计算机视觉实验 Ex 文档

梁俊华 数据科学与计算机学院 16340129

一、实验要求

1）学号尾数为 9，改写 Code0 的源代码.

2.）封装要求：（1）所有的图像读写、数据处理只能 CImg 库（整个工程文件不允许使用 Openvc 之类的第三方库）；（2）代码封装要求函数接口简洁清晰，可参考 Code2 的方式封装.

3）在原来的代码基础上，增加一个函数：首先把相邻的边缘连成长的线条，并删除长度小于 20 的 Edge.

4）对算法的若干组参数，对所有测试图像进行测试，并分析各参数对结果的影响.

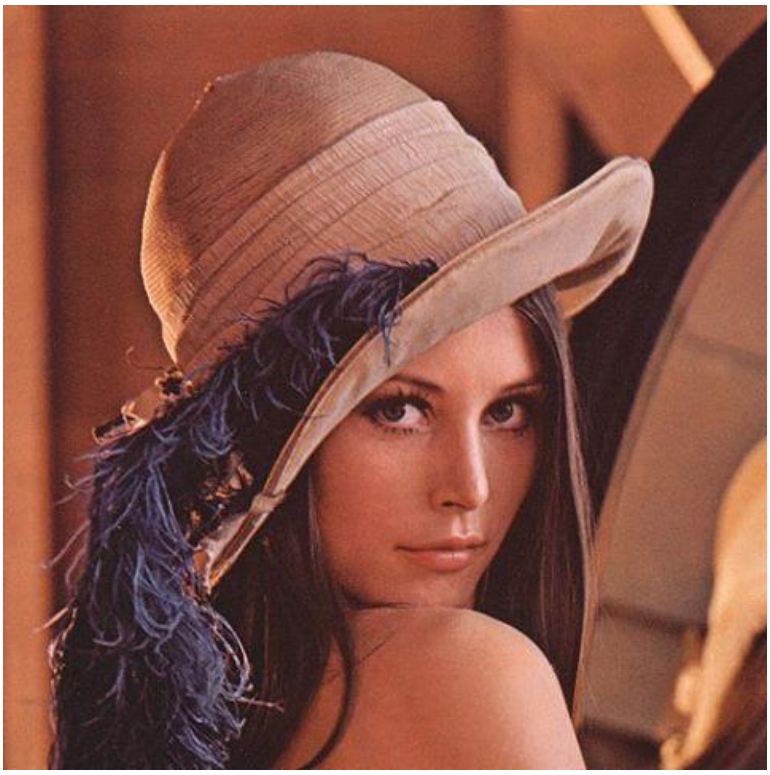
二、测试环境

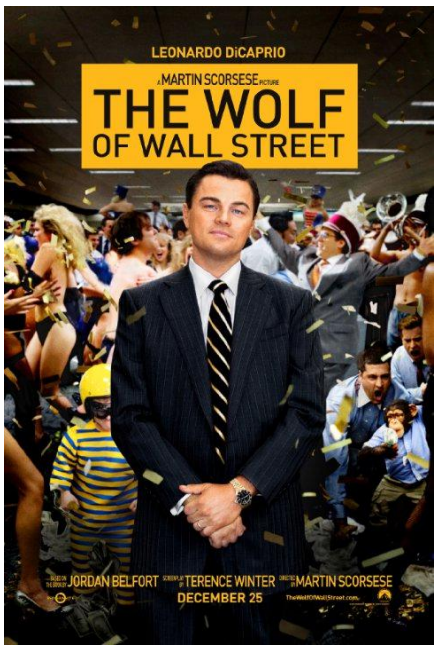
实验二测试的操作系统为 Windows10，使用配置好的 MinGW64 进行编译，进入文件所在的目录后，编译的指令如下图所示：

```
C:\Users\Alva\Desktop\ComputerVision\Ex2. Canny\Code0
λ g++ canny_source.h -std=c++11 -O2 -lgdi32 -c

C:\Users\Alva\Desktop\ComputerVision\Ex2. Canny\Code0
λ g++ canny_source.cpp -std=c++11 -O2 -lgdi32

C:\Users\Alva\Desktop\ComputerVision\Ex2. Canny\Code0
λ a.exe 1.bmp 1.0 0.3 0.7 50
```

三、测试数据

测试数据是老师提供的 4 张图片，主要的测试则是用老师提供的 4 张图片，转换为.bmp 格式后进行测试。四张.bmp 的测试图片如下所示：





四、测试说明

本次实验中，编译出可执行文件后，需要在编译时输入五个参数，其中五个参数的具体说明如下：

1) image = 需要进行图像处理的图像，需要是 bmp 格式或者 pgm 格式.

2) sigma = 高斯模糊的标准差.

3) tlow = 在滞后过程中用于计算阈值下界的因子，范围是（0-1）.

4) thigh = 在滞后过程中用于计算阈值上界的因子，范围是（0-1）.

5) len = 在最后过程中需要删除的边的长度.

五、实验原理

本实验的实验原理是 canny 边缘检测算法，canny 边缘检测算法的具体步骤有以下五个部分：

1) 进行高斯模糊处理.

2) 计算图像的一阶偏导 dx dy.

3) 计算梯度强度.

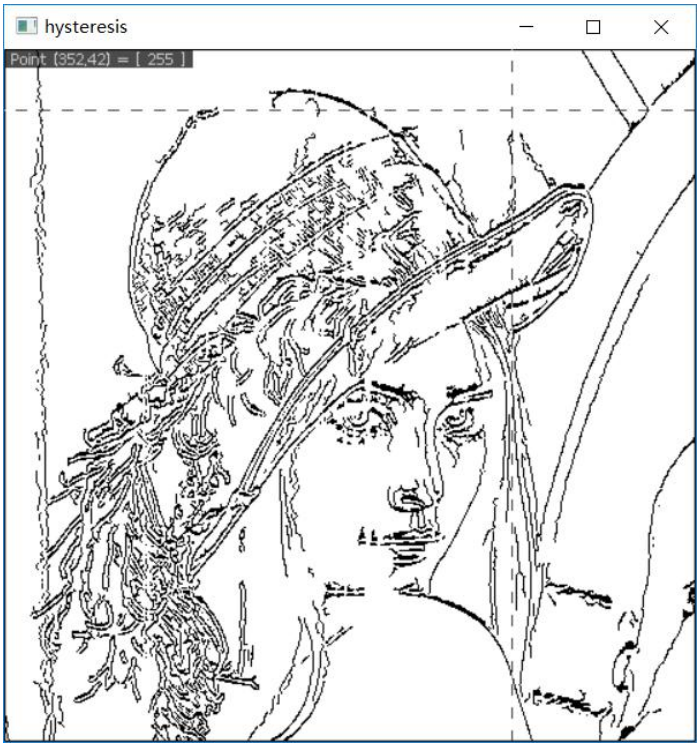
4) 进行非极大化抑制处理.

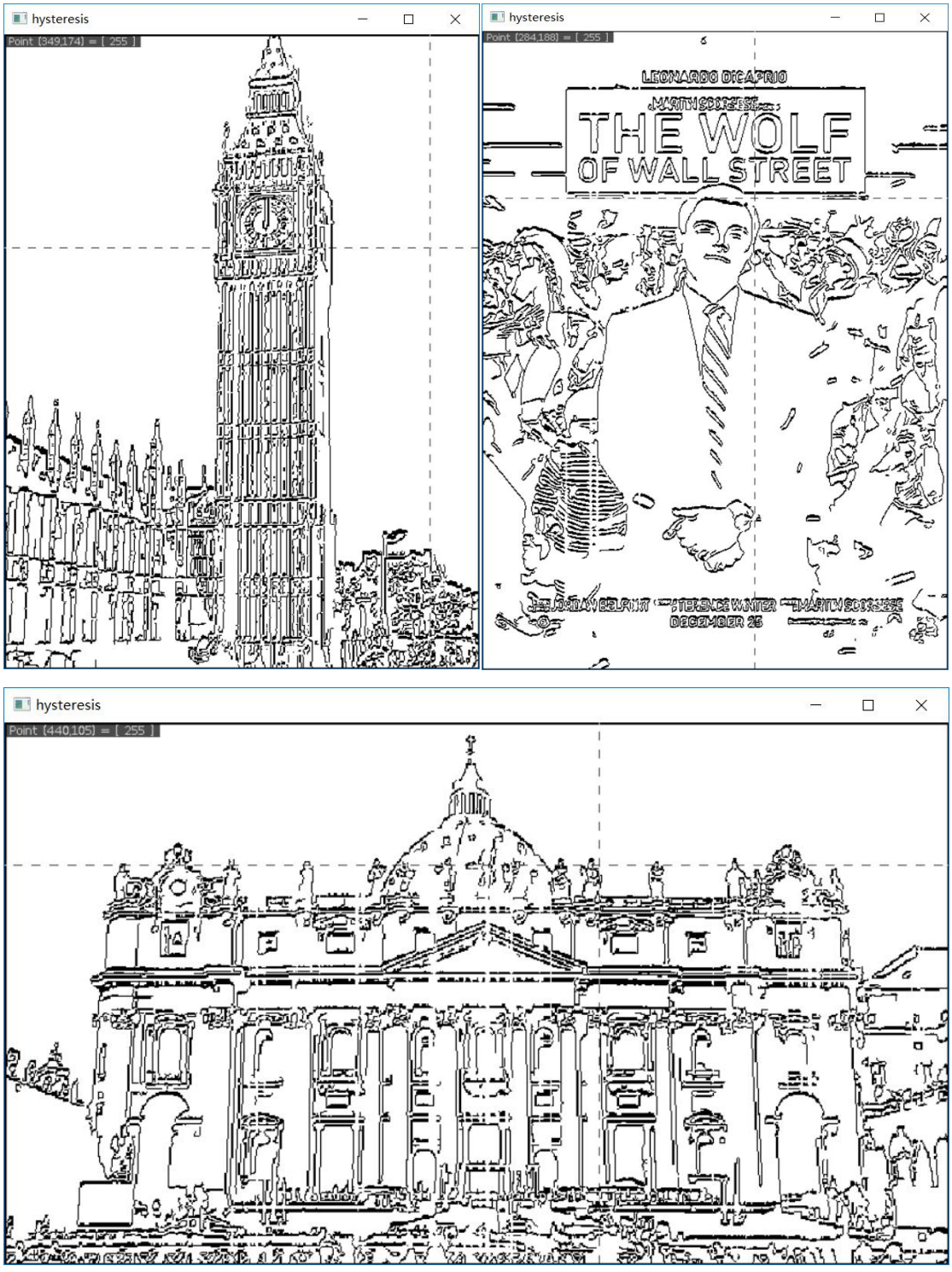
5) 进行滞后阈值处理.

问题四的算法主要参考网站[7]所提供的思路，也就是找出可能是端点的点，然后搜索其十六邻域的中为 EDGE 的像素点，将该点与中心点之间的点都设置为 EDGE 从而实现连接。删除边长小于 20 的边主要用到的算法是 BFS 算法，用 BFS 算法求出连通分量里面，像素点小于 20 的连通分量，然后将这个连通分量设置为 0.

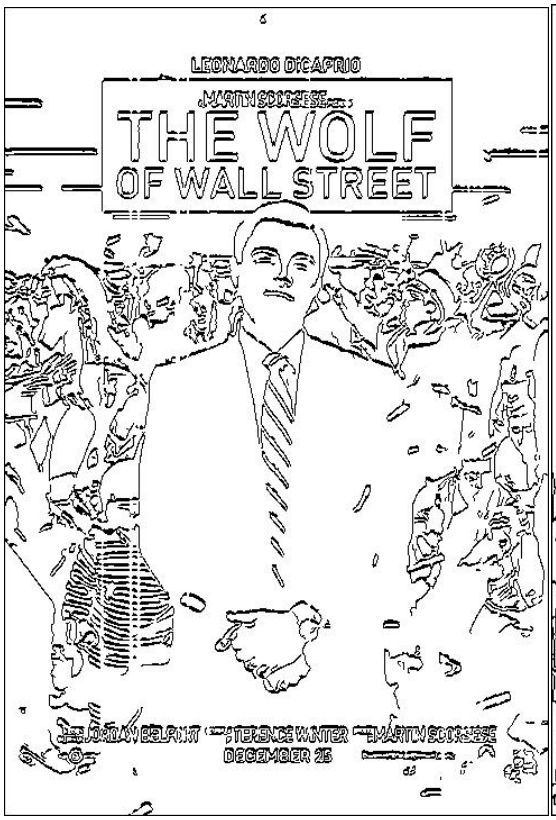
六、实验结果

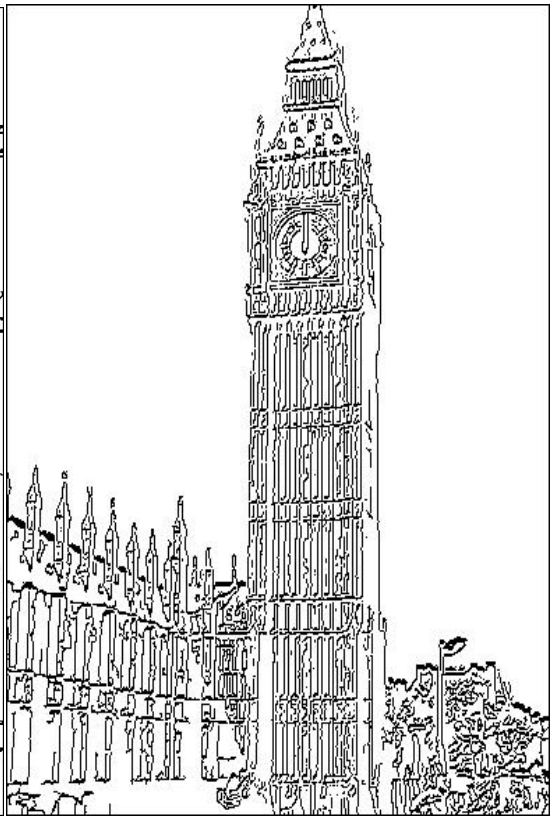
四张 bmp 图片在调用 canny 算法后的实验结果如下所示（其中 sigma = 1.0, tlow = 0.3, thigh = 0.7, len = 20）：

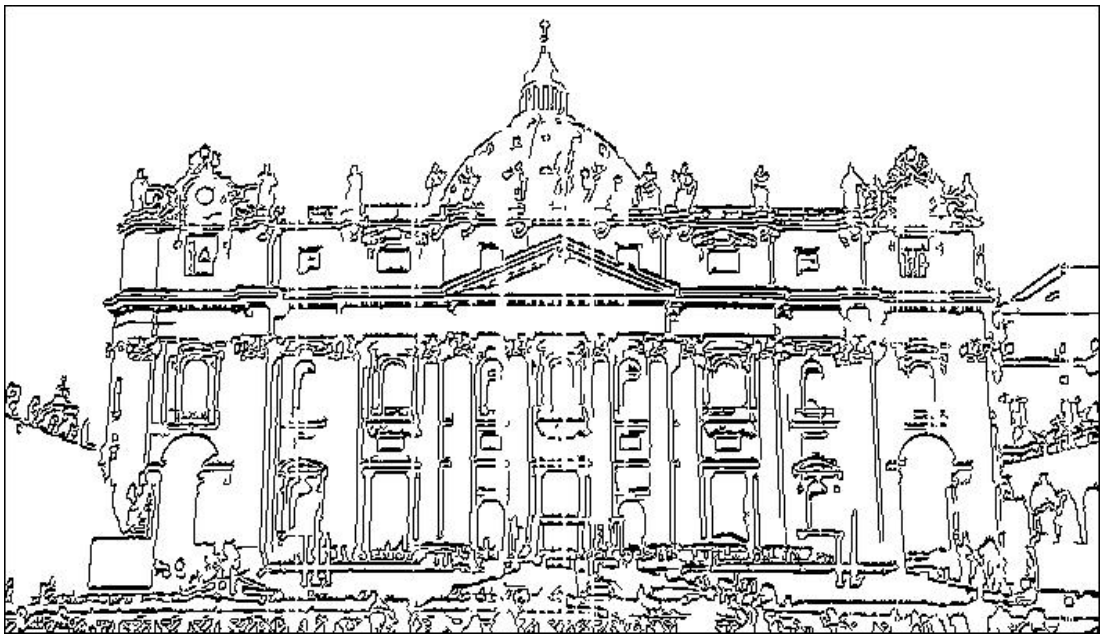




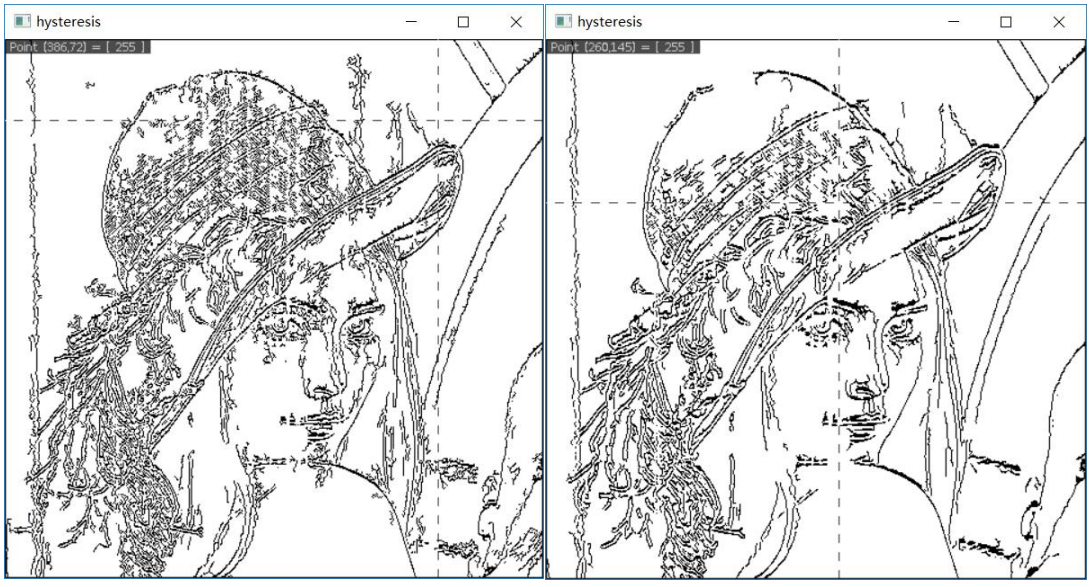
四张 bmp 图片再调用连接删除后的结果如下图所示（为突出效果，这里的 len 改为 50）：

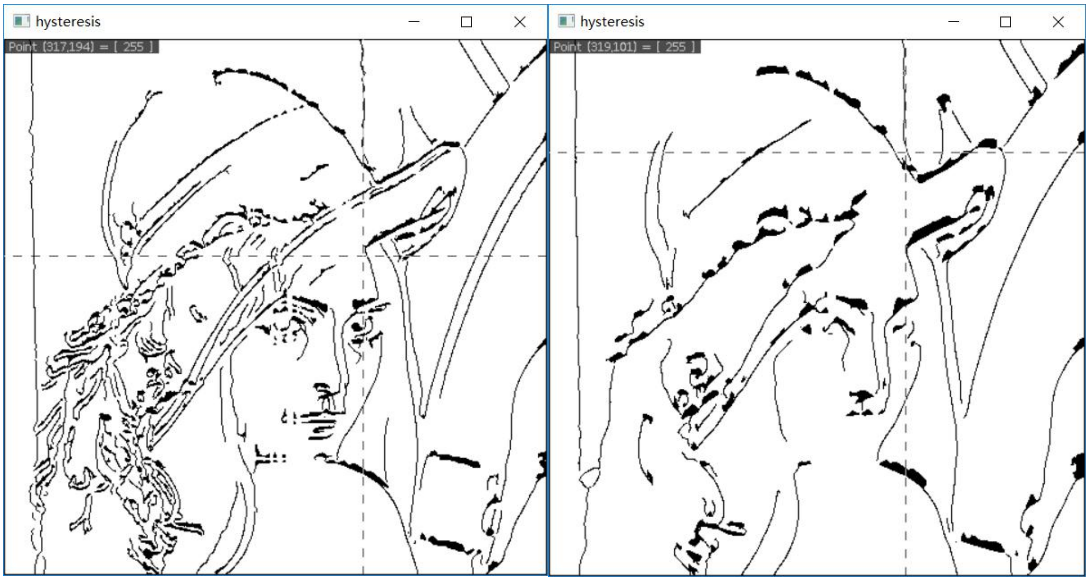

LEONARDO DICAPRIO
MARTIN SCORSESE
THE WOLF
OF WALL STREET
JORDAN BELFORT
TERENCE WINTER
MARTIN SCORSESE
DECEMBER 25



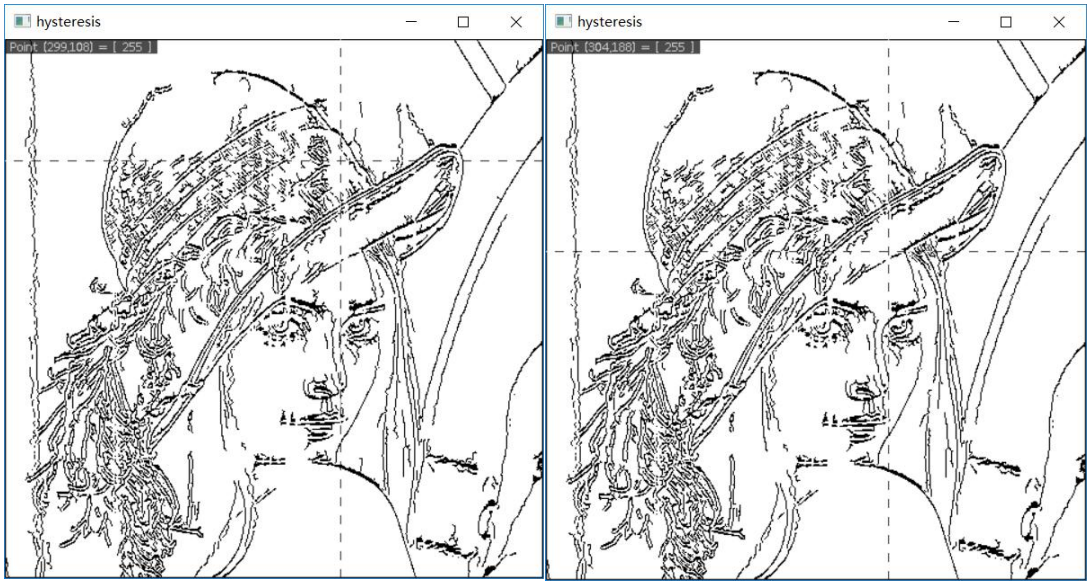


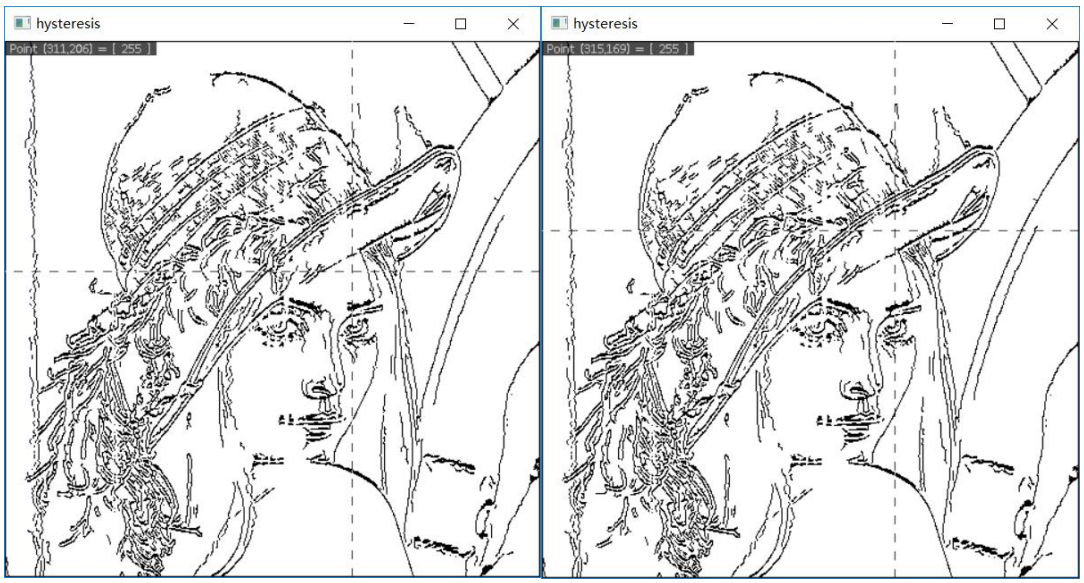
用 Lena 测试 Canny 算法，首先测试 sigma 对图像的影响，令 sigma = 0.5， 1.0， 2.0， 4.0(tlow = 0.3, thigh = 0.7)，测试的结果如下所示（分别为 0.5，1.0，2.0，4.0）：



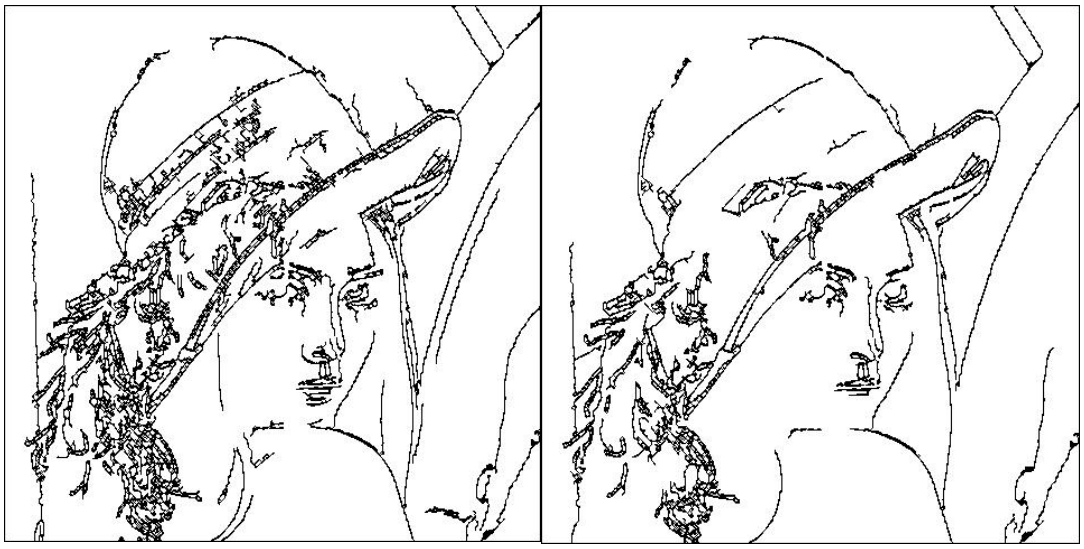


用 Lena 测试 Canny 算法，测试 tlow 对图像的影响，令 tlow = 0.2，0.3，0.4， 0.5(sigma = 1.0, thigh = 0.7)，测试的结果如下所示（分别为 0.2，0.3，0.4，0.5）：





用 Lena 测试 Canny 算法，测试 thigh 对图像的影响，令 thigh = 0.6， 0.7， 0.8， 0.9(sigma = 1.0, thigh = 0.3)，测试的结果如下所示（分别为 0.6，0.7，0.8，0.9）：



七、实验分析

首先分析三个参数对实验结果的影响（参照实验结果控制变量部分的图）：

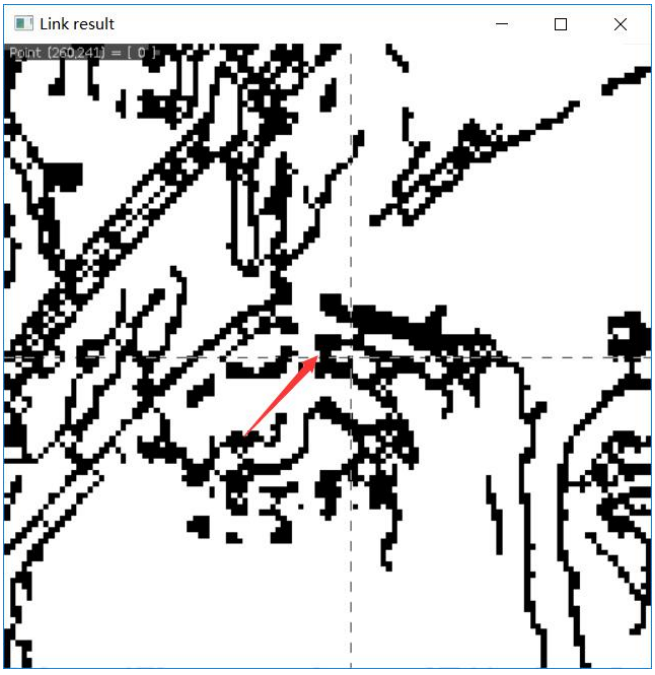
1）Sigma 对图像的影响是，当 tlow 和 thigh 不变时，Sigma 越大，即高斯模糊的程度越大，图像所保留的条纹越少.

2）tlow 对图像的影响是，当 thigh 和 sigma 不变时，tlow 越大，即在追踪过程中判定为 NOEDGE 的边会越多，所以图像所保留的条纹越少，但是变化并不明显.

3）thigh 对图像的影响是，当 sigma 和 tlow 不变时，thigh 越大，判定为 EDGE 的条件越苛刻，这时候保留的边数也会越少.

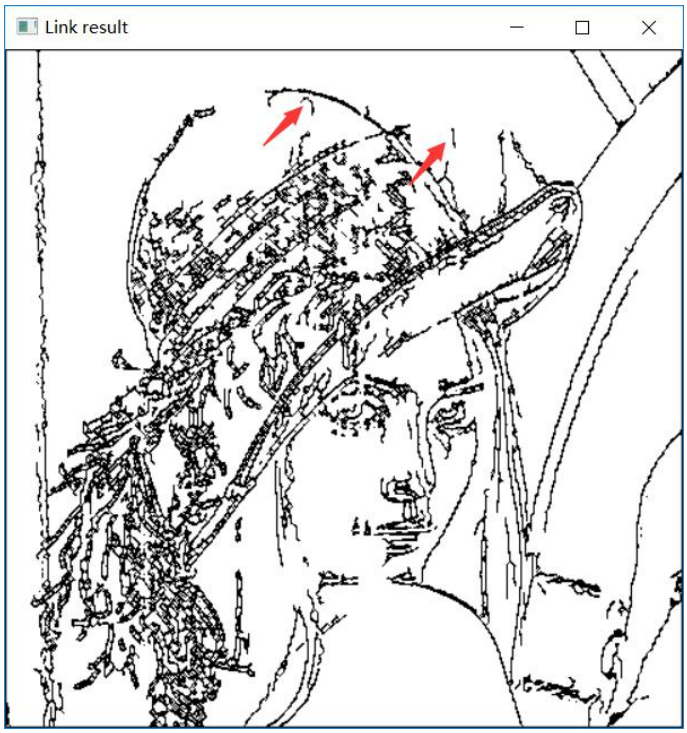
然后分析断点连接的恢复效果，这次算法中断点连接的效果并不是很明显，很多相隔较长的边就无法连接起来了，因此可以改进用蚁群算法来改进。而在本次实验中，连接的体现在下面的截图中，在连接前的图像如下所示：

连接后的图像如下所示：

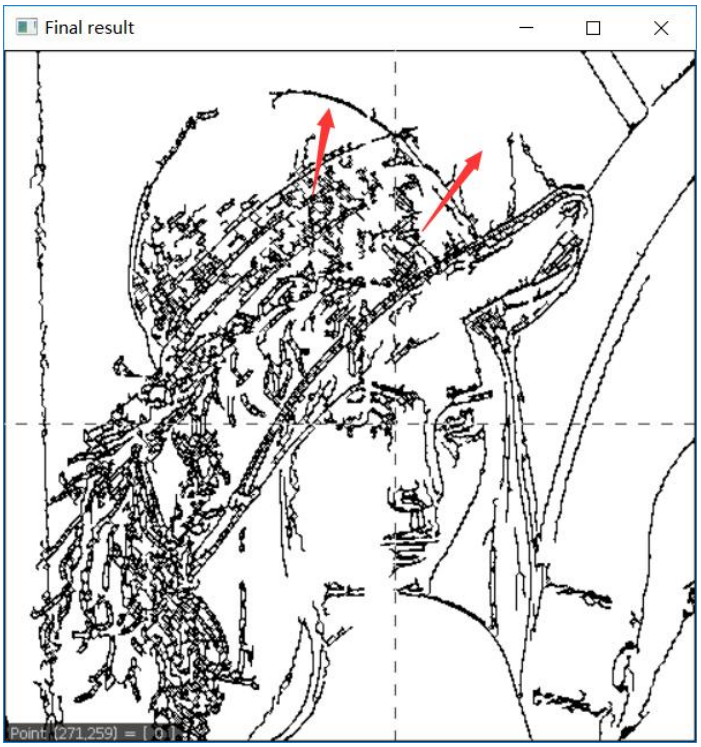


在红色箭头部分，也就是 Lena 的眼睛部分，可以看出连接效果.

最后是删除长度小于 20 的边，因为线的长度是包括八邻域的像素点，因此如果删除长度为 20 的话，图像的效果并不是很明显，因此删除 50 的效果可以看下述对比效果，在删除前的图像如下所示：



可以看出是有明显的瑕疵的，但是在调用删除的方法后，实现的结果如下：



可见一些长度较小的边被删除掉了.

八、参考网站

[1] 高斯模糊 http://www.ruanyifeng.com/blog/2012/11/gaussian_blur.html
[2] 矩阵卷积 https://blog.csdn.net/qq_32846595/article/details/79053277
[3] 图像梯度 https://blog.csdn.net/image_seg/article/details/78790968
[4] Canny 算子非极大值抑制
https://blog.csdn.net/kezunhai/article/details/11620357
[5] Canny 边缘检测 http://www.cnblogs.com/techyan1990/p/7291771.html
[6] 八邻域追踪算法
https://blog.csdn.net/sinat_31425585/article/details/78558849
[7] 边缘断裂处理算法 https://blog.csdn.net/cxf7394373/article/details/8790844