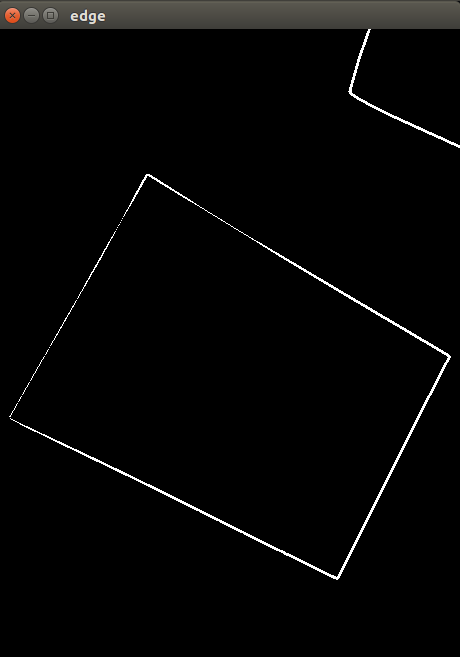
 

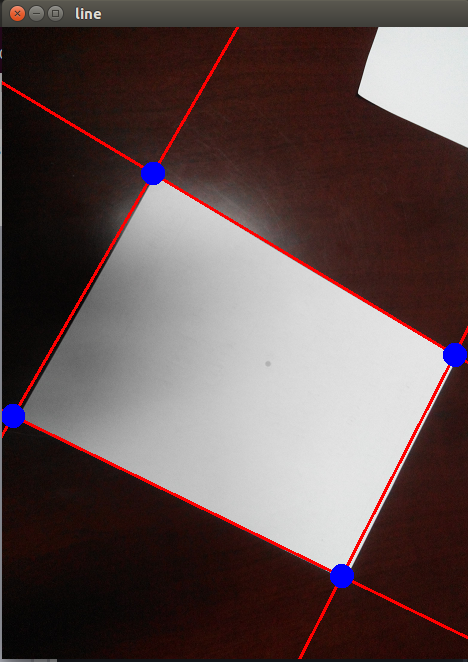
图片3：



图片四：



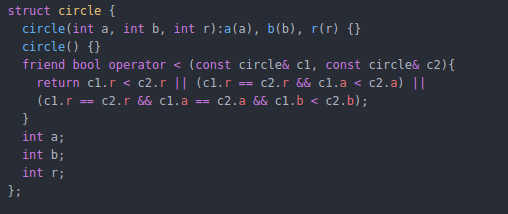
其余的结果已经存在dataSet1中，命名为answer\_i.bmp i为图片对应的编号。

思考：

加快运速度的方法可以使用down sampling的方式，将图片的大小减小，从而能够减少运算次数，加快运行的速度。

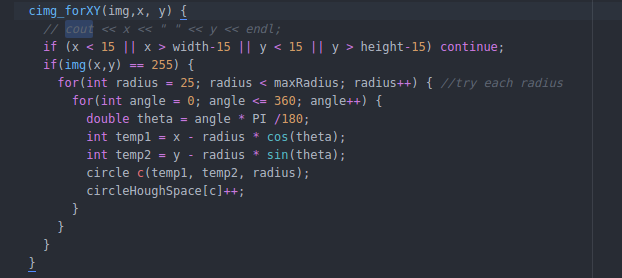
**3-2** 输出图像的边缘，将图像中的边缘拟合成圆，并且输出图片数量。

首先定义结构体circle。



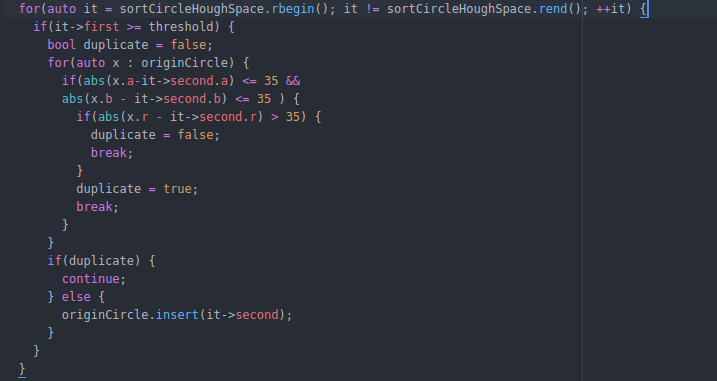
用途与3-1步骤中的类似。

在进行hough transform之前，对图片进行down sampling，可以加快运算的速度。 在找圆的部分，由于不知道圆形的半径，所以需要假定一个半径之后，进行hough transform。所以需要对每一个可能的半径都进行一次运算。最后找出最合适的点。

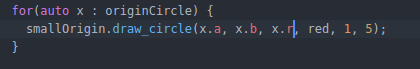


在试验中，圆形半径的区间为[25, height>width?(heigh/2-10):(width/2-10)，对每一个可能的半径进行尝试并且进行映射。

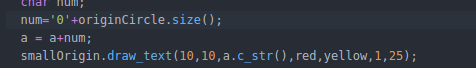
对得到的结果进行排序之后，设定一个threshold，在超过在threshold的情况下，如果不存在duplicate的圆，就认定它为一个圆。



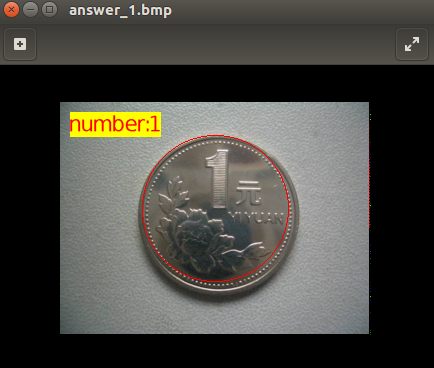
对所有的点检测了之后，在图片上画出标记的圆形。

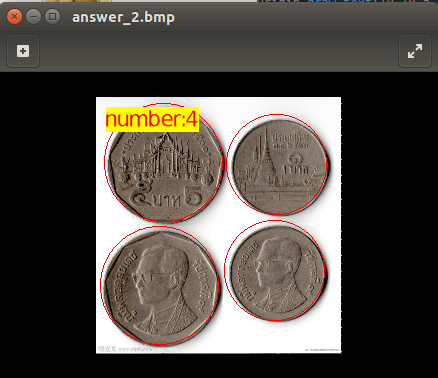


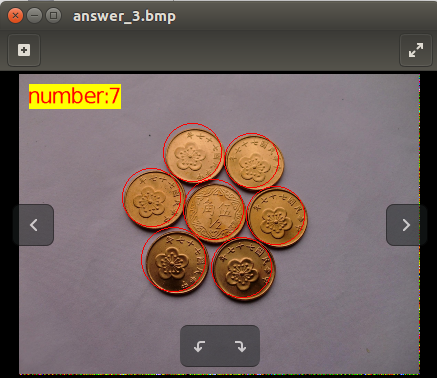
并且标记出找到的个数：

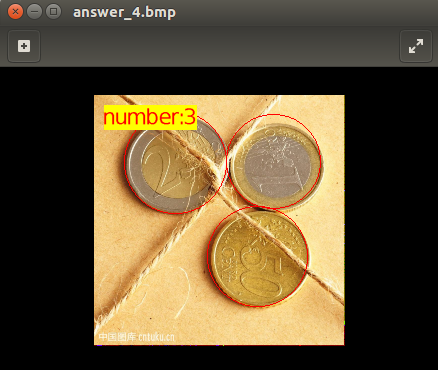


得到的实验结果如图：











得到的结果已经储存在dataset2/answer\_i.bmp中，其中i为图片的编号。