


---

# 第一次课程作业

Lu Peng

School of Computer Science,  
Beijing University of Posts and Telecommunications

A solid red horizontal bar at the bottom of the slide.

# 钱币检测

---

**任务：**编写一个钱币定位系统，其不仅能够检测出输入图像中各个钱币的边缘，同时，还能给出各个钱币的圆心坐标与半径。

## 代码撰写说明：

- 代码可以使用C++或者python语言进行编写，推荐使用python；
- 可直接调用Opencv的Canny与HoughCircle算法完成系统设计；
- 推荐自主实现Canny与Hough算法，此为加分项最高可加20分。

## 设计文档撰写说明：

介绍算法整体流程，各个函数的功能说明，函数的输入参数说明，给出最终拟合结果图，分析各个参数对于最终定位结果的影响。

# 算法流程

---

## 步骤1：使用Canny 算法提取图像边缘

- ① 使用高斯滤波器滤波；
- ② 计算图像的梯度图并获得梯度方向
- ③ 对梯度图进行非极大化抑制
- ④ 使用双阈值法获得最终的边缘图。

## 步骤2：在边缘图上利用Hough变换计算圆心与半径。

- ① 建立参数空间；
- ② 依据边缘点的梯度方向对参数空间进行投票；
- ③ 依据预设定的投票阈值筛选出初步结果；
- ④ 对已筛选出的结果进行非极大化抑制，得到精确的参数（圆心和半径）。

# 样例代码说明 (邱吉撰写)

**环境：**使用python3环境，安装opencv库。

**使用：**运行main.py 即可在对应的result目录下看到如下结果。

```
Circle core: (324.000000, 309.000000) Radius: 63.000000  
Circle core: (467.400000, 900.600000) Radius: 60.600000  
Circle core: (775.500000, 607.500000) Radius: 76.500000  
Circle core: (1071.000000, 960.000000) Radius: 63.000000  
Circle core: (1239.000000, 321.000000) Radius: 63.000000  
Circle core: (1449.000000, 669.000000) Radius: 63.000000
```

Canny result



Hough result

