计算机视觉和机器学习 Ex3 测试文档

1. 测试环境

Windows10+VS2015

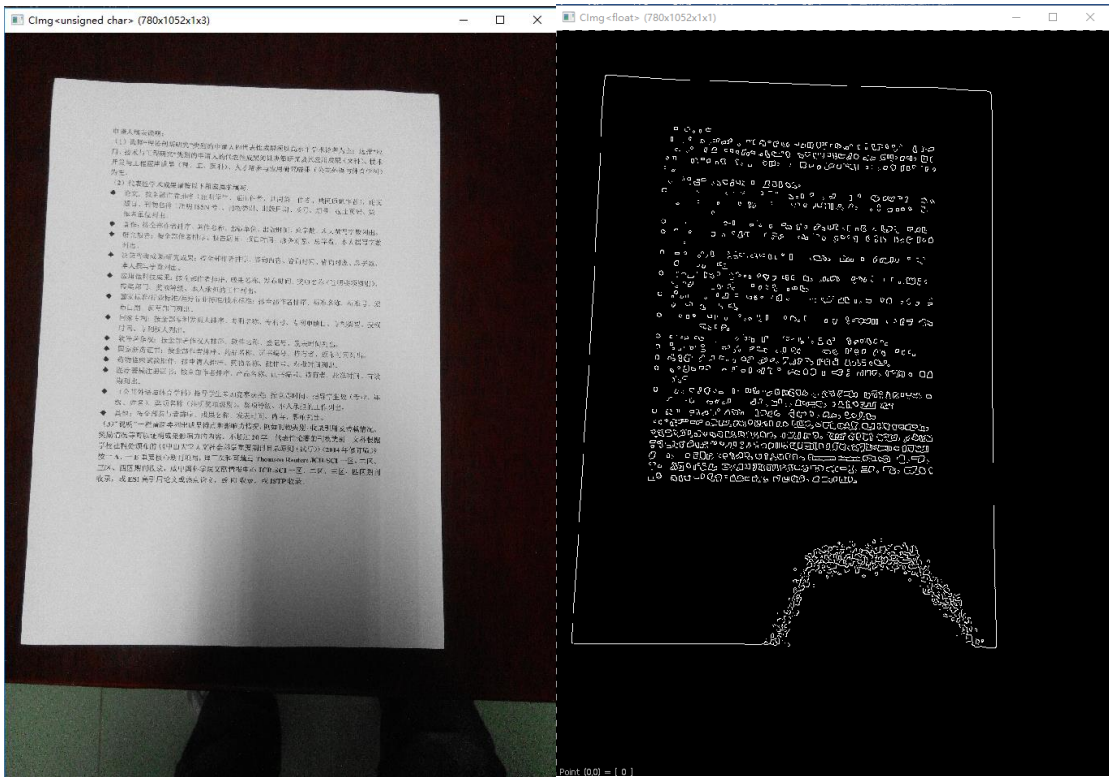
2. 测试数据

老师提供的图片数据

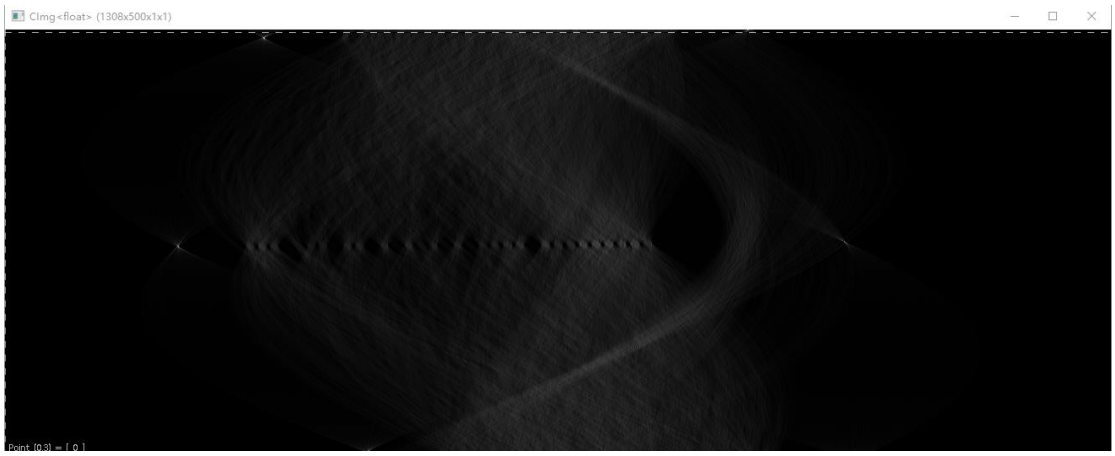
3. 测试结果

直线检测：

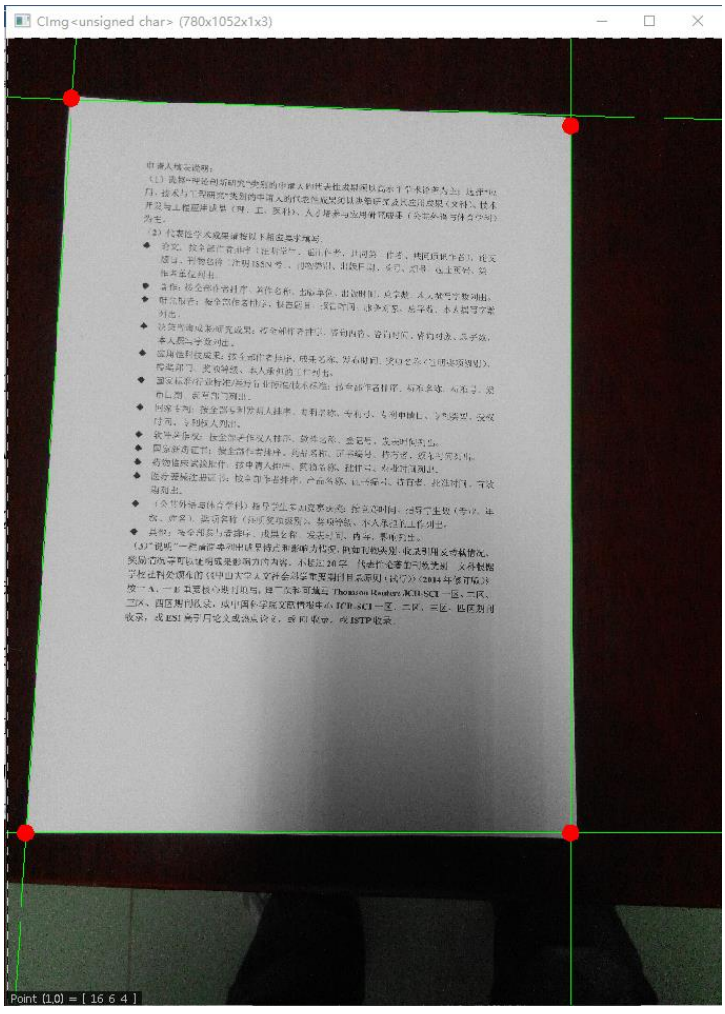
原图和边缘检测结果如下：



Hough 空间如下：


CImg<float> (1308x500x1x1)
Point (0,3) = [0]

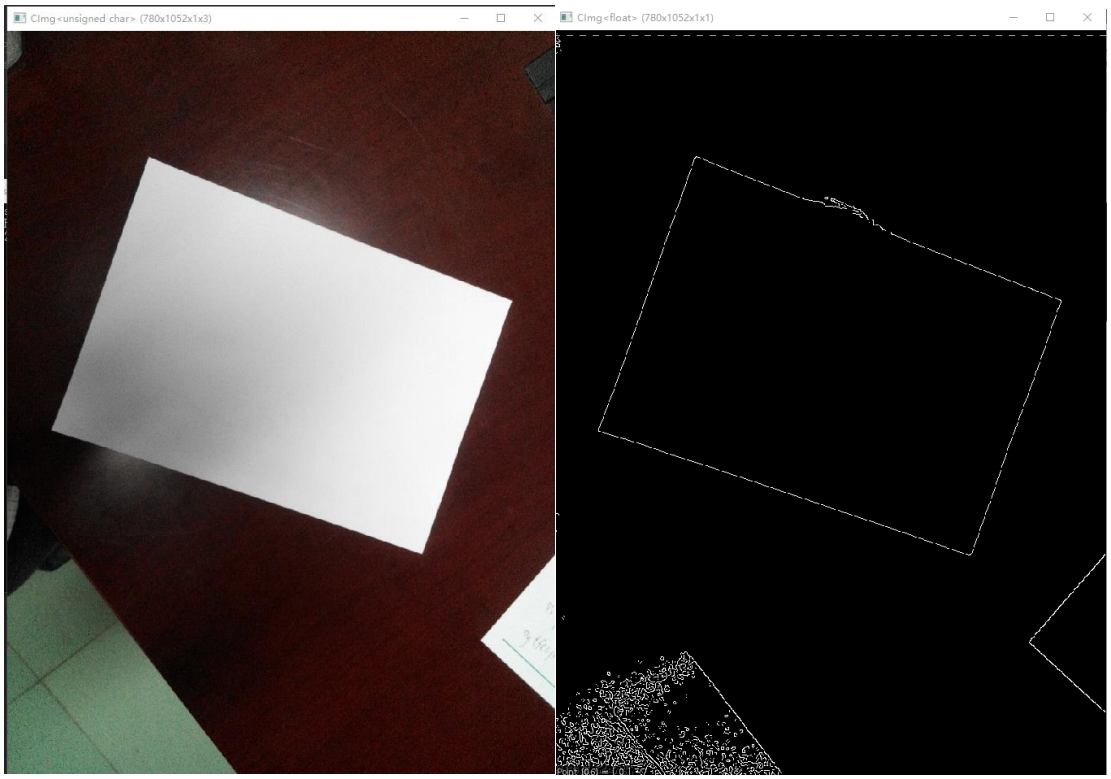
最终结果：

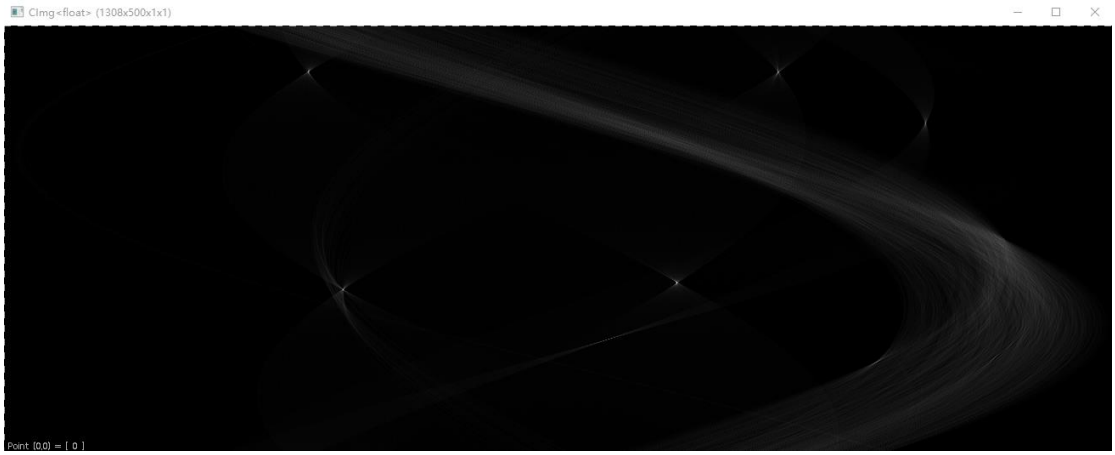

CImg<unsigned char> (780x1052x1x3)
Point (1,0) = [16 6 4]

直线方程如下：

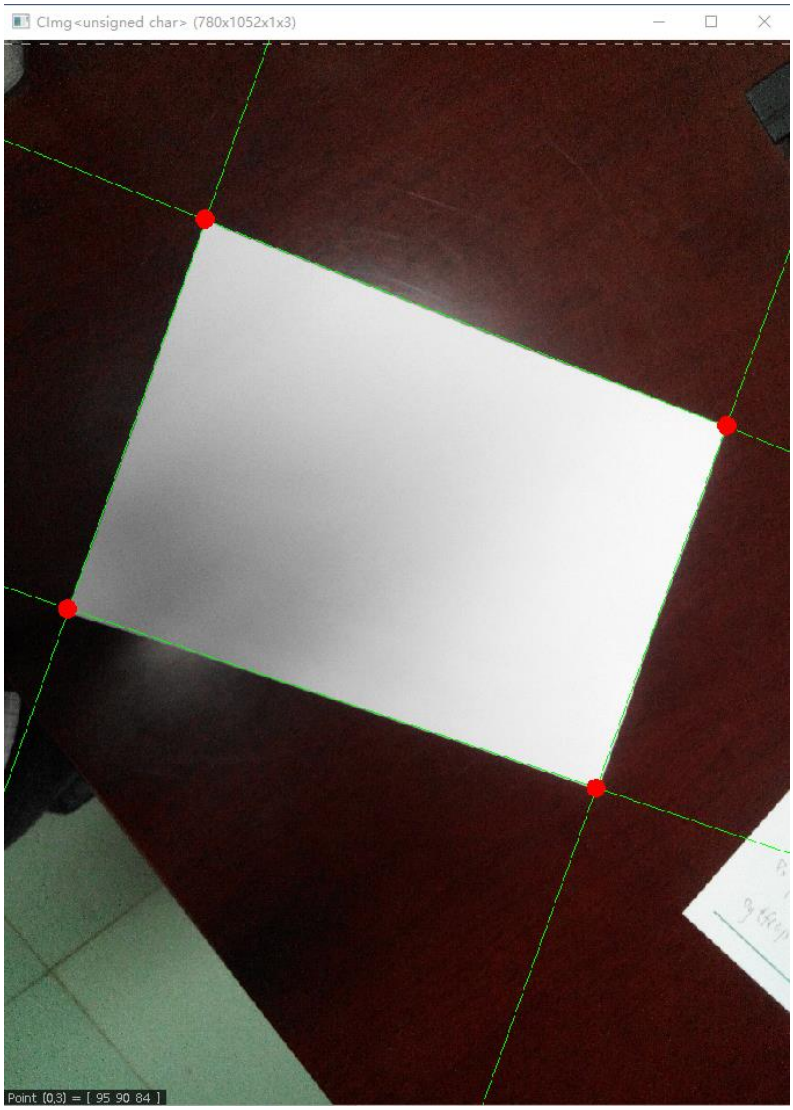
```
line 0 : x = 224
line 1 : y = -15.8945 * x + (-5542.24)
line 2 : y = 338
line 3 : y = 0.0314263 * x + (-450.222)
CImg<unsigned char> (780x1052x1x3): this = 00BCFA1C, size =
04FF602F (non-shared) = [ 31 16 19 21 23 25 32 26 ... 70 47
 = 79.1692, coords_min = (75,0,0,0), coords_max = (68,56,0,
```

第二张图：

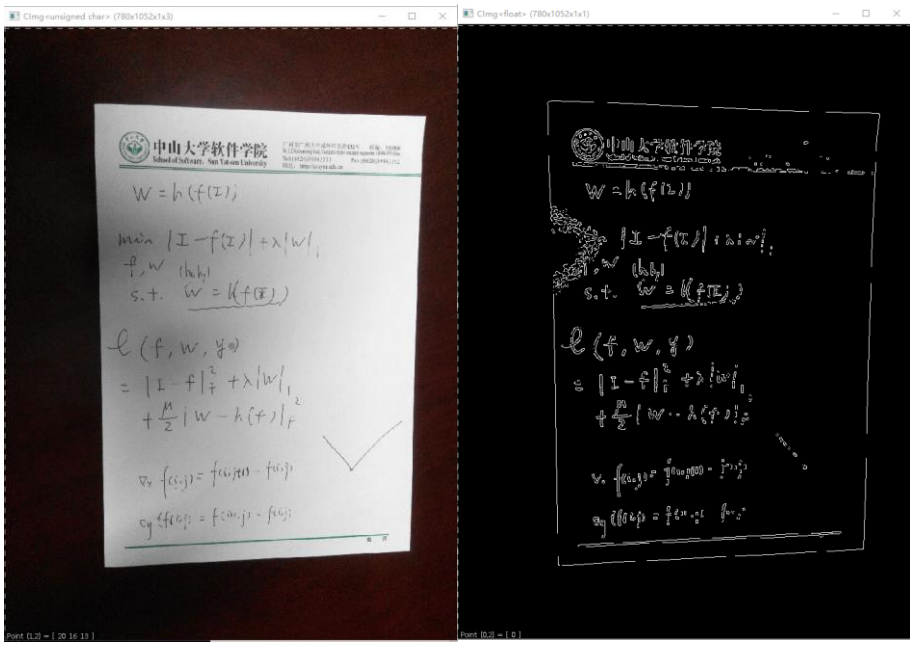


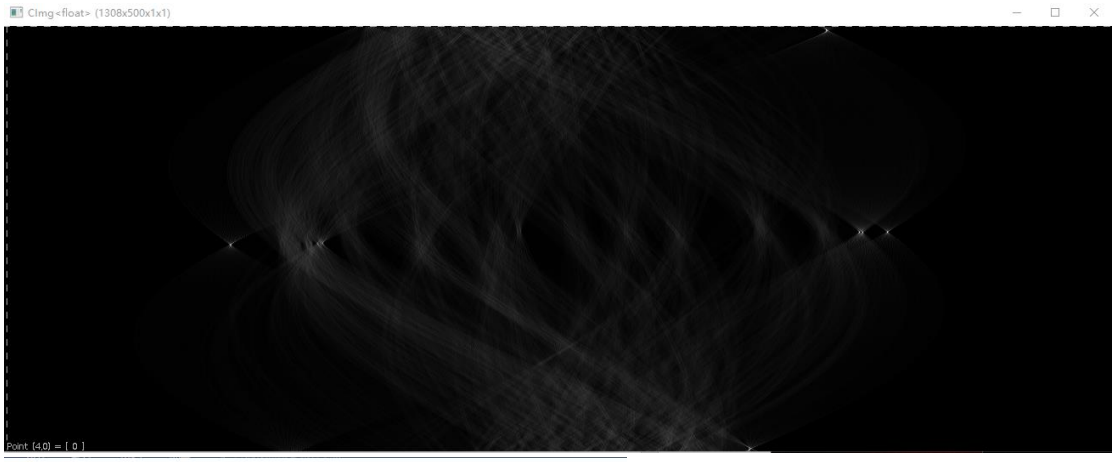


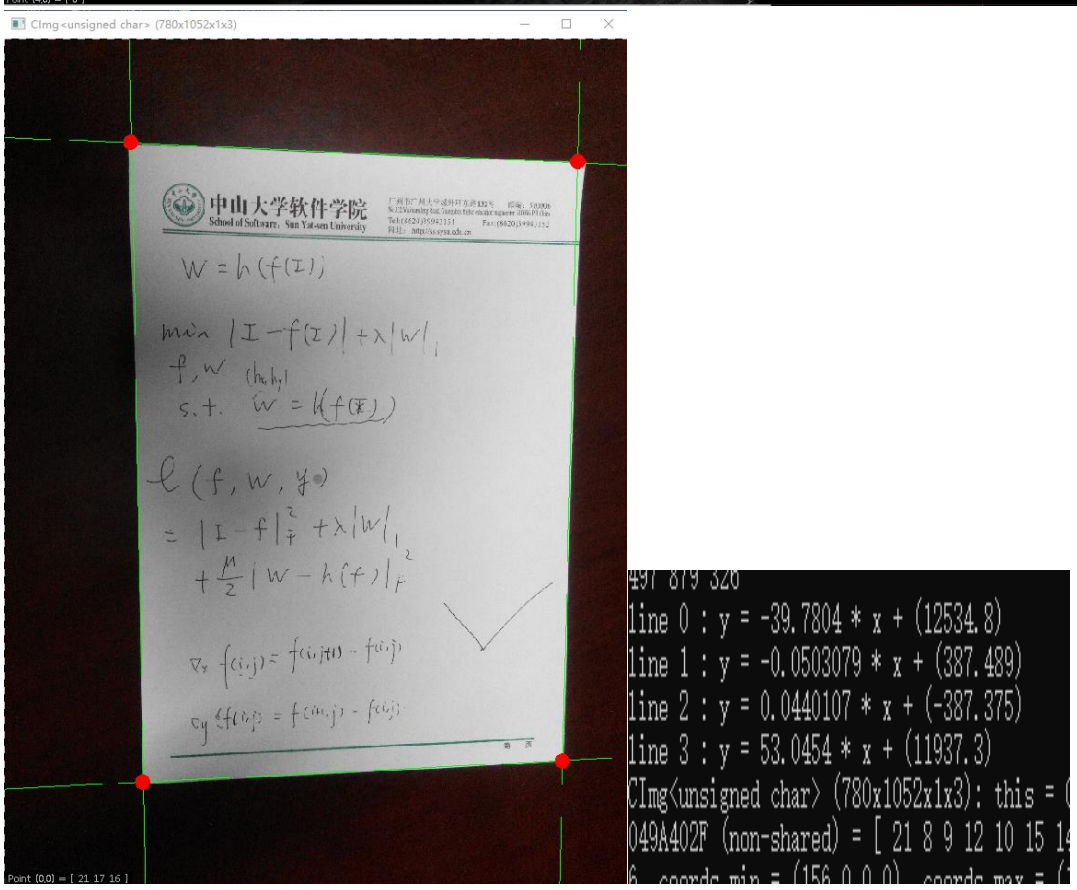

```
310 399 264
line 0 : y = -2.83334 * x + (-889.371)
line 1 : y = -2.77761 * x + (758.699)
line 2 : y = 0.33887 * x + (145.708)
line 3 : y = 0.395928 * x + (-274.259)
CImg<unsigned char> (780x1052x1x3): this =
```


CImg<unsigned char> (780x1052x1x3)
Point (0,3) = [95 90 84]

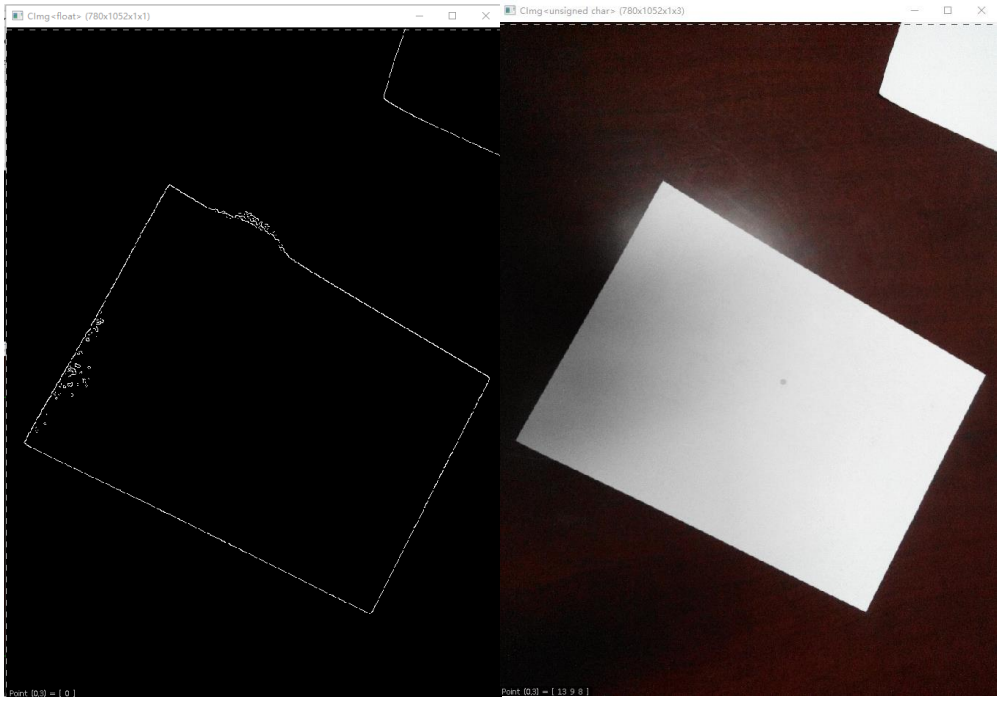
第三张图：

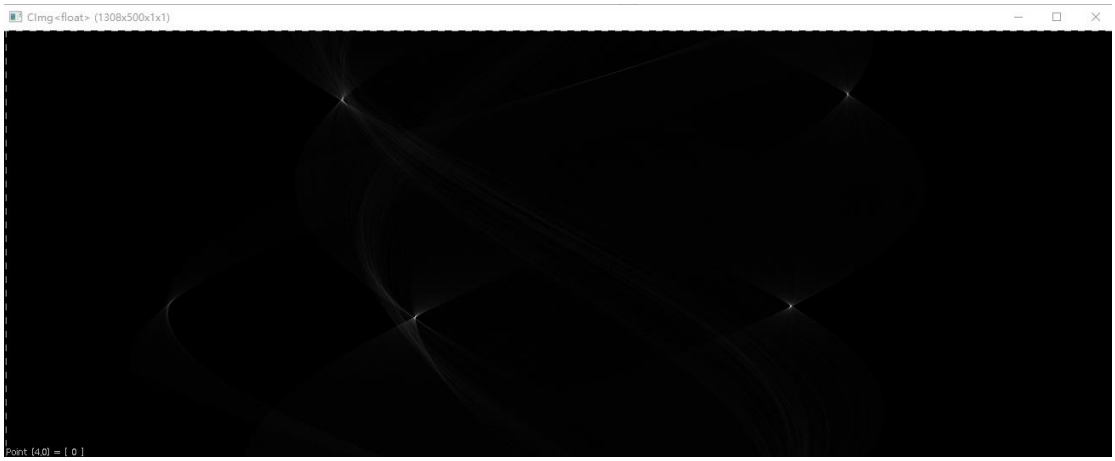

CImg<unsigned char> (780x1052x1x3)
CImg<float> (780x1052x1x1)

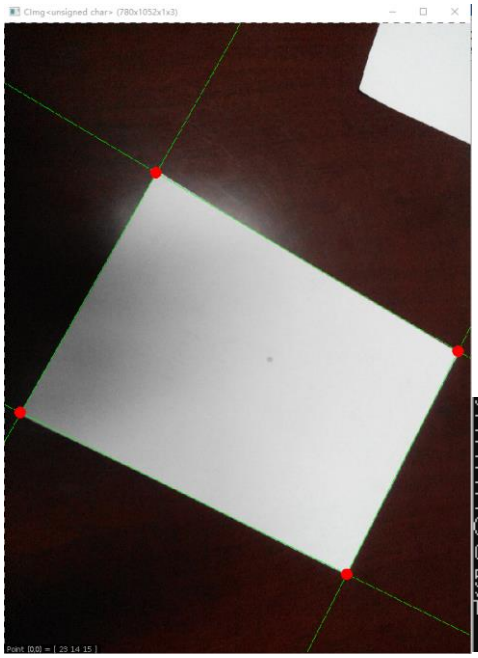

CImg<float> (1308x500x1x1)
Point (4,0) = [0]


CImg<unsigned char> (780x1052x1x3)
中山大学软件学院
School of Software, Sun Yat-sen University
Point (0,0) = [21 17 16]
line 0 : y = -39.7804 * x + (12534.8)
line 1 : y = -0.0503079 * x + (387.489)
line 2 : y = 0.0440107 * x + (-387.375)
line 3 : y = 53.0454 * x + (11937.3)
CImg<unsigned char> (780x1052x1x3): this =
049A402F (non-shared) = [21 8 9 12 10 15

第四张图：

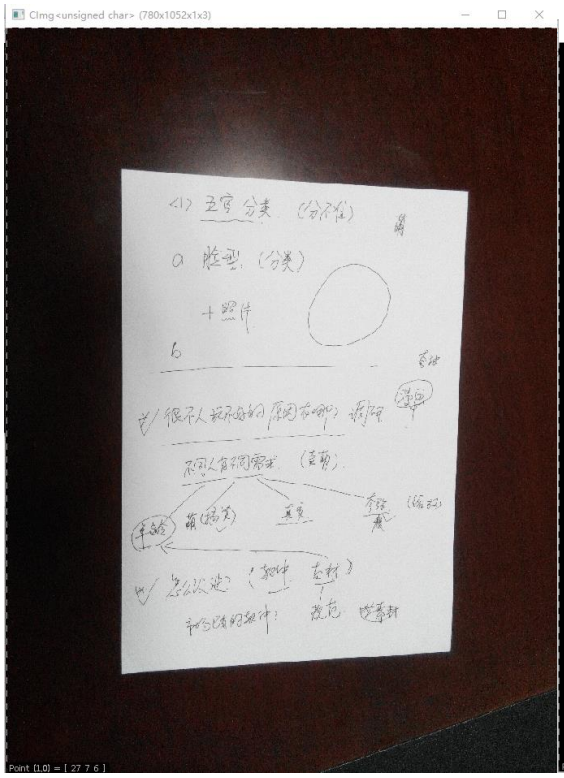


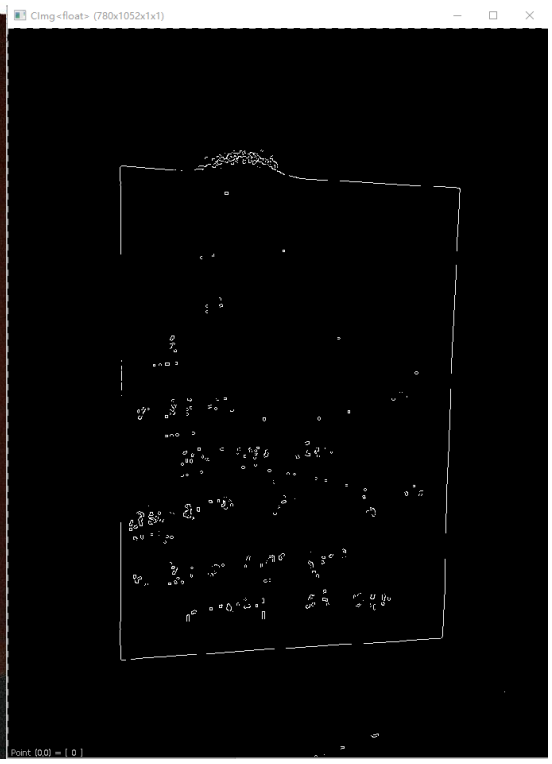




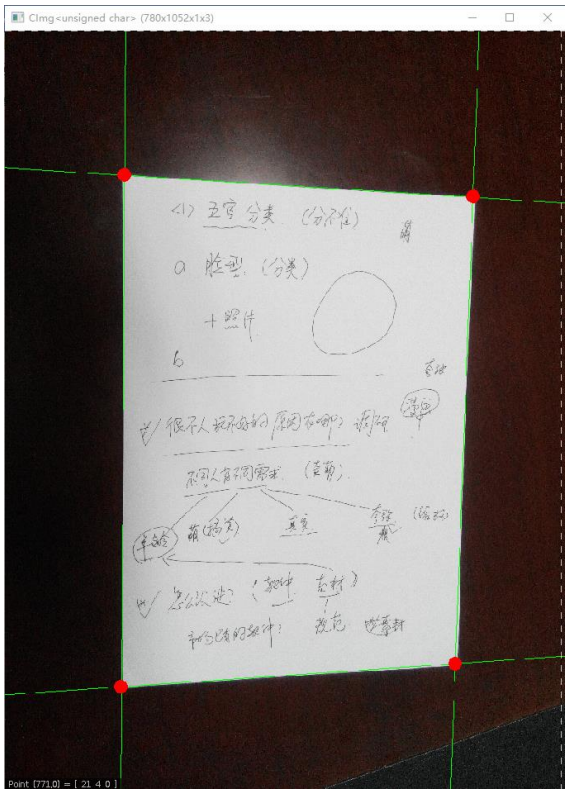

```
line 0 : y = -1.99348 * x + (756.05)
line 1 : y = -1.76606 * x + (-517.528)
line 2 : y = 0.493797 * x + (303.354)
line 3 : y = 0.591398 * x + (-196.342)
CImg<unsigned char> (780x1052x1x3): this = 00F
053E702F (non-shared) = [ 23 17 15 11 14 11 14
5418, coords_min = (394,0,0,0), coords_max = (
请按任意键继续. . .
```

第五张图：

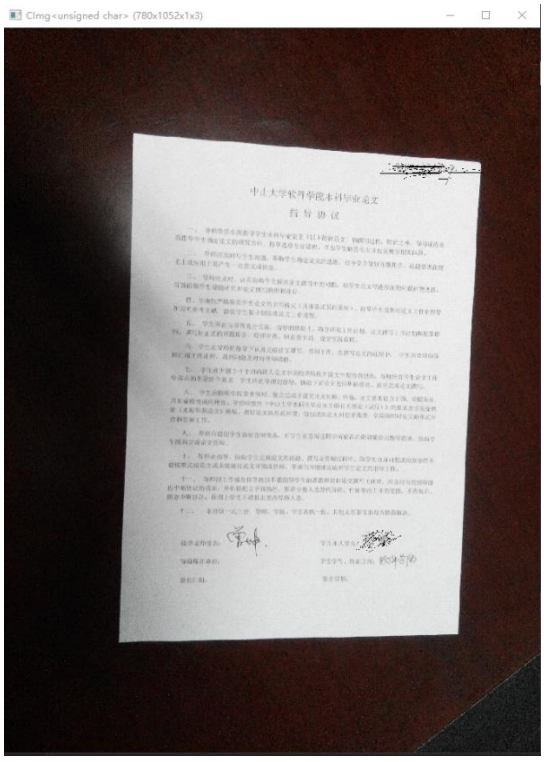


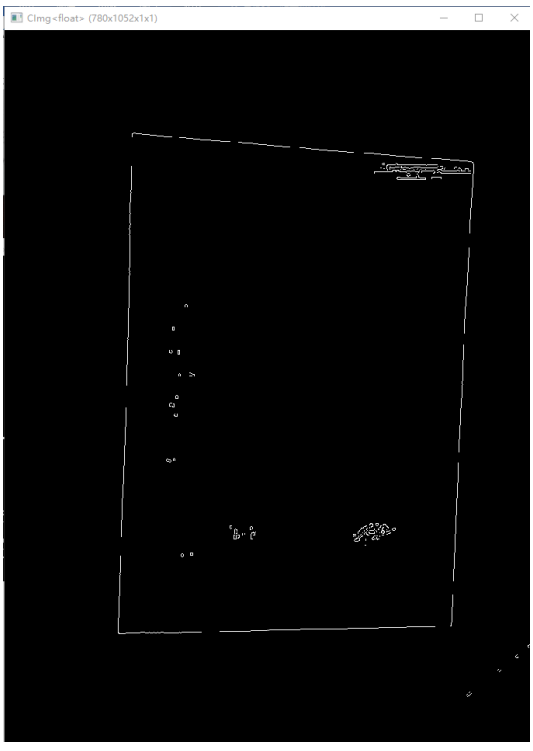


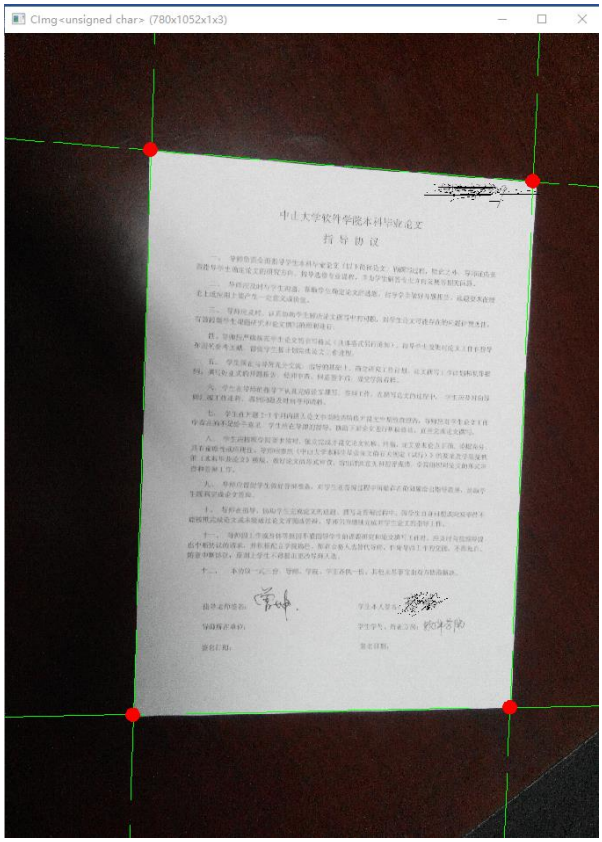

```
260 342 186
line 0 : y = -159.153 * x + (-36128.4)
line 1 : y = -26.5133 * x + (6579.96)
line 2 : y = -0.0692253 * x + (367.878)
line 3 : y = 0.0629147 * x + (-312.617)
CImg<unsigned char> (780x1052x1x3): this = 0071F980, size
04BDB02F (non-shared) = [ 18 27 32 22 39 26 39 41 ... 36 3
= 91.7368, coords_min = (167,0,0,0), coords_max = (163,190
请按任意键继续. . .
```

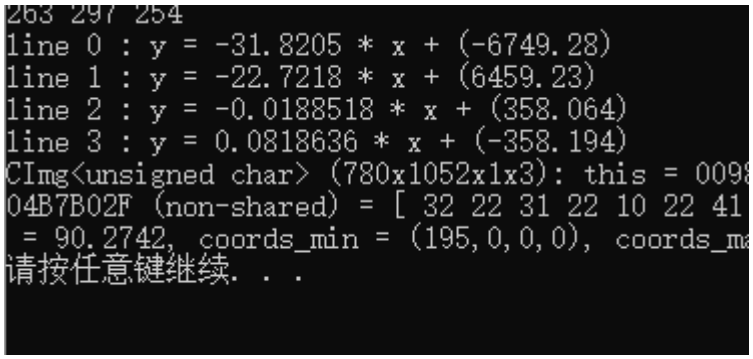

CImg<unsigned char> (780x1052x1x3)
Point [771,0] = [21 4 0]

第六张图：


CImg<unsigned char> (780x1052x1x3)


CImg<float> (780x1052x1x1)





如上所见，使用霍夫变换，所有图都很好的检测除了 A4 纸的边缘直线，并且去除了一些噪声的干扰，这里主要是一开始的时候对图片进

行了二值化处理，去除了噪声，因此最终对直线检测都有很好的结果。

思考：如何在保证精度的情况下加快运行速度？

1. 可以使用并行化计算从而加快运行速度。
2. 可以通过一些图像预处理或者调节 canny 边缘检测中的参数，使得最终边缘检测出的轮廓具有较少的噪声点，从而可以降低 hough 变换的开销，加快运行速度。

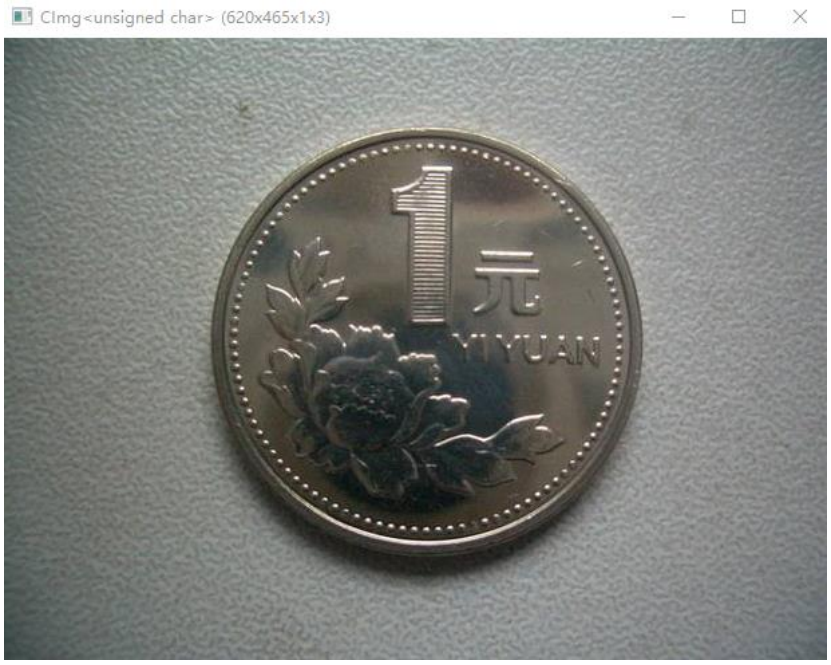
硬币检测：

硬币检测关键在于检测圆周，但是圆周不同于直线，圆周有三个参数，圆心坐标 x,y 和半径 r，因此，如果对圆使用霍夫变换投影到三维空间，将会大大增加计算的开销，因此这里使用的是霍夫变换的一个变种，霍夫梯度法，主要思路如下：

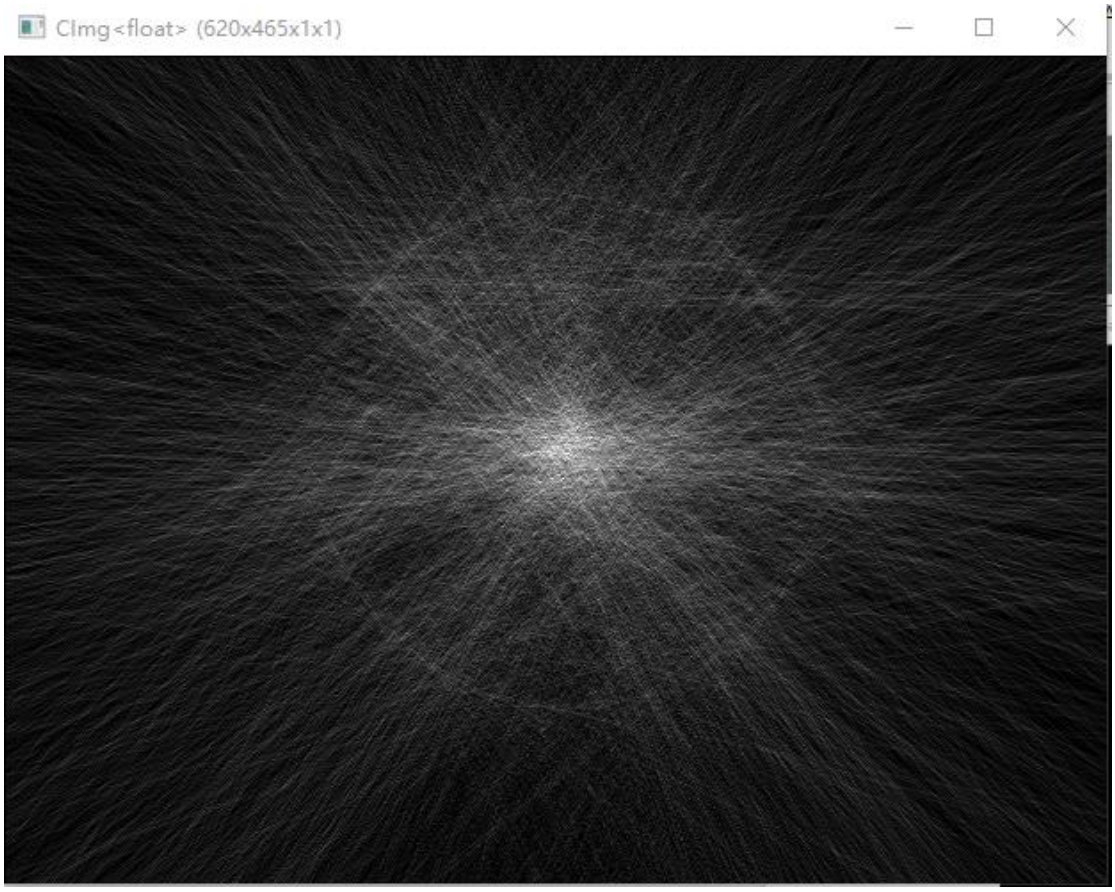
1. 检测圆心
2. 搜索半径

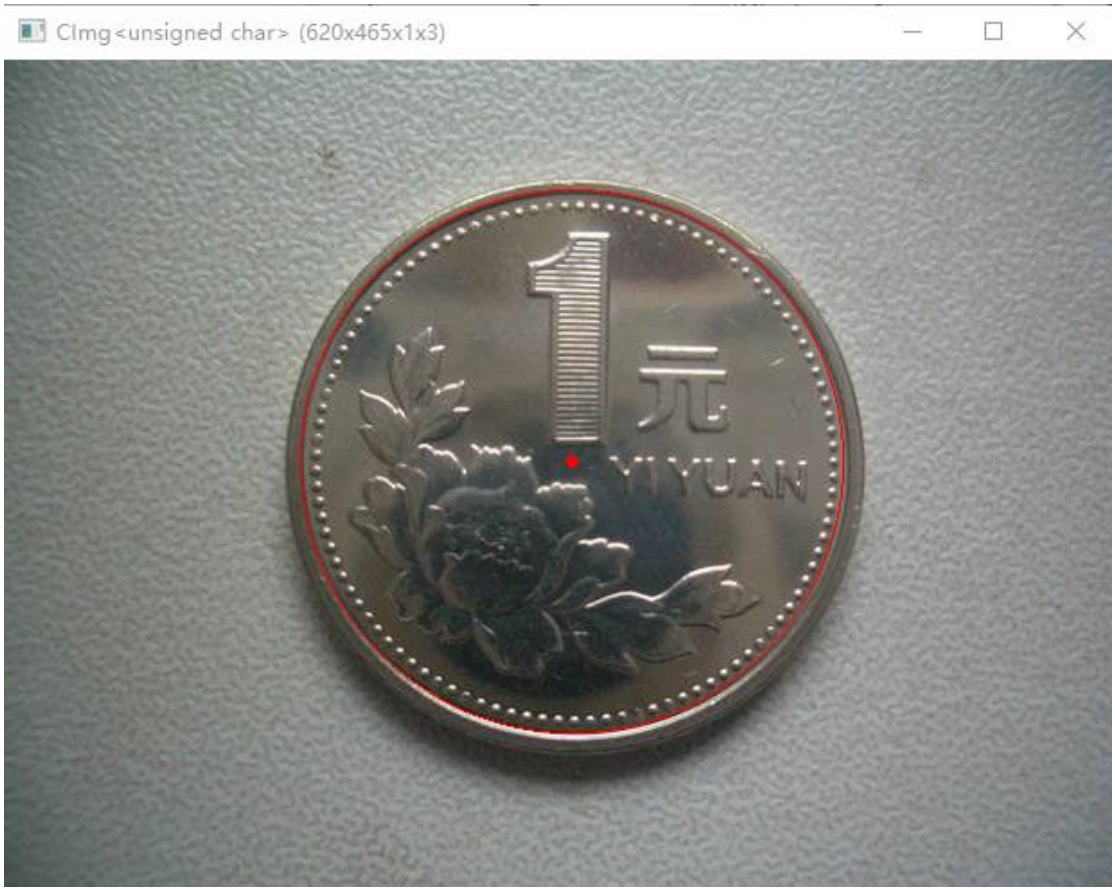
首先通过对边缘检测中所有点画出梯度直线，因为圆周的梯度实现最终交于圆心，因此通过这步操作我们可以将绘出的 hough 空间中最亮的点认定为圆心。

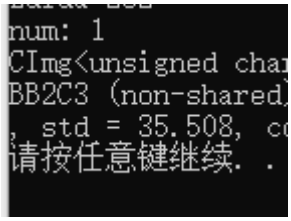
而后通过找出圆心和所有轮廓点中，距离相等最多的那个距离，即可认定为半径，但是这种方法对参数的调节依赖性很大，因为图片大小不一样，很多阈值都很难确定，接下来给出几张较好的效果：


CImg<unsigned char> (620x465x1x3)
1
元
YI YUAN


CImg<float> (620x465x1x1)

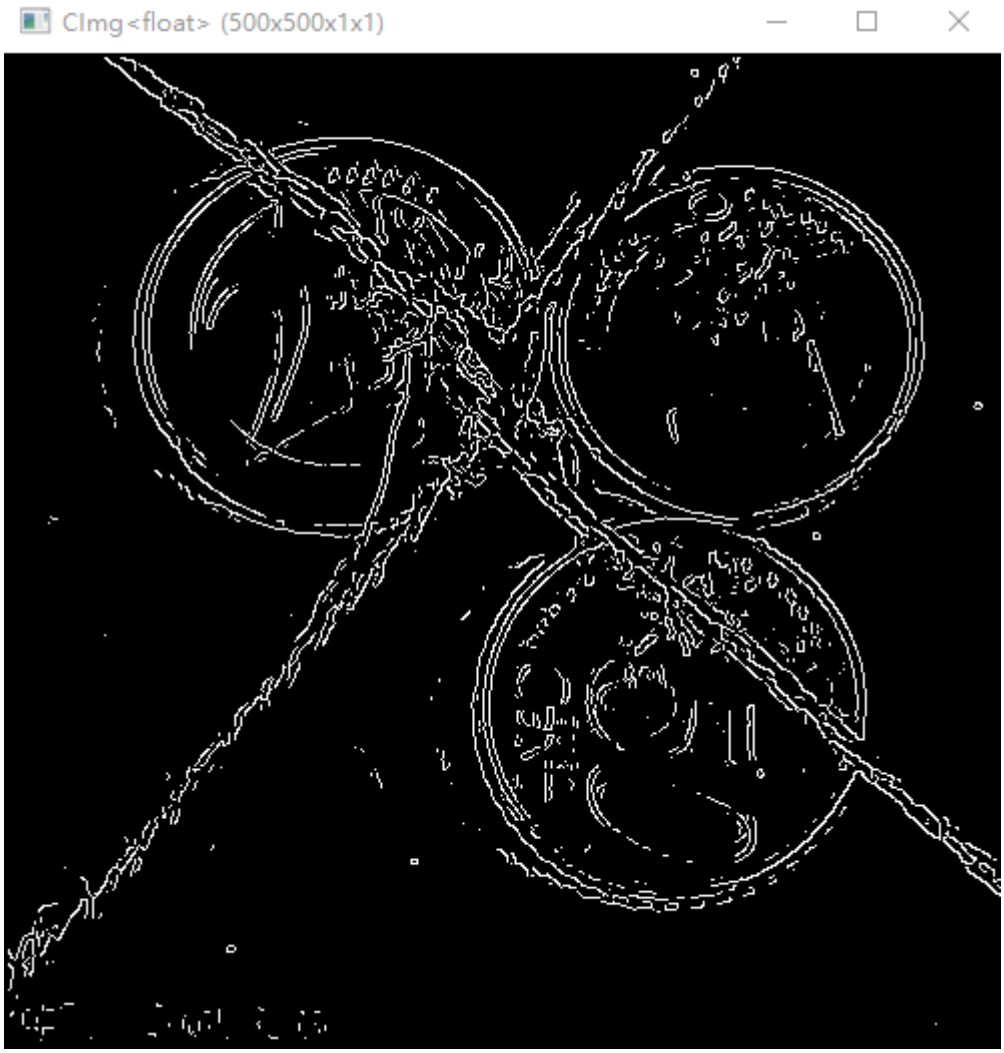

CImg<float> (620x465x1x1)

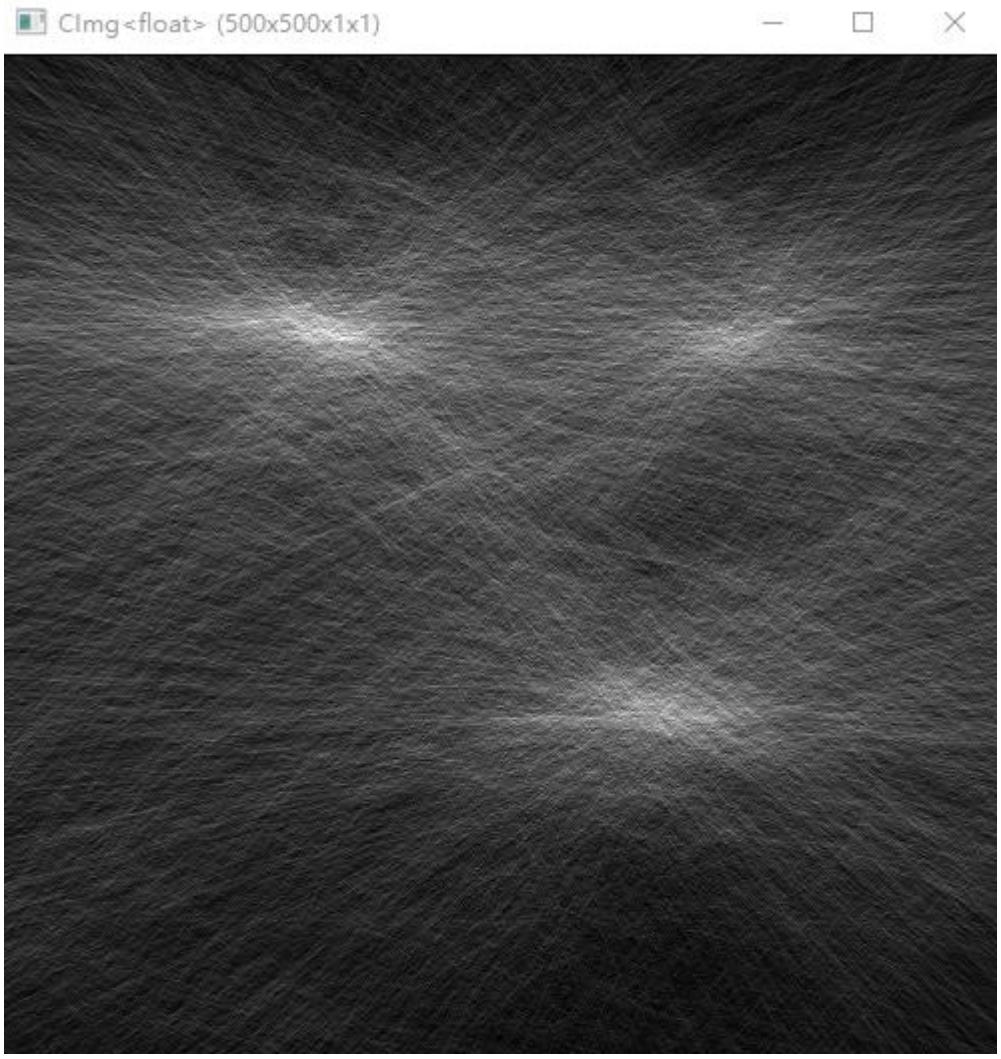

CImg<unsigned char> (620x465x1x3)
1
元
YIYUAN


num: 1
CImg<unsigned cha
BB2C3 (non-shared
, std = 35.508, c
请按任意键继续. .

第二张图：

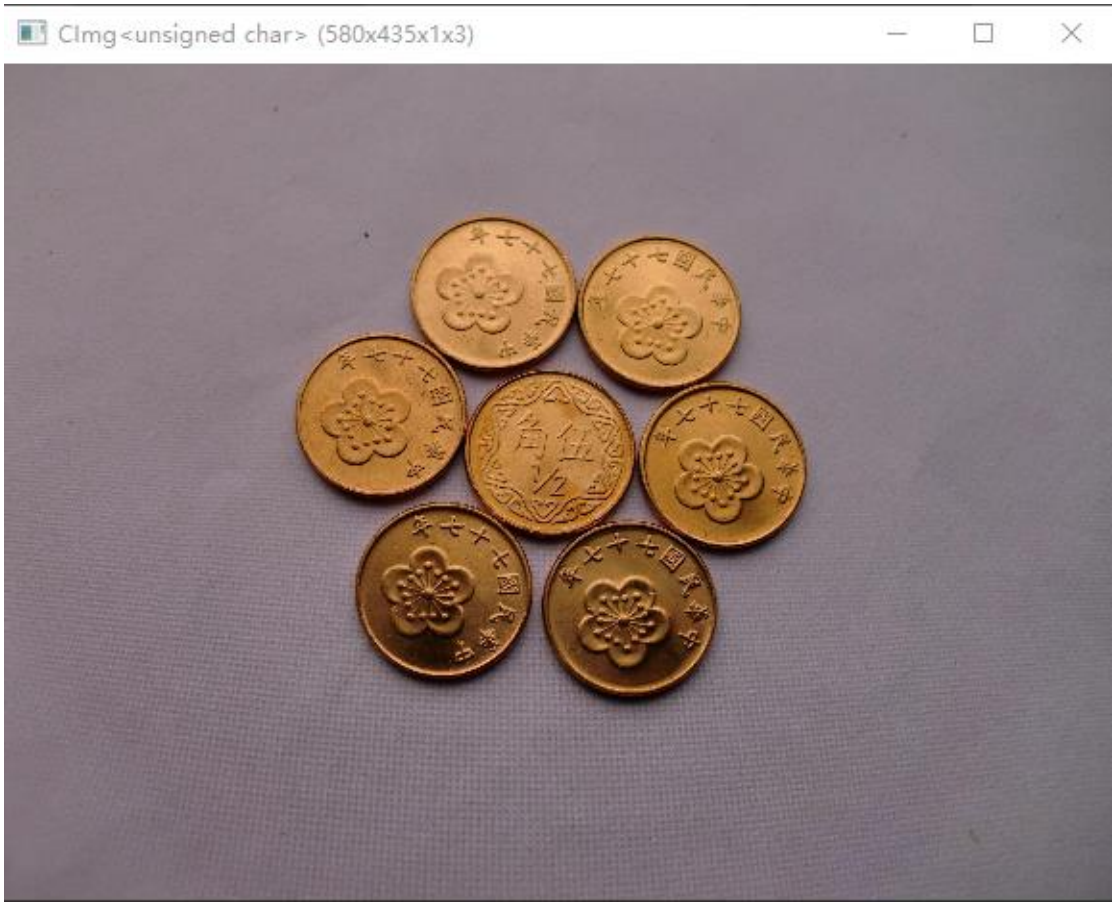

CImg<unsigned char> (500x500x1x3)
中国图库 cntuku.cn


CImg<float> (500x500x1x1)


CImg<float> (500x500x1x1)


```
coords_max = (171,141,0,
num: 3
CImg<unsigned char> (500
6D1EF (non-shared) = [ 2
 168.139, std = 60.763,
请按任意键继续. . .
```

第三张图：


CImg<unsigned char> (580x435x1x3)


CImg<float> (580x435x1x1)

但是此图由于图片较小，参数问题，并没有得到较好的检测，但是可以从霍夫空间看出，圆心处确实是最亮的。