中山大學
SUN YAT-SEN UNIVERSITY

《计算机视觉》

（作业二实验报告）

学 院 名 称 ：	数据科学与计算机学院
专业（班级）：	17 软件工程 2 班
学 生 姓 名 ：	张淇
学　　号 ：	17343153
时　　间 ：	2019　年　9　月　21　日

成 绩 ：

作 业 二 ：用CImg重写给定代码并测试

一. 实验目的

1. 熟悉并使用CImg的相关函数。
2. 理解并重写Canny算法。

二. 实验内容

1. 改写并封装给定Canny代码（我的学号最后一位是0，所以改写code0）
2. 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。
3. 对算法的若干组参数，对所有测试图像进行测试，保存输出 Canny 算法每个主要步骤的结果图像，并分析各参数对结果的影响。

三. 实验原理

实现 canny 边缘检测的原理通俗来说就是用离散化的梯度逼近函数根据二维灰度矩阵梯度向量来寻找图像灰度矩阵的灰度跃变位置，然后再图像中将这些点连起来就形成了图像的边缘。

算法的实现步骤：

1) 应用高斯滤波来平滑图像，目的是去除噪声
2) 找寻图像的强度梯度（intensity gradients）

3) 应用非最大抑制(non-maximum suppression)技术来消除边误检(本来不是但检测出来是)

4) 应用双阈值的方法来决定可能的（潜在的）边界

5) 利用滞后技术来跟踪边界

四. 实验器材

Win10、 VS2017、 Cimg.h

五. 实验过程与结果

本实验将按照上述算法进行:

参数为: σ = 1， low_bound = 40, high_bound = 100

1. 将图片转换成灰度图:

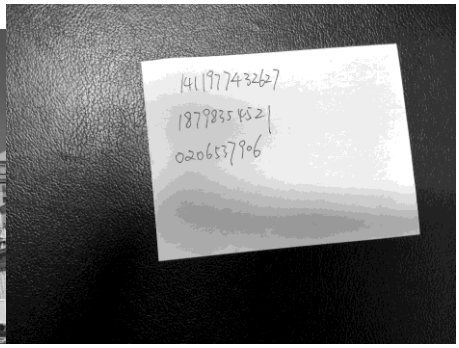
RGB三通道图转换成8位灰度图的公式如下:

r * 0.2126 + g * 0.7152 + b * 0.0722

程序中对应的函数为:

```
CImg<uchar> toGrayScale();
```

得到的结果如下:



2. 高斯滤波去噪

高斯滤波可以看成是对整张灰度图进行加权平均的过程，从而让整张图的灰度值变得平均，从而看起来更加平滑（噪声更少）。对应函数为：

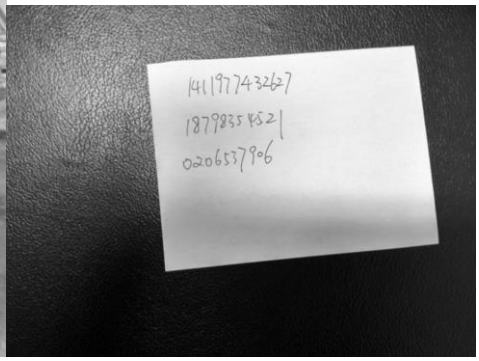
$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

其中σ的值决定了高斯函数的宽度。

考虑到debug，在实验中采用先生成滤波器再使用的方法，因此这一步骤对应两个函数：

```
vector<vector<double>> createFilter(int, int, double);
CImg<uchar> useFilter(vector<vector<double>>);
```

得到的结果如下：



3. 找寻图像的强度梯度

Canny算法的基本思想是找寻一幅图像中灰度强度变化最强的位置。所谓变化最强，即指梯度方向。平滑后的图像中每个像素点的梯度可以由Sobel算子（一种卷积运算）来获得，其在横纵方向上的公式如下：

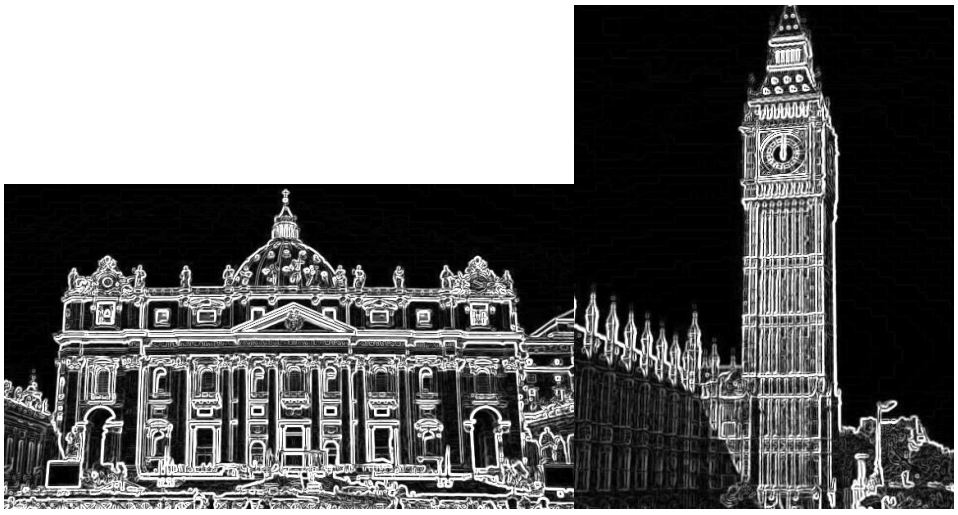
$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \qquad Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A$$

美中不足的是，Sobel算子并没有将图像的主体与背景严格地区分开来，换言之就是Sobel算子没有基于图像灰度进行处理，由于Sobel算子没有严格地模拟人的视觉生理特征，所以提取的图像轮廓有时并不能令人满意。

在本程序中对应的函数为：

```
CImg<uchar> sobel();
```

得到的结果为：



4. 应用非最大抑制（NMP）技术来消除边误检

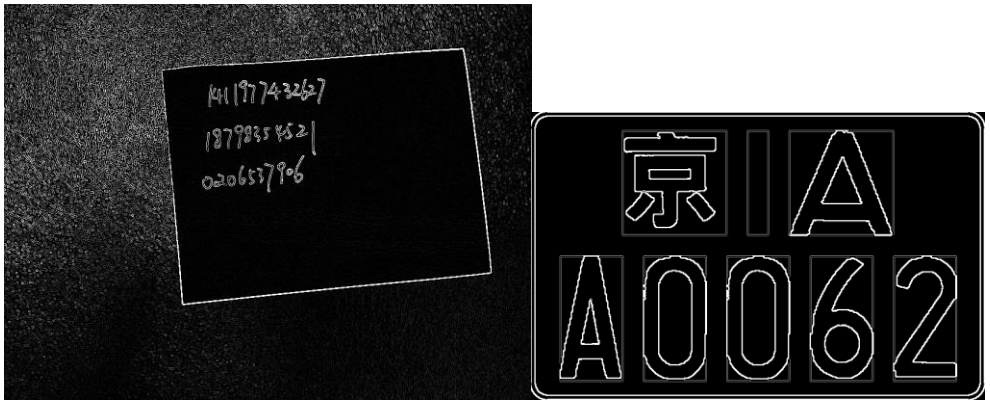
这一步的目的是将模糊（blurred）的边界变得清晰（sharp）。通俗的讲，就是保留了每个像素点上梯度强度的极大值，而删掉其他的值。对于每个像素点，进行如下操作：

- 将其梯度方向近似为以下值中的一个（0,45,90,135,180,225,270,315）（即上下左右和45度方向）
- 比较该像素点，和其梯度方向正负方向的像素点的梯度强度
- 如果该像素点梯度强度最大则保留，否则抑制（删除，即置为0）

在程序中对应的函数为：

```
CImg<uchar> nonMaxSupp();
```

得到的结果为；


京 A
A0062


AB·222BB

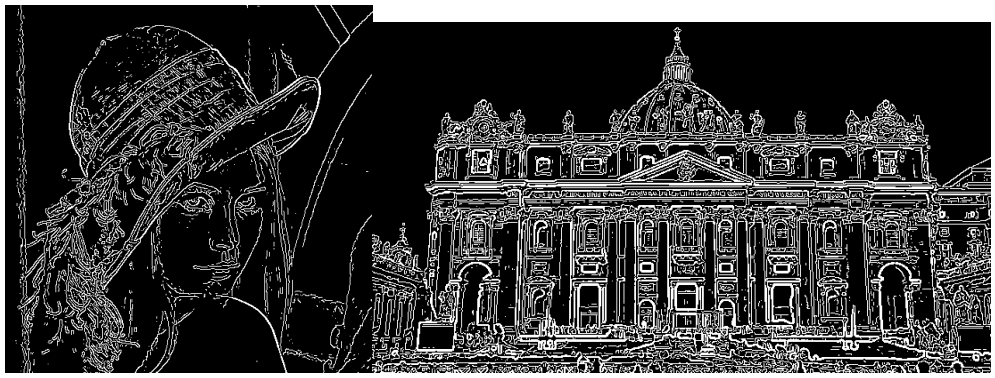
5. 应用双阈值的方法来决定潜在的边界

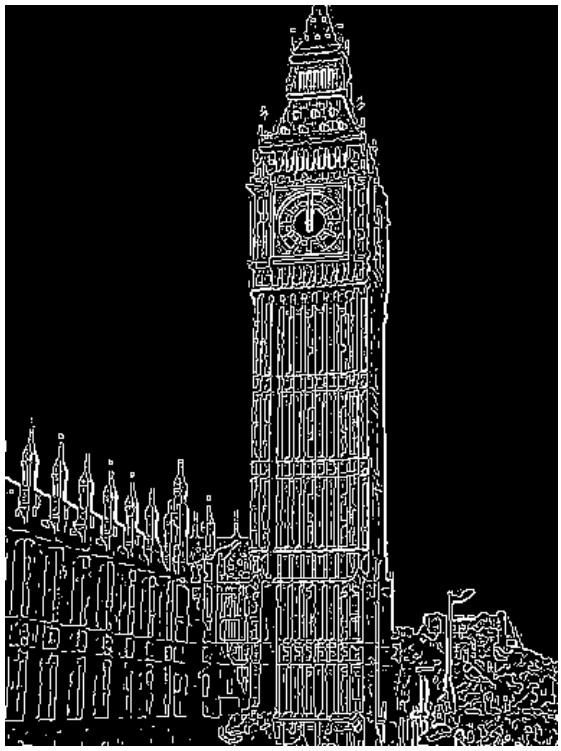
经过非极大抑制后图像中仍然有很多噪声点。Canny算法中应用了一种叫双阈值的技术。即设定一个阈值上界和阈值下界（opencv中通常由人为指定的），图像中的像素点如果大于阈值上界则认为必然是边界（称为强边界，strong edge），小于阈值下界则认为必然不是边界，两者之间的则认为是候选项（称为弱边界，weak edge）。

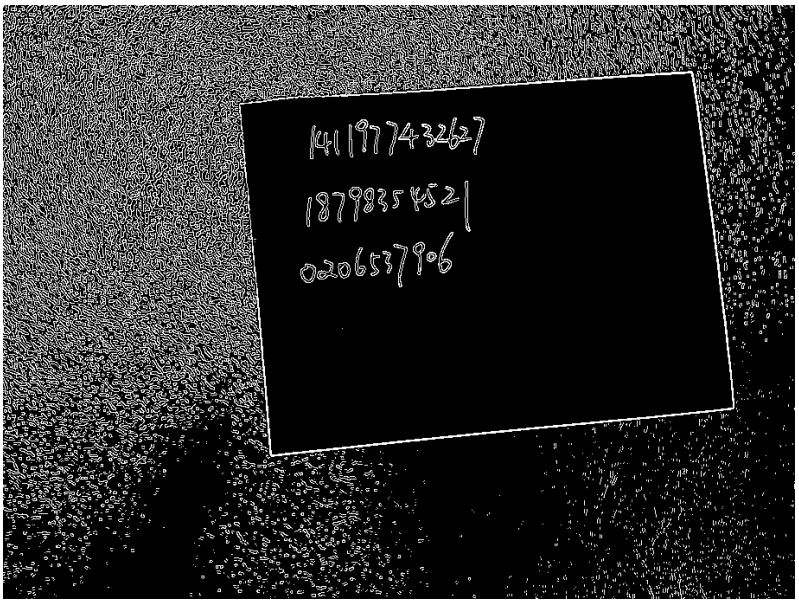
在本程序中对应的函数为：

```
CImg<uchar> threshold(int, int);
```

此步骤得到的结果为：







6. 增加函数：连接相邻的断边、删去长度小于20的边

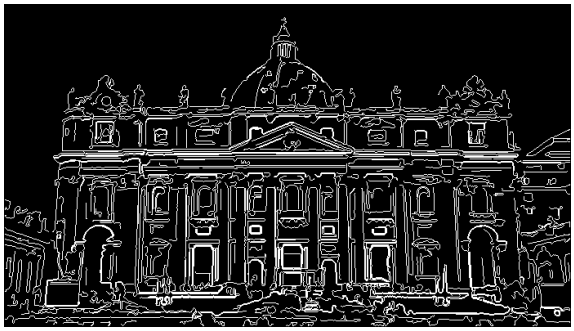
连接相邻断边的方法为：对每一个边缘上的点进行遍历，以这个像素为中心看一个矩形窗内的其他像素，如果存在边缘幅度响应M与边缘相角响应α都与中心像素差别不太大的像素，那么把这个像素也纳为边界内。

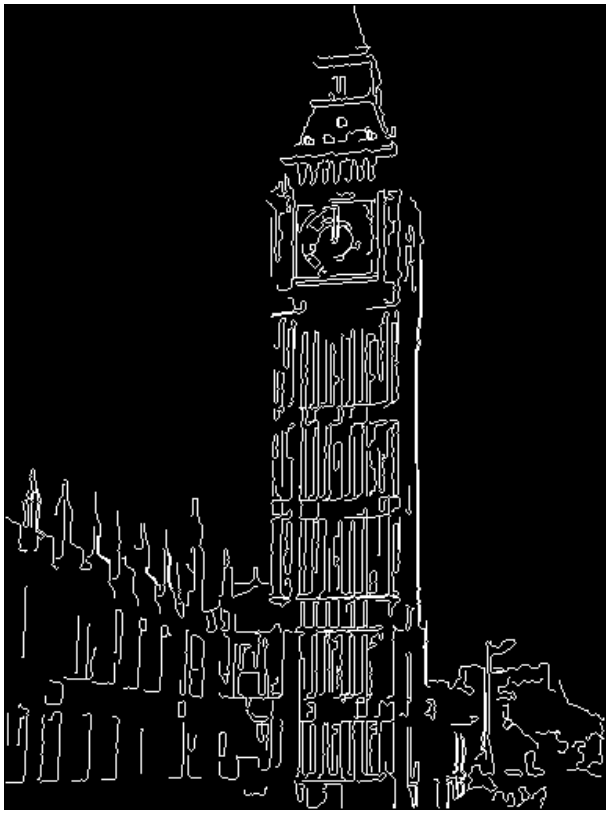
最后再把长度＜20的边缘删除。

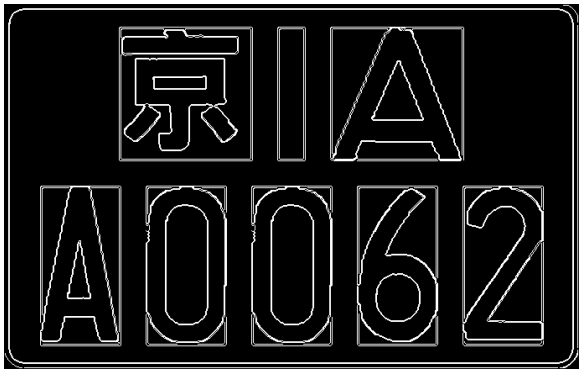
本程序中对应的函数为：

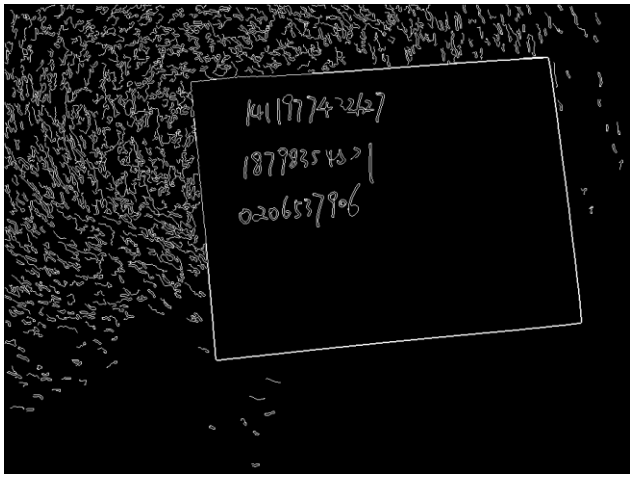
```
CImg<uchar> edgeTrack(CImg<uchar>)
```

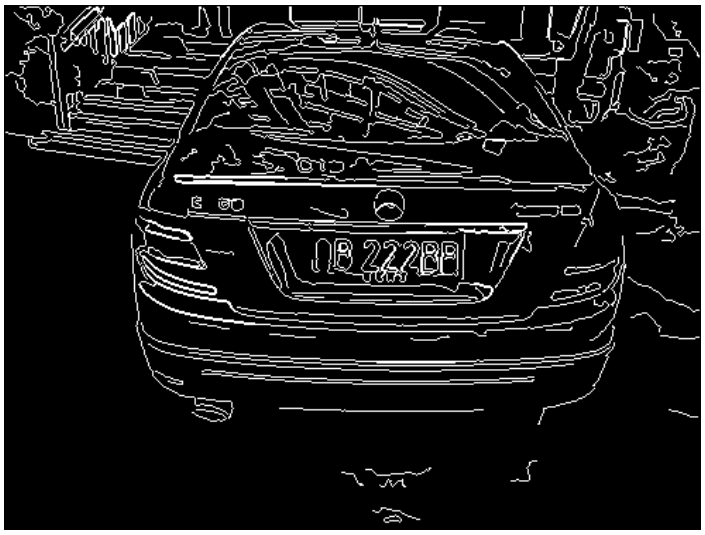
得到的结果为：











删除短边的优点：能够让图像的边缘更加清晰，更加接近物体轮廓，减少不必要的干扰。

删除短边得缺点：会使原来完好得一些短边删掉，使图像得边缘部分缺失，以及损失了图像本身得一些细节。

7. 保持双阈值、最短边缘长度（20）不变，调整σ的值（以lena.bmp为例）

sigma = 1

sigma = 2

sigma = 3

sigma = 4

sigma = 5

分析：通过调节高斯标准差大小可以看出来参数越小，对图像噪声点滤波的效果就越差，因而在最终边缘图中噪声点越多；而如果标准差过大，模糊效果太强，边缘之间的很多细节又会给模糊掉，导致最后分割图的边缘细节又不是那么明显。

8. 保持σ、双阈值不变，调整最短边缘长度：

distance = 0

distance = 10

distance = 20


distance = 30

分析：当不删除最短边（distance = 0）时，图像中存在过多断边边缘难以辨认。当最短边的长度逐渐上升时，图像中的轮廓（边缘）越来越清晰，但是当最短边的长度过大时，会丢失图像本身的细节，也会使本身不相接的边缘被“误删”。

9. 保持σ、最短边缘长度不变，调整阈值：


(20，100)

(40，50)

(40，100)

(40，200)

(80，100)

分析：修改低阈值后，低阈值减小，将会增加很多噪声；而低阈值增大，又会丢失很多强边缘像素。这是因为对于弱边缘像素（在低阈值和高阈值之间），这些像素可以从真实边缘提取也可以是因噪声或颜色变化引起的。为了获得准确的结果，应该抑制由后者引起的弱边缘像素。高阈值减小，此时将会增加很多强边缘像素，因为大于选定这个阈值的像素点都将被确定为边缘；而高阈值增大，原来的强边缘像素大部分会转化为弱边缘像素，丢失一部分边缘像素点。这是因为被划分为强边缘的像素点已经被确定为边缘，因为它们是从图像中的真实边缘中提取出来的。