

---

# 题目 图像分割之边缘检测和连接

姓名 吴紫航      学号 171860659      邮箱 [401986905@qq.com](mailto:401986905@qq.com)      联系方式 call: 18956668797

(南京大学 计算机科学与技术系, 南京 210093)

## 1 实现细节

本节分四个部分讲解实现细节, 对应本次作业中: 边缘检测的三个方法和边缘连接的方法

### 1.1 Sobel基本边缘检测法

#### 实现步骤

- (1) 用高斯滤波器进行平滑
- (2) 用 Sobel 算子计算两个方向的梯度 $|g_x|$ 和 $|g_y|$
- (3) 合并得到 $|g_x|+|g_y|$
- (4) 阈值化, 保留超过最大值 30%的像素, 设置为 255, 其余像素置 0

#### 遇到的问题

关于阈值的选取需要人为的去调整, 过低的阈值难以去噪声点, 过高的阈值又使得边缘大量间断, 最终选取阈值为 30%; 此外, 在阈值化的同时, 进行二值化是必要的, 方便后续的边缘连接的处理。

### 1.2 Marr-Hildreth边缘检测器

#### 实现步骤

- (1) 用高斯低通滤波器平滑图像
- (2) 计算上一步结果的拉普拉斯
- (3) 寻找上述结果的零交叉

#### 遇到的问题

关于零交叉的计算, 原本按课件上的写法, 要求上下、左右、两对角有符号相反差异大于阈值为零交叉, 但是效果并不理想, 边缘毛刺较多。后参考了网上一篇博客, 具体链接参见报告末尾, 改进了零交叉的算法: 先将  $3 \times 3$  的区域通过  $2 \times 2$  的全 1 的滤波器卷积, 生成  $2 \times 2$  的区域, 再判断区域内的最大和最小值是否符号相反且大于阈值。

### 1.3 Canny边缘检测器

#### 实现步骤

- (1) 用高斯滤波器进行平滑
- (2) 用 Sobel 算子计算两个方向的梯度 $g_x$ 和 $g_y$

(3) 进一步得到梯度大小  $M(x, y) = \sqrt{g_x^2 + g_y^2}$  和梯度方向  $\alpha(x, y) = \cot\left(\frac{g_y}{g_x}\right)$

(4) 非最大抑制，即寻找和梯度方向最接近的方向，若当前像素不是方向上最大值则抑制

(5) 滞后阈值，即用两个阈值把像素集合划分为 2 个部分

(6) 连通性分析，即将低阈值集合进行去噪，跟高阈值集合没连通的点都去掉，然后合并两个集合

#### 遇到的问题

非最大抑制按课件的写法是把非最大的像素置零，从实际代码的效率考虑，反过来操作，因为图像是稀疏的，故先整体置零，如果像素是最大的，像素值就设置为 255

## 1.4 边缘连接方法

#### 实现步骤

(1) 给定二值图像和起始点

(2) 如果起始点像素为 0 则直接结束，否则把起始点加入 output 中

(3) 对于起始点的 8 邻域，如果附近存在没访问过的点，且点的像素值为 255，则加入到 output 当中

(4) 以刚刚加入的点为新的起始点，循环的按照第 3 步的方法去寻找。

(5) 当 8 邻域内没符合条件的点时，停止循环。输出 output

#### 遇到的问题

(1) 原本打算在判断局部相似性的时候使用梯度，但效果不理想，于是简单处理，即直接找 8 邻域内像素值不为 0 的点即可，当然寻找的时候需要避免重复，本质思想就是图遍历；

(2) 考虑边界情况要特殊处理，例如整个图像的边和角上的点不能直接用 8 连通性，否则数组越界；

(3) 需要注意坐标参数传递的顺序，imtool 函数的参数顺序是先列(x)再行(y)。而我们实现的 my\_edgeling 参数传递顺序是先行(row)再列(col)。

(4) 起始点在加入 output 后，不将其标记为已到达，目的是方便将来再次到达使得边界成为闭合曲线

(5) 边缘检测的方法采用效果相对最好的 Canny 检测器，在此基础上继续边界跟踪

## 2 结果

本节分三个部分，分别介绍实验的结构设置、实验效果、结论分析

### 2.1 实验设置

(1) code 文件内对应框架代码 edge\_test.m、my\_edge.m、my\_edgeling.m。

其中 my\_edge 函数的参数 method 为 1、2、3 时对应 sobel、Marr-Hildreth、Canny 三种方法

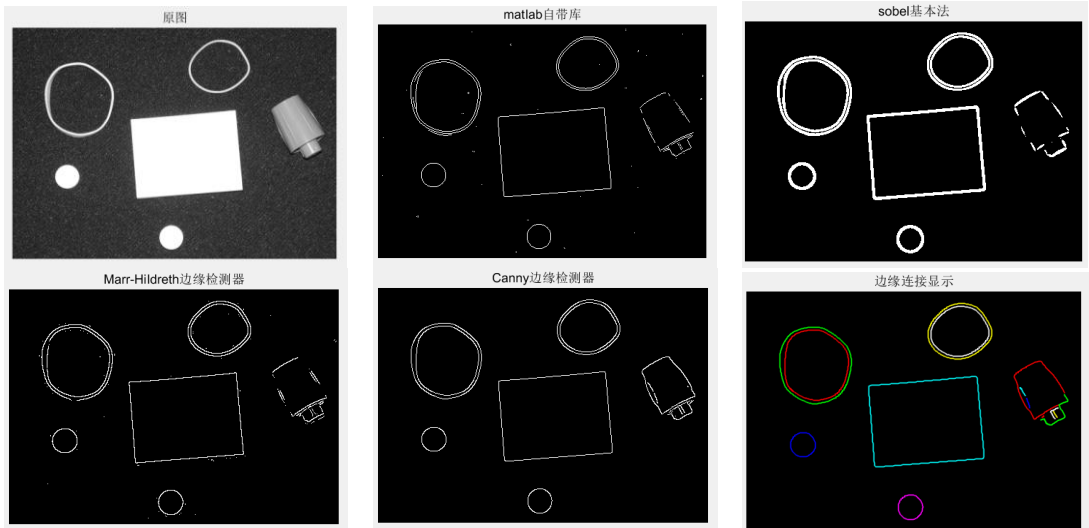
(2) asset\image 文件夹内对应图片文件

直接运行代码即可查看原图、各种边缘检测的效果和边缘连接的效果，要替换输入的图片即把代码中的 imread 函数的参数修改即可。

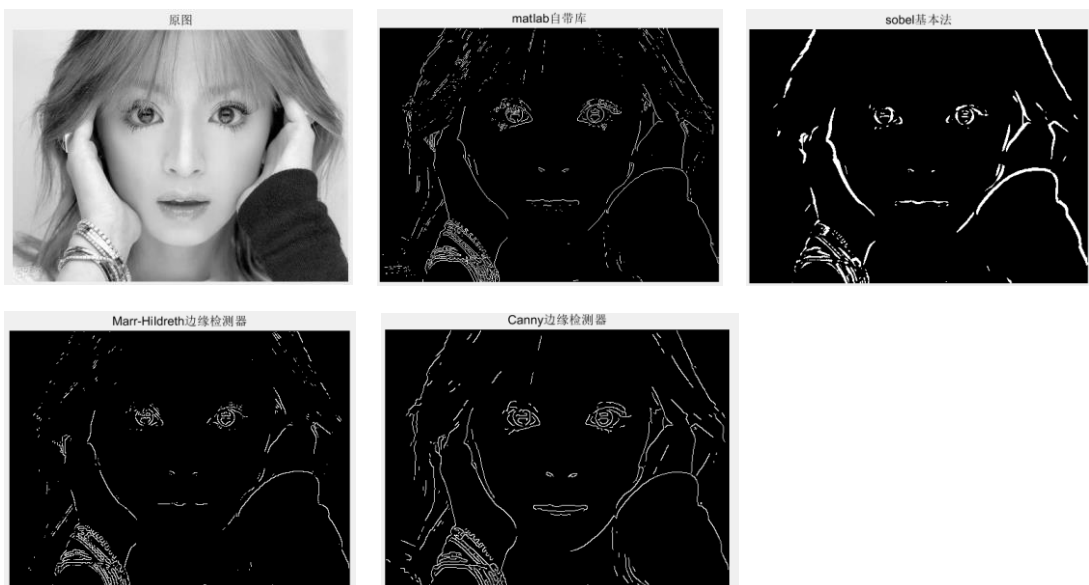
(3) doc 文件夹里即是正在阅读的报告

## 2.2 实验结果

### 2.2.1 rubberband\_cap.png

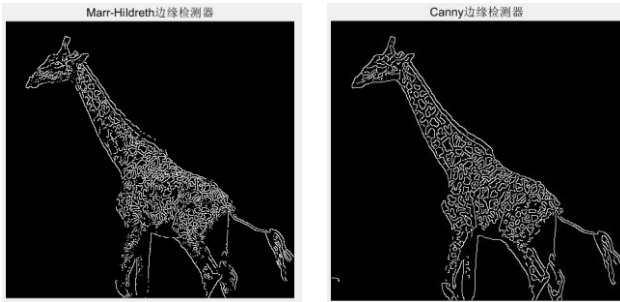


### 2.2.2 ayu.jpg



### 2.2.3 giraffe.jpg

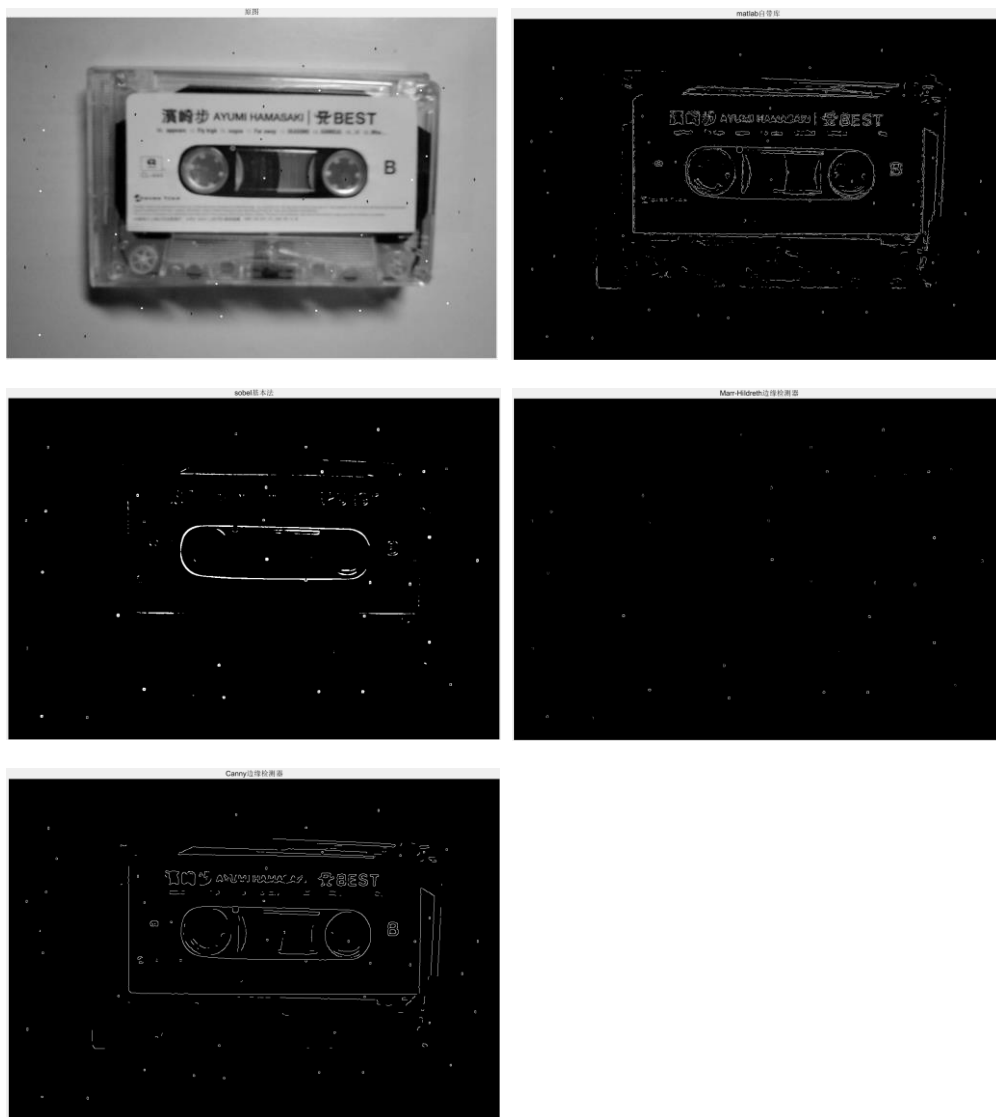




## 2.2.4 leaf.jpg



### 2.2.5 noise.jpg



### 2.3 结果分析

本实验实现的三种边缘检测方法在代码效率和实现效果来看各不相同。分析结果总结如下表

| 方法 | Sobel            | Marr-Hildreth                         | Canny   |
|----|------------------|---------------------------------------|---|
| 特点 | 思路简单，但检测出的边缘比较粗糙 | 计算方便，边界较细，但存在一些间断点。对于噪声严重的图片较难识别出清晰边界 | 效果最好，边界很细，且没有伪边界。边缘的连续性较好，在噪声严重的图片中也能检测出较为理想的边缘 |

---

### 3 参考资料

[1] matlab 练习程序(Marr-Hildreth 边缘检测)

参考了代码中零交叉计算的方法，对课件原本的计算方法进行了改进

<https://www.cnblogs.com/tiandsp/archive/2012/12/14/2818054.html>

[2] matlab plot 画线的颜色设定

参考了 plot 画图的不同颜色的曲线写法

<https://www.cnblogs.com/takeaction/p/3789871.html>