

计算机视觉-Ex3

Ex 3-1

要求

输入：普通 A4 打印纸，上面可能有手写笔记或者打印内容，但是拍照时可能角度不正。

输出：

1. 图像的边缘点(输出边缘图像`Iedge`)
2. 计算 A4 纸边缘的各直线方程, 输出如下结果: (a)各个直线的参数方程; (b) 在上面的边缘图`Iedge`绘制直线, 用蓝色显示, 得到图像`I2`; (c)在`I2` 图上显示 A4 纸的相关边缘点,用红色点显示, 得到图像`I3`
3. 输出 A4 纸的四个角点(在`I3` 上用半径为 5 的圆绘制角点,得到图像`I4`)

程序简述

代码实现中使用到了上一次作业实现的 Canny 进行图像边缘的输出，并添加了一个 A4Detect 类，类的构造需传入原图文件名、霍夫变换后筛选线时的阈值比例、Canny 边缘检测的阈值（为提升边缘检测的速度，双阈值处理时使用两个相同的阈值，即仅执行一遍阈值处理），以及一个代表是否跳过 NMS 的 bool 型变量。类中使用霍夫变换进行 A4 纸边缘的直线检测。

构建一个 A4Detect 对象后调用其 runHough 方法进行 A4 纸边缘检测，该方法将调用 Canny 获取边缘图并保存至 `cannyEdge.bmp` 中, 然后调用 `houghTransform` 方法获取以极坐标形式表示的所有检测到的直线，然后转为笛卡尔坐标系形式的直线，将直线方程输出到程序的标准输出中，最后将与 A4 纸相关的直线段及其交点绘制在原图上并返回绘制后的一个 `CImg` 对象。

在 `houghTransform` 方法中，将首先获取 `houghSpace`，原理是使用一个列为 θ 角，行为 r 的二维矩阵，全部元素初始化为 0。对于每一个有效的边缘点，将过该点的从 0° 到 360° 的所有直线在矩阵中加一，直线极坐标方程中 r 的公式为 $r =$

$\sqrt{(x^2 + y^2)} \times \cos(\theta_0 - \theta)$, 其中 θ_0 为边缘点的角度， θ 为直线的角度。

`houghSpace` 中存储的即为每一条直线的"票数", 接下来首先找出票数最多的一条线，然后乘以输入的阈值比例，使用该结果对所有线进行筛选，再排除较为接近的直线获得最后的 A4 纸边缘的直线。

在原图上标注角点和 A4 纸相关边缘的做法是在找出的直线间找交点作为角点，然后再对每两个交点进行计算，若存在一条在找到的直线集合中的直线，则代表这两交点间的线段是 A4 纸边缘，将其用红色标注出来。

用于测试的主程序的用法如下：

```
./a4detect [-t thresh(default 0.5) cannyThresh(default 50)][filename]
```

-t 选项后接霍夫变换和 canny 边缘检测的阈值，最后一个参数指定读入的文件名。程序完成检测后会将 Canny 输出的边缘图片存入 cannyEdge.bmp 文件，并将输出的标注好角点与边缘的图像进行显示并存入 coins.bmp 文件。

程序编译

程序编译使用 cmake 进行，在 Mac, Linux 以及 Windows 环境下分别在代码目录下使用如下对应的命令进行：

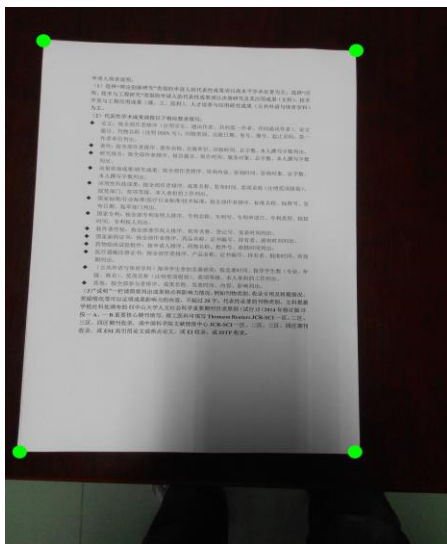
```
#Linux/Mac
cmake .
make
#Windows (需使用 Native Tools Command Line Prompt for Visual Studio)
cmake -G "NMake Makefiles" .
nmake
```

程序测试

使用给定的测试集进行测试，测试的结果图片与终端输出及所使用的程序参数如下：

1. IMG20150320143133.bmp（霍夫变换阈值 0.5，Canny 阈值 80）

```
./a4detect -t 0.5 80 IMG_20150320_143133.bmp
```



```

./a4detect-mac -t 0.5 80 IMG_20150320_143133.bmp
Using 0.5 as threshold for hough transform.
Using 80 as threshold for Canny.
Using 1851 as threshold.
MaxLength: 5239
y=-359.999x+884881
y=-19.0811x+5407.37
y=-6.12323e-17x+3460
y=0.0349208x+253.154
(2448.4,3460)
(2457.07,338.957)
(102.057,3460)
(269.628,262.57)
CImg<unsigned char> (3120x4208x1x3): this = 0x7ffec4fb9388, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x11ec55000..0x1211e4eff (non-shared) = [ 31
21 16 15 16 24 26 16 ... 42 44 21 26 48 45 42 41 ], min = 0, max = 255, mean =
100.244, std = 79.1377, coords_min = (2335,0,0,0), coords_max = (320,264,0,0).

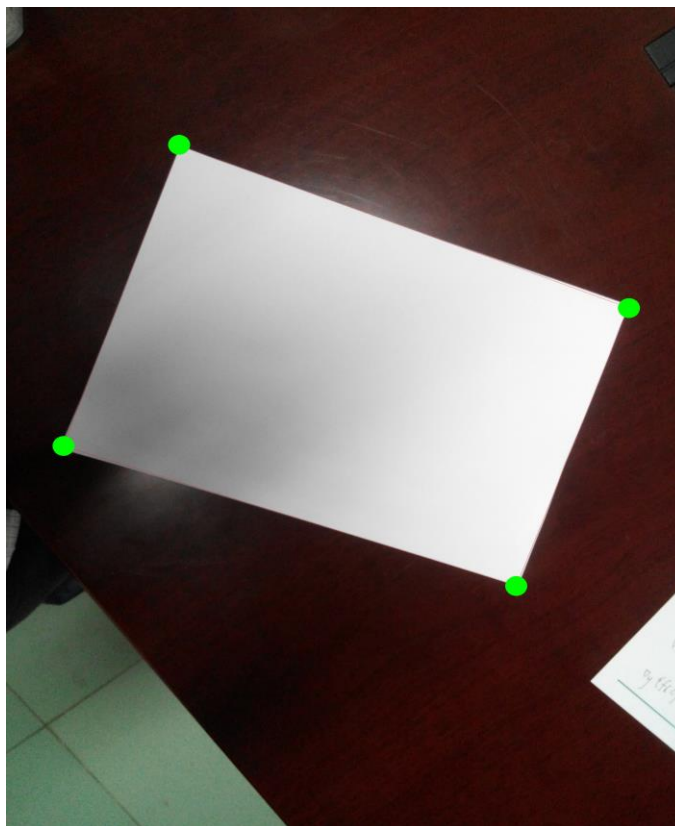
```

A4纸边缘直线方程

A4纸边缘角点

2. IMG20150320143145.bmp (霍夫变换阈值 0.6, Canny 阈值 60)

```
./a4detect -t 0.6 60 IMG_20150320_143145.bmp
```



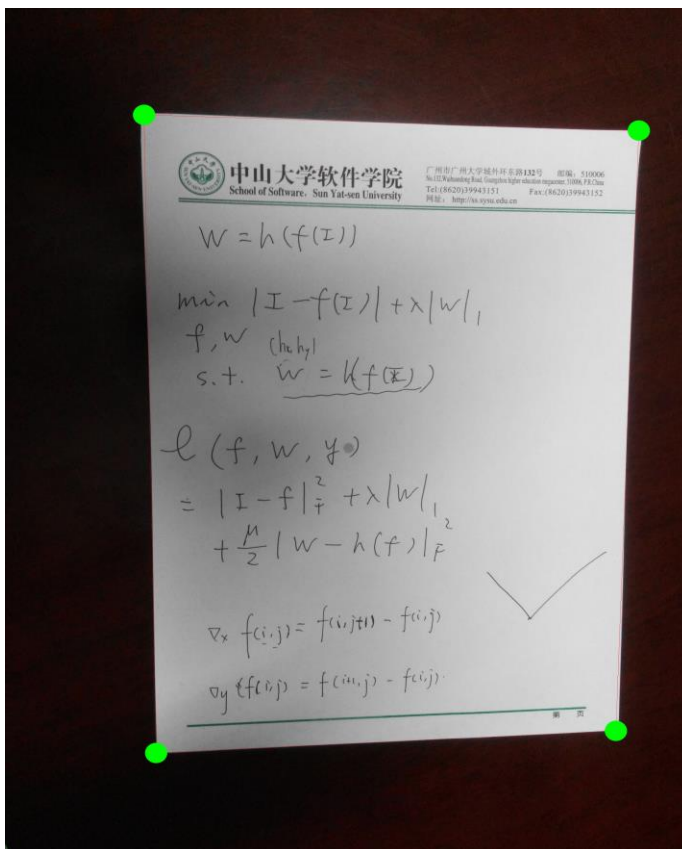
```

./a4detect-mac -t 0.6 60 IMG_20150320_143145.bmp
Using 0.6 as threshold for hough transform.
Using 60 as threshold for Canny.
Using 1840 as threshold.
MaxLength: 5239
y=-2.90421x+3013.19
y=-2.74748x+9402.95
y=0.344328x+2153.32
v=0.404026x+382.88
(264.697,2244.46)
(795.08,704.113)
(2344.79,2960.69)
(2862.15,1539.26)
CImg<unsigned char> (3120x4208x1x3): this = 0x7ffee873f388, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x11cefc000..0x11f48beff (non-shared) = [ 95
93 99 114 108 83 77 101 ... 5 0 0 19 27 13 3 6 ], min = 0, max = 255, mean = 83
.1367, std = 82.883, coords_min = (79,7,0,0), coords_max = (842,723,0,0).

```

3. IMG20150320143211.bmp(霍夫变换阈值 0.6, Canny 阈值 80)

```
./a4detect -t 0.6 80 IMG_20150320_143211.bmp
```



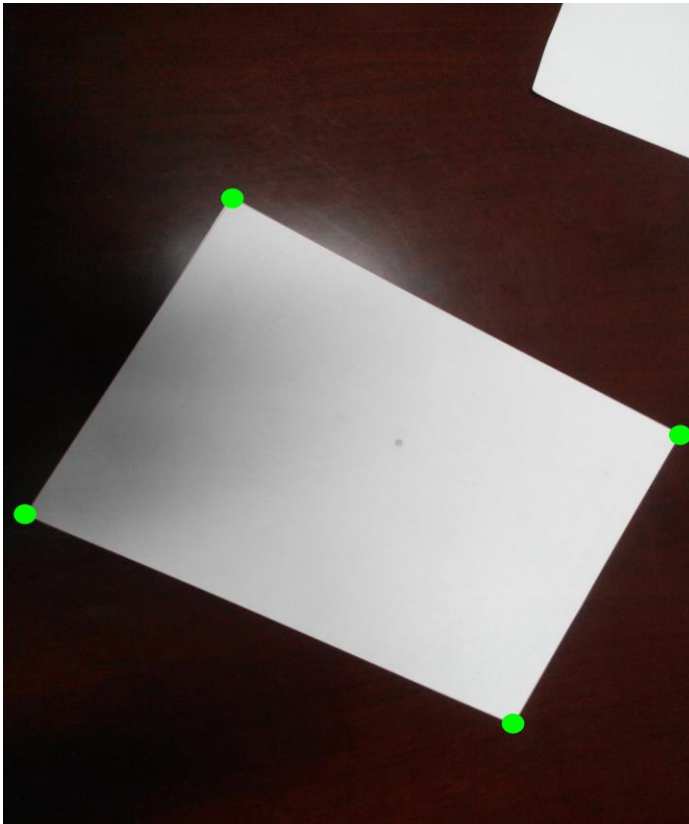
```

./../a4detect-mac -t 0.6 80 IMG_20150320_143211.bmp
Using 0.6 as threshold for hough transform.
Using 80 as threshold for Canny.
Using 2490 as threshold.
MaxLength: 5239
y=-28.6363x+83038.4
y=-0.0524078x+3741.13
y=0.0349208x+509.31
y=57.29x-35639.8
(2774.2,3595.74)
(2878.47,609.829)
(686.768,3705.14)
(631.37,531.358)
Cimg<unsigned char> (3120x4208x1x3): this = 0x7ffee9976388, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x119589000..0x11bb18eff (non-shared) = [ 20
19 16 11 7 5 5 5 ... 0 0 2 5 6 6 3 0 ], min = 0, max = 255, mean = 97.8777, std
= 82.5654, coords_min = (7,321,0,0), coords_max = (682,533,0,0).

```

4. IMG20150320143220.bmp(霍夫变换阈值 0.5, Canny 阈值 60)

```
./a4detect -t 0.5 60 IMG_20150320_143220.bmp
```



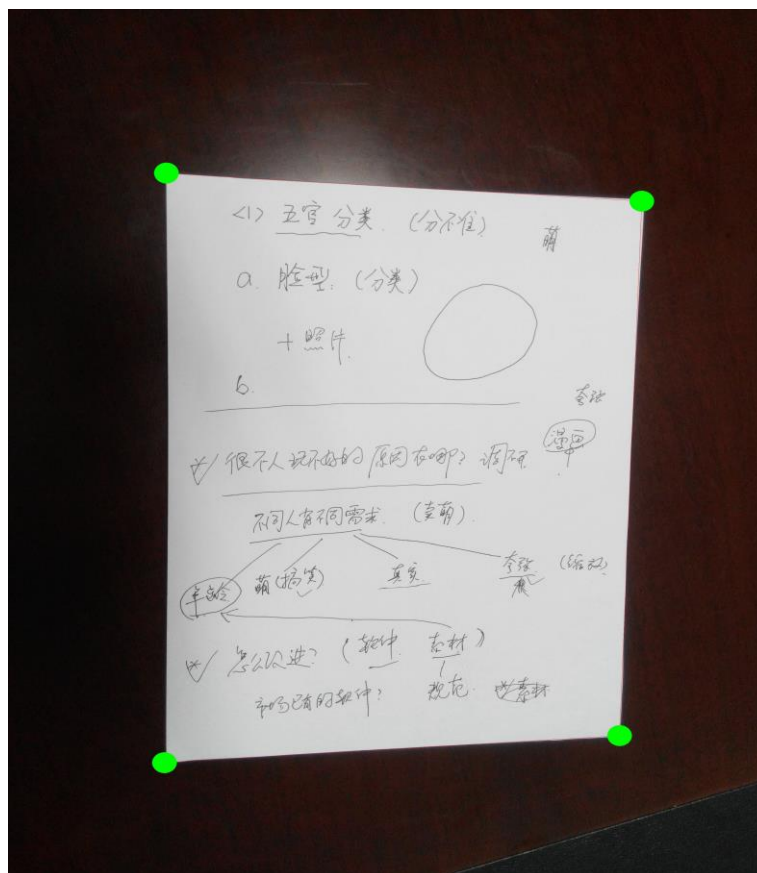

```

./a4detect-mac -t 0.5 60 IMG_20150320_143220.bmp
Using 0.5 as threshold for hough transform.
Using 60 as threshold for Canny.
Using 2357 as threshold.
MaxLength: 5239
y=-1.96261x+8163.17
y=-1.73205x+2778
y=0.487733x+2560.1
y=0.600861x+379.156
(2286.65,3675.37)
(3036.51,2203.68)
(98.1641,2607.97)
(1028.26,996.998)
CImg<unsigned char> (3120x4208x1x3): this = 0x7ffee15fc398, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x1236da000..0x125c69eff (non-shared) = [ 22
17 13 15 16 14 10 10 ... 18 14 8 2 0 1 5 9 ], min = 0, max = 255, mean = 85.819
6, std = 85.6623, coords_min = (1021,946,0,0), coords_max = (1072,1022,0,0).

```

5. IMG20150320143334.bmp(霍夫变换阈值 0.5, Canny 阈值 60)

```
./a4detect -t 0.5 50 IMG_20150320_143334.bmp
```



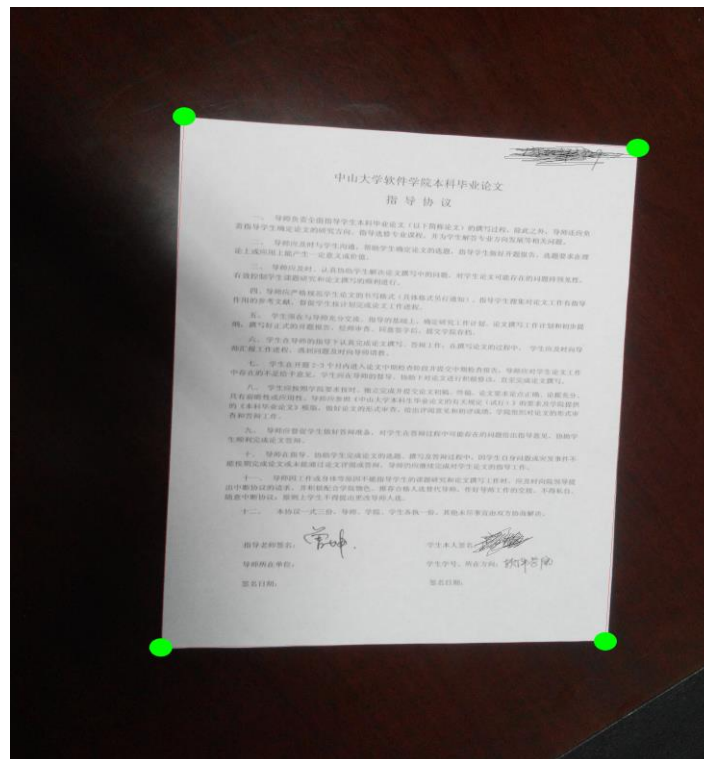
```

./../a4detect-mac -t 0.5 50 IMG_20150320_143334.bmp
Using 0.5 as threshold for hough transform.
Using 50 as threshold for Canny.
Using 2202 as threshold.
MaxLength: 5239
y=-359.999x+234720
y=-28.6363x+75330.6
y=-0.0699268x+3689.99
y=0.0699268x+747.822
(641.877,3645.1)
(649.799,793.26)
(2507.87,3514.62)
(2598.14,929.502)
CImg<unsigned char> (3120x4208x1x3): this = 0x7ffedfd16388, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x1235d5000..0x125b64eff (non-shared) = [ 18
22 23 23 27 32 28 20 ... 24 24 26 26 22 20 21 24 ], min = 0, max = 255, mean =
97.2494, std = 91.8269, coords_min = (783,0,0,0), coords_max = (700,797,0,0).

```

6. IMG20150320143345.bmp(霍夫变换阈值 0.5, Canny 阈值 50)

./a4detect -t 0.5 50 IMG_20150320_143345.bmp



```

./a4detect-mac -t 0.5 50 IMG_20150320_143345.bmp
Using 0.5 as threshold for hough transform.
Using 50 as threshold for Canny.
Using 2101 as threshold.
MaxLength: 5239
y=-28.6363x+22607.8
y=-19.0811x+53653.4
y=-0.0174551x+3567.54
v=0.0874887x+540.055
(665.305,3555.93)
(768.275,607.27)
(2627.29,3521.68)
(2770.85,782.473)
CImg<unsigned char> (3120x4208x1x3): this = 0x7ffee1d2c398, size = (3120,4208,1,
3) [37 Mio], data = (unsigned char*)0x121298000..0x123827eff (non-shared) = [ 30
42 49 30 20 34 31 31 ... 26 25 21 16 15 17 17 15 ], min = 0, max = 255, mean =
99.9703, std = 90.3342, coords_min = (434,1,0,0), coords_max = (819,611,0,0).

```

测试中的参数分析

A4 纸边缘检测程序接收的参数为霍夫变换后筛选直线的阈值和 Canny 检测的阈值，霍夫变换的阈值若太小则可能会存在某一条边未检测出来，太小则可能会出现检测出的边缘过多的情况。

Canny 的阈值则影响了 Canny 边缘检测时检测出的边缘点数目，若设得太小则结果会受到字或其他边缘的干扰，如第三个测试样例，纸张上有一条较长的线，为避免该线被识别为纸张边缘该图的 Canny 阈值需要设得较高。

Ex 3-2

要求

输入：图像上面有若干硬币。

输出：

1. 图像的边缘(输出边缘图像Iedge)
2. 把图像中边缘拟合成圆(在图像Iedge 绘制出对应圆形,用蓝色显示,得到图像I2),圆周相关的像素(在I2 上显示与圆形相关的像素,用红色显示,得到图像I3)
3. 输出图像中硬币的数量

程序简述

硬币识别的程序的主要过程与 A4 纸边缘识别基本相似，包含了一个类 `CoinDetect`，构造时接收的参数为 Canny 的阈值，霍夫变换后筛选圆的阈值比例，输入文件名以及需要查找的圆的最小和最大半径。

构造 `CoinDetect` 对象后调用其 `runHough` 方法进行硬币检测，该方法将先使用 Canny 进行边缘检测，然后调用 `getHoughCircles` 方法进行霍夫变换检测圆。

因为圆可以由圆心和半径进行定义，程序使用了一个三个元素的数组来表示圆。

霍夫变换检测圆的过程是对于输入参数的最小最大半径区间按 5 为步长进行圆的扫描，扫描过程中对于每一个有效边缘点作半径为当前扫描半径的圆，在票数表中将所有圆心为刚作的圆上的点，半径为扫描半径的圆票数加一。

扫描后找出最高的票数与阈值比例相乘获得筛选的阈值，然后在高于筛选阈值的圆中排除相似的圆，剩下的圆即为求得的结果，圆的数量就是图中硬币的数量，程序将会在标准输出中输出检测出的硬币数目。

对于找到的所有圆，将在原图上画出，为了使画出的圆在图上更为明显，会画出圆半径 ± 5 范围内的所有圆，增加画出的圆的线条厚度。

用于测试的主程序的用法如下：

```
./coin-detect [-t thresh(default 0.5) cannyThresh(default 50)]  
filename low_r high_r
```

参数中文件名和 `r` 的大小值是必须指定的，程序完成检测后将会把 Canny 检测出的边缘存入 `cannyEdges.bmp`，标记完圆后的图片存入 `coins.bmp`。

程序编译方式与 A4 纸检测的程序编译方式相同。

程序测试

使用给出的测试集进行测试，使用的参数和测试结果图片与终端输出如下：

1. 2.bmp(Canny 阈值 70，大小半径为 150, 160)

```
./coin-detect -t 0.7 60 2.bmp 150 160
```



```
er ● ./../coin-detect-mac -t 0.7 60 2.bmp 150 160
Using 0.7 as threshold for hough transform.
Using 60 as threshold for Canny.
Processing image: 2.bmp
Width: 620
Height: 465
Checking r from 150 to 160
100% voting completed.
Hough threshold is 32
Processing 254090 Voted circles.
100% processed.
317 223 155
Total coins: 1
CImg<unsigned char> (620x465x1x3): this = 0x7ffee1ecd340, size = (620,465,1,3) [
844 Kio], data = (unsigned char*)0x10e05e000..0x10e131283 (non-shared) = [ 79 79
80 80 82 86 93 98 ... 117 121 121 115 110 108 108 108 ], min = 0, max = 255, me
an = 126.28, std = 40.9438, coords_min = (305,63,0,1), coords_max = (305,63,0,0)
```

2. 5.bmp(Canny 阈值 80, 大小半径 200, 225)

```
./coin-detect -t 0.7 80 5.bmp 200 225
```



昵图网 www.nipic.com 877,9522

W0:20100710003515565364

```
er • ./../coin-detect-mac -t 0.7 80 5.bmp 200 225
Using 0.7 as threshold for hough transform.
Using 80 as threshold for Canny.
Processing image: 5.bmp
Width: 982
Height: 1024
Checking r from 200 to 225
100% voting completed.
Hough threshold is 38
Processing 3774644 Voted circles.
100% peocessed.ocessed.
253 771 225
733 712 200
270 273 225
742 286 200
Total coins: 4
CImg<unsigned char> (982x1024x1x3): this = 0x7ffee42ed340, size = (982,1024,1,3)
[2946 Kio], data = (unsigned char*)0x10bc34000..0x10bf147ff (non-shared) = [ 24
2 241 241 241 240 240 240 241 ... 255 255 255 255 255 255 255 255 ], min = 0, ma
x = 255, mean = 157.341, std = 81.1165, coords_min = (361,61,0,0), coords_max =
(505,19,0,0).
```


3. 6.bmp(Canny 阈值 50, 大小半径 150, 180)

```
./coin-detect -t 0.7 50 6.bmp 150 180
```



```
er • ./../coin-detect-mac -t 0.7 50 6.bmp 150 180
Using 0.7 as threshold for hough transform.
Using 50 as threshold for Canny.
Processing image: 6.bmp
Width: 2320
Height: 1740
Checking r from 150 to 180
100% voting completed.
Hough threshold is 73
Processing 8854632 Voted circles.
100% processed.
1019 496 170
1305 1131 175
919 1094 175
788 741 170
1362 522 165
1136 810 165
1507 835 165
Total coins: 7
CImg<unsigned char> (2320x1740x1x3): this = 0x7ffee49d3340, size = (2320,1740,1
3) [11 MiB], data = (unsigned char*)0x10b557000..0x10c0e3a3f (non-shared) = [ 1
4 134 134 135 135 136 136 136 ... 94 94 94 94 94 94 94 94 ], min = 0, max = 255
mean = 122.426, std = 32.6721, coords_min = (1204,974,0,0), coords_max = (1006
321,0,0).
```

4. 11.bmp(Canny 阈值 40, 大小半径 175, 200)

`./coin-detect -t 0.7 40 11.bmp 175 200`



```
er ● ./../coin-detect-mac -t 0.7 40 11.bmp 175 200
Using 0.7 as threshold for hough transform.
Using 40 as threshold for Canny.
Processing image: 11.bmp
Width: 1000
Height: 1000
Checking r from 175 to 200
100% voting completed.
Hough threshold is 56
Processing 5005932 Voted circles.
100% peocessed.ocessed.
664 655 195
325 285 190
729 276 185
Total coins: 3
CImg<unsigned char> (1000x1000x1x3): this = 0x7ffee0ef8340, size = (1000,1000,1,
3) [2929 KiB], data = (unsigned char*)0x10f033000..0x10f30f6bf (non-shared) = [
227 223 220 222 227 228 224 219 ... 142 137 134 136 143 149 149 147 ], min = 0,
max = 255, mean = 166.014, std = 64.2018, coords_min = (716,86,0,1), coords_max
= (129,0,0,0).
```


5. 12.bmp(Canny 阈值 50, 大小半径 470,520)

```
./coin-detect -t 0.7 50 12.bmp 470 520
```



```
er ● ./../coin-detect-mac -t 0.7 50 12.bmp 470 520
Using 0.7 as threshold for hough transform.
Using 50 as threshold for Canny.
Processing image: 12.bmp
Width: 2592
Height: 1944
Checking r from 470 to 520
100% voting completed.
Hough threshold is 34
Processing 27597633 Voted circles.
100% processed.
1936 1050 510
932 1035 470
Total coins: 2
CImg<unsigned char> (2592x1944x1x3): this = 0x7ffee5331340, size = (2592,1944,1,
3) [14 Mio], data = (unsigned char*)0x10abf2000..0x10ba5c8ff (non-shared) = [ 16
0 161 162 164 167 169 170 171 ... 157 167 166 141 170 144 152 164 ], min = 0, ma
x = 255, mean = 141.064, std = 39.6085, coords_min = (1432,1328,0,0), coords_max
= (991,496,0,0).
```

6. f55bcad35f5d993c80f928f3dc940aa8.bmp(Canny 阈值 50, 大小半径 40,70)
./coin-detect -t 0.7 50 f55bcad35f5d993c80f928f3dc940aa8.bmp 40 70



```
er ● ./../coin-detect-mac -t 0.7 50 f55bcad35f5d993c80f928f3dc940aa8.bmp 40 70

Using 0.7 as threshold for hough transform.
Using 50 as threshold for Canny.
Processing image: f55bcad35f5d993c80f928f3dc940aa8.bmp
Width: 400
Height: 300
Checking r from 40 to 70
100% voting completed.
Hough threshold is 122
Processing 793773 Voted circles.
100% processed.
77 220 60
81 98 60
202 97 60
188 213 45
285 196 40
Total coins: 5
CImg<unsigned char> (400x300x1x3): this = 0x7ffeef8ae320, size = (400,300,1,3) [
351 Kio], data = (unsigned char*)0x10067e000..0x1006d5e3f (non-shared) = [ 253 2
53 253 253 253 253 253 253 ... 255 255 255 255 255 255 255 255 ], min = 0, max =
255, mean = 134.348, std = 79.9898, coords_min = (334,54,0,0), coords_max = (32
,0,0,0).
```

测试输入参数分析

测试程序接收的参数为霍夫变换后筛选圆形的阈值比例，Canny 的阈值以及搜索圆形时半径的范围。其作用分别如下：

1. 圆形的阈值比例作用是在投票后根据最高的票数乘以该比例作为阈值排除票数较低的圆，在本次的测试样例中该参数均被设为 **0.7**，原因是本次的测试样例的圆大小差异较小，所获得票数的差异也因此不大。当出现图中圆大小差异较大的情况时，阈值比例需要对应降低以避免较小的圆被排除。
2. Canny 的阈值用于排除梯度值较小的边缘，过低时检测的结果可能会受到其他边缘点的干扰，太高时部分的圆的边缘可能会有不完整的情况，导致检测结果中可能会漏掉应被检测出的圆。
3. 检测时圆的半径范围并不是检测圆形的程序正确运行所必须的参数，但若不指定范围程序将需要检测非常大的一个半径范围，对于程序运行时间影响较大，因此在此次程序中选择了对于输入的图片需要指定半径范围的做法。