计算机视觉第二次作业

朱明杰 15331441

一. 实验任务

1. 有三个 Canny 相关的 Code 以及测试数据若干，改写 Code1。
2. 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. 实验工具

Visual studio 2015

CImg Library

三. 算法流程

1. Canny 算子

i. 用高斯滤波器平滑图像

高斯滤波器$\mathbf{H}$可以写成

$$\mathbf{H}(x,y) = C\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

其中常数C为归一化因子，保证高斯滤波器（模板）的和为 1。

然后将滤波器$\mathbf{H}$与原图像$\mathbf{F}$做卷积，得到平滑后的图像

$$\mathbf{G} = \mathbf{F} * \mathbf{H}$$

ii. 用一阶偏导的有限差分来计算梯度的幅值和方向

其中所给的这个 SB 的代码用的 Sobel 算子来检测边缘，但是这个 SB 把 Sobel 算子给写反了。下面给出正确的 Sobel 算子：

$$\mathbf{H}_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \mathbf{H}_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

然后将上一步滤波后的图像$\mathbf{G}$与$\mathbf{H}_x$、$\mathbf{H}_y$做卷积即可。

$$\boldsymbol{\varphi}_x = \mathbf{G} * \mathbf{H}_x, \boldsymbol{\varphi}_y = \mathbf{G} * \mathbf{H}_y$$

这里顺便说一下，就减少计算量而言，我们可以先将 H_x、H_y 与第一步的高斯滤波器 H 进行卷积操作得到两个方向的梯度平滑算子，然后将得到的算子分别与原图像 F 做卷积。不过这个 SB 写的代码里面没有这样写，所以我还是按照步骤一步一步来。

在之后的计算中我们会利用到幅值与辐角的信息，所以保留幅值与辐角即可。具体计算如下：

$$\boldsymbol{\varphi} = \sqrt{\boldsymbol{\varphi}_x^2 + \boldsymbol{\varphi}_y^2}$$

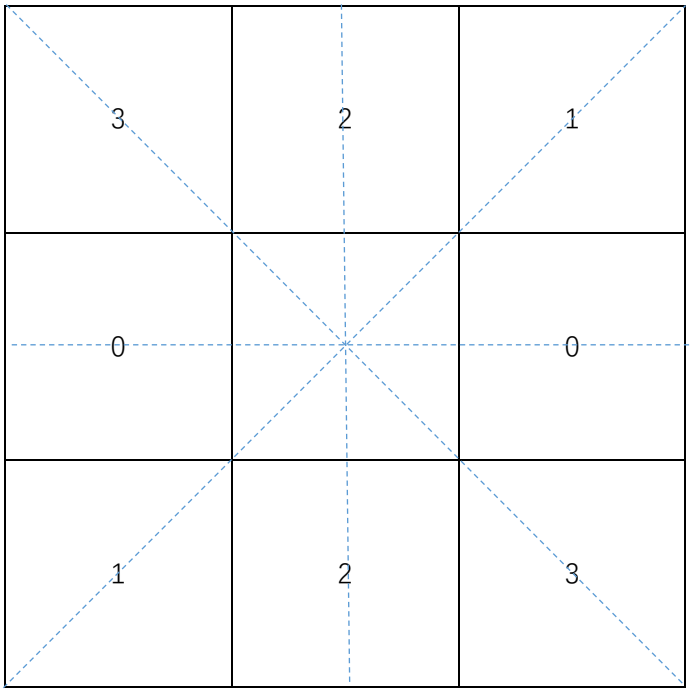
$$\boldsymbol{\theta}_{\boldsymbol{\varphi}} = \arctan\frac{\boldsymbol{\varphi}_y}{\boldsymbol{\varphi}_x}$$

其中，为了后面计算步骤的方便（其实是为了和 MATLAB 中的中间步骤进行对比），可以将幅值矩阵$\boldsymbol{\varphi}$的值域归一化；为了避免“被零除”的现象，我们可以利

用 atan2 函数来计算$\boldsymbol{\theta_{\varphi}}$。那个 SB 不知道 atan2 函数不要紧，他居然可以弄不清角度制与弧度制，真的是个人才。

iii. 对梯度幅值进行非极大值抑制

仅仅得到全局的梯度并不足以确定边缘，因此为确定边缘，必须保留局部梯度最大的点，而抑制非极大值。这里，我们就要利用到上一部计算的梯度的方向。



如上图所示，一个像素点的 3×3 邻域中，蓝虚线为所有可能的比较方向。作所有角平分线，得到辐角范围判断的依据。

具体比较方法：在每一点上，邻域的中心像素M与沿着梯度线的两个像素相比。如果M的梯度值不比沿梯度线的两个相邻像素梯度值大，则令$M = 0$。

记这一步处理后的边缘矩阵为$\mathbf{N}$。

iv. 用双阈值算法检测和连接边缘

上面得到的边缘矩阵$\mathbf{N}$的值域还是$[0,1]$，并不是我们想要的二值矩阵。注意到此时的$\mathbf{N}$中还有一些假边缘。减少假边缘段数量的典型方法是对使用一个阈值。将低于阈值的所有值赋零值。但问题是如何选取阈值？

我们这里用双阈值算法。

- 设置两个阈值a_1、a_2，通常$2.5a_1 = a_2$
- a_2阈值下假边缘少，但是轮廓有间断；对于轮廓的端点，利用a_1阈值的 8 邻点位置寻找可以连接到轮廓上的边缘。算法不断地收集边缘，直到将轮廓连接起来为止

到了这一步，这个 SB 已经是在完全乱搞了。他直接在代码中暴力写死a_1、a_2的取值，然而我觉得这肯定是不可取、有失公正的。所以我这里参考了一下 MATLAB 的 Canny 算子实现方法。MATLAB 中定义了两个变量：非边缘像素的占比 PercentOfPixelsNotEdges 与阈值比 ThresholdRatio，在 MATLAB 里面分别取 0.7 和 0.4。非边缘像素的占比取 0.7 的含义为：将边缘矩阵$\mathbf{N}$的所有值从小到大排序，前 70%的值对应的位置均非强边缘，后 30%的值对应的位置均为强边缘。阈值比即为a_2/a_1，和我们上面的取法是一样的。不过，MATLAB 中梯度算子的实现是用 Prewitt 算子来做的；在我的实践过程中，非边缘像素的占比 PercentOfPixelsNotEdges 值还应取得更大，差不多为 0.7777777 是

最好的。

2. 边缘细化

虽然 Canny 算子只负责边缘检测，但是边缘细化仍然是边缘提取中非常重要的一环。细化是一种二值图像处理运算。可以把二值图像区域缩成线条，以逼近区域的中心线。细化的目的是减少图像成分，只留下区域最基本的信息，以便进一步分析和处理。

其思想为：在 3x3 邻域内检查图像前景中的每一个像素，迭代削去简单边界点，直至区域被细化成一条线。

对于每一个像素，如果

A. 没有上邻点（下邻点、左邻点、右邻点）；

B. 不是孤立立点或孤立立线；

C. 去除该像素点不会断开连通区域；

D. 删除该像素点；

E. 重复 A-D 步骤直到没有像素点可以去除。

四. 算法代码（CImg）实现

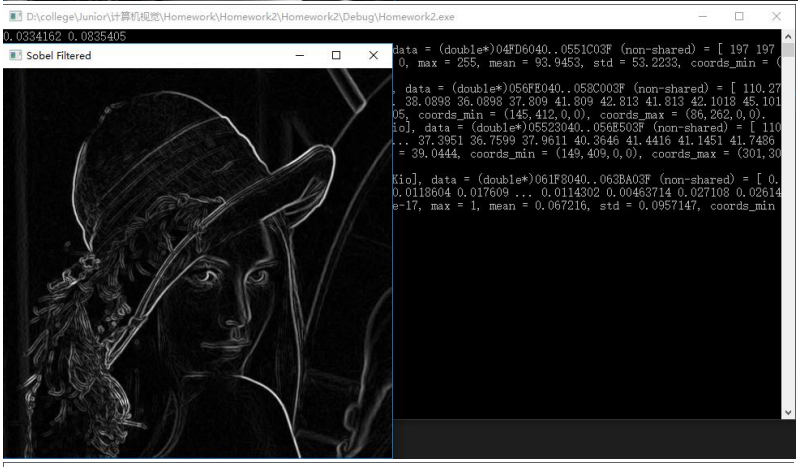
见压缩包下代码。

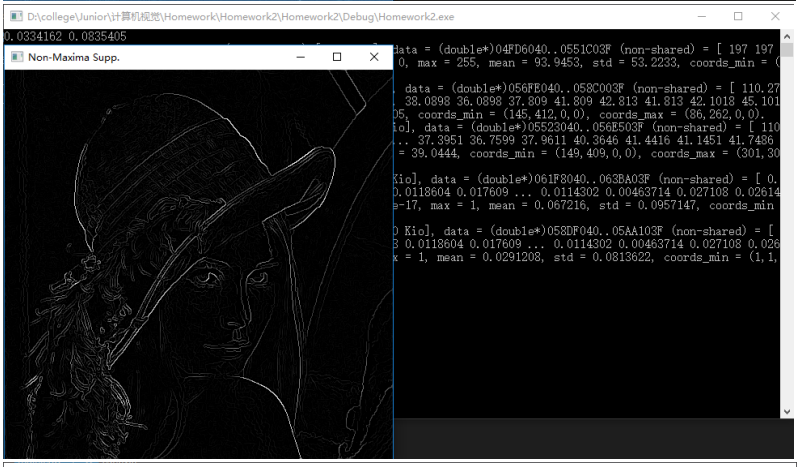
五. 实验结果

1. lena.bmp

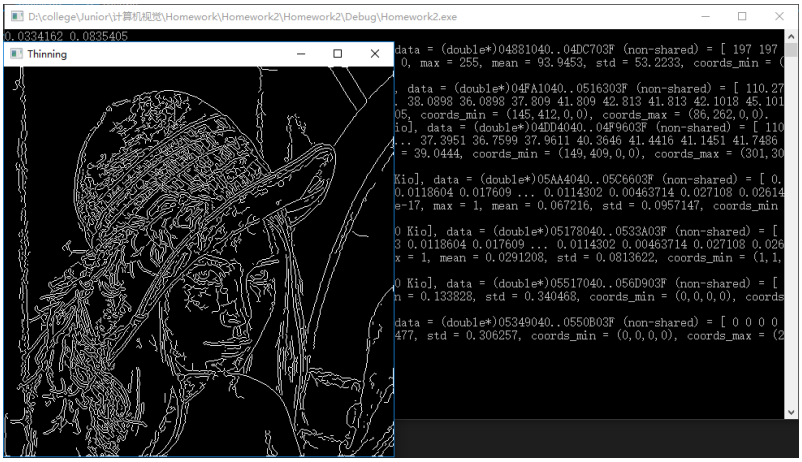
实验效果（逐步显示）：


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0334162 0.0835405
Gaussian Blur

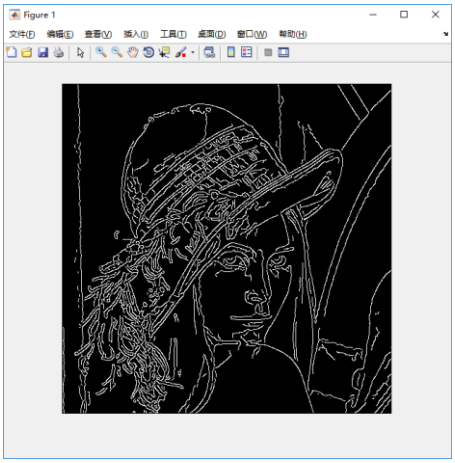

D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0334162 0.0835405
Sobel Filtered


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0334162 0.0835405
Non-Maxima Supp.


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0334162 0.0835405
Double Threshold

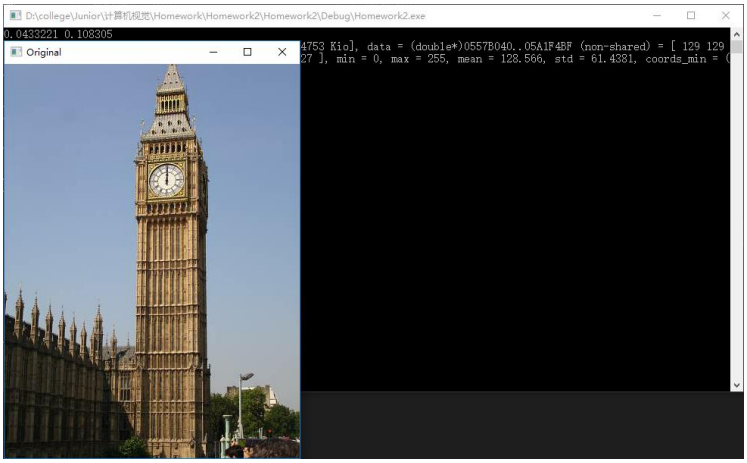


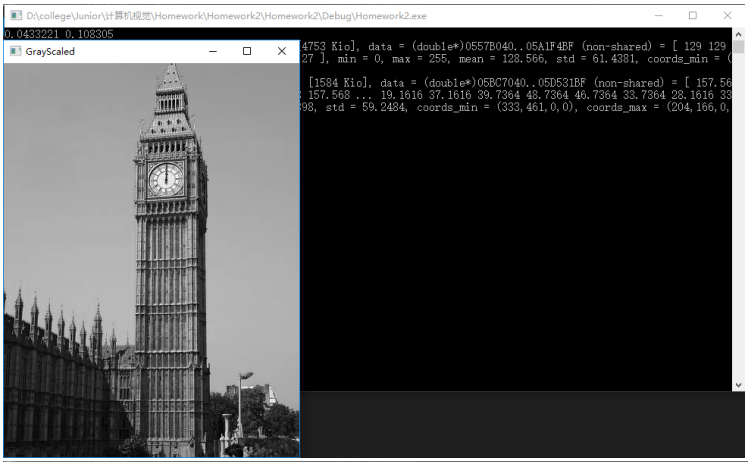
MATLAB 中 Canny 算子对照效果：

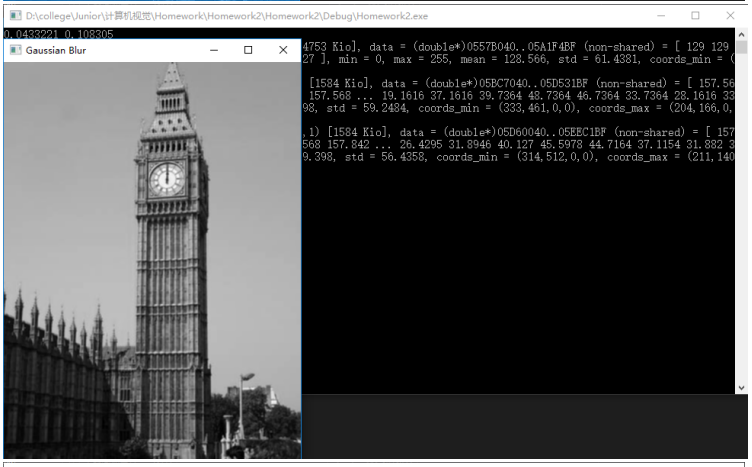


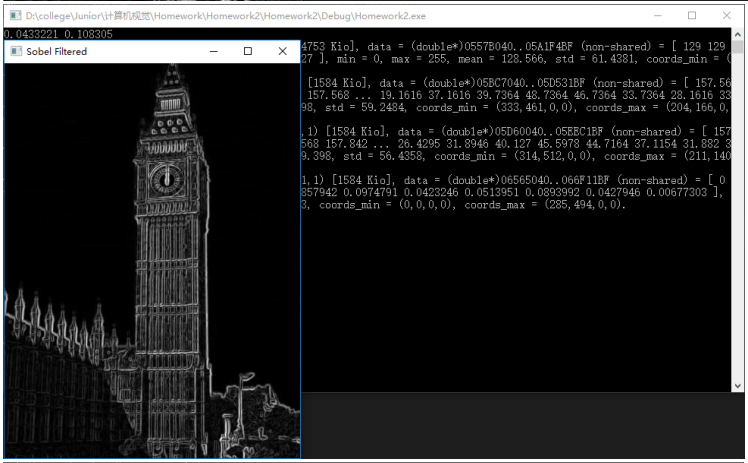
2. bigben.bmp

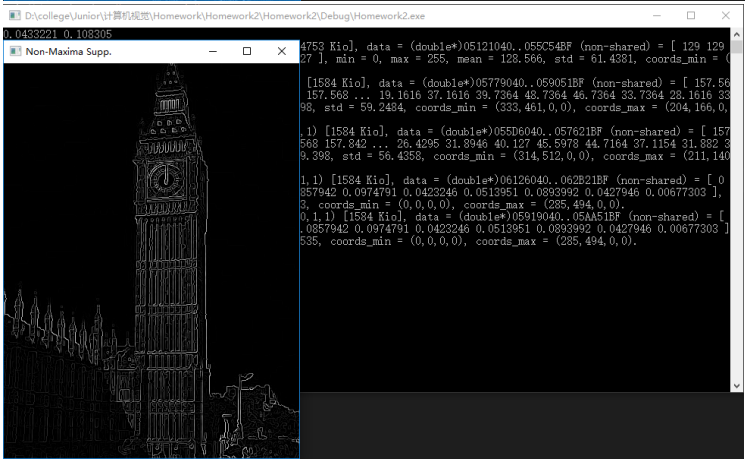
实验效果（逐步显示）：

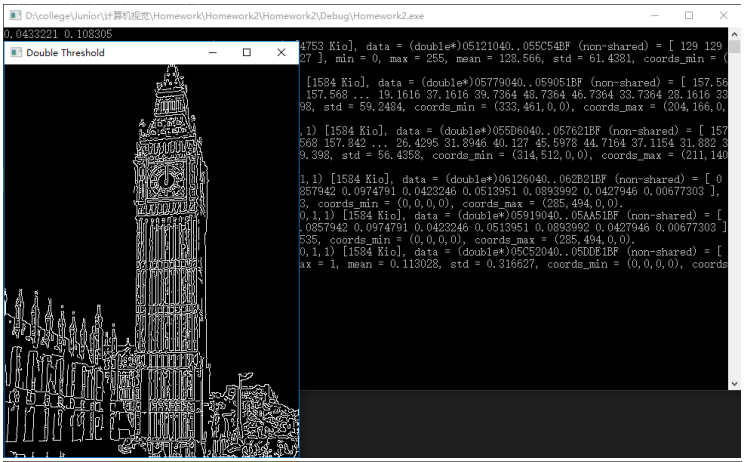


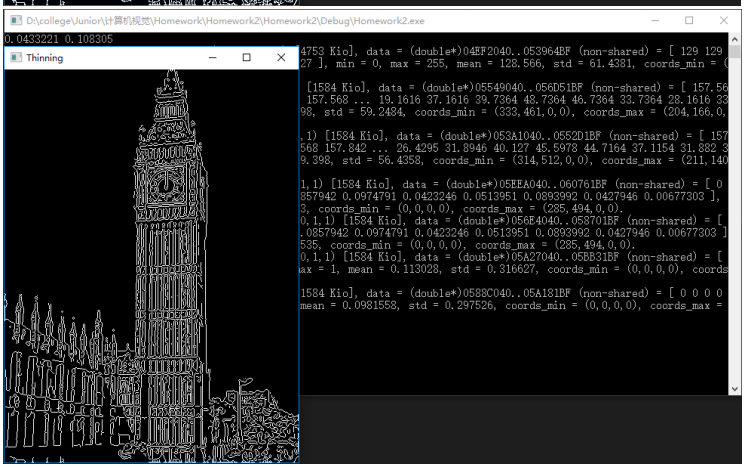

D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0433221 0.108305
GrayScaled


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0433221 0.108305
Gaussian Blur

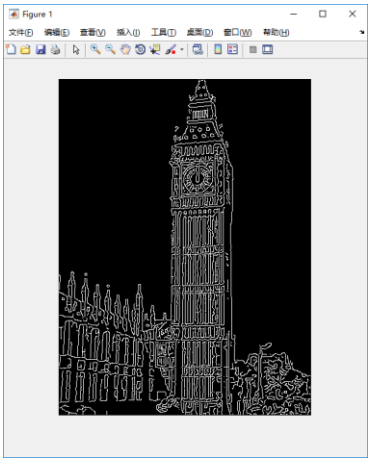

D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0433221 0.108305
Sobel Filtered


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0433221 0.108305
Non-Maxima Supp.

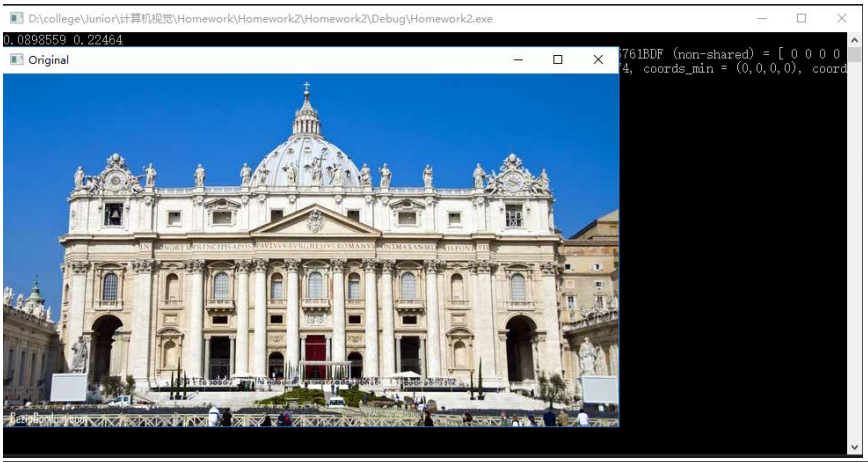




MATLAB 中 Canny 算子对照效果：

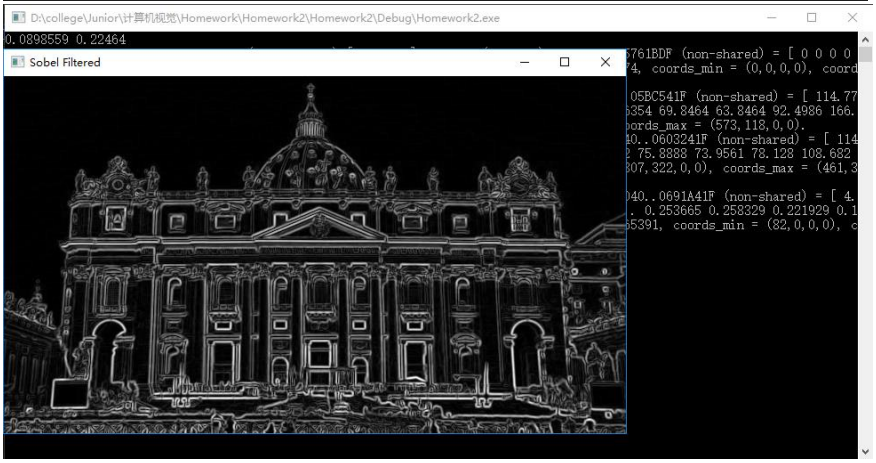


3. stpietro.bmp
   实验效果（逐步显示）：

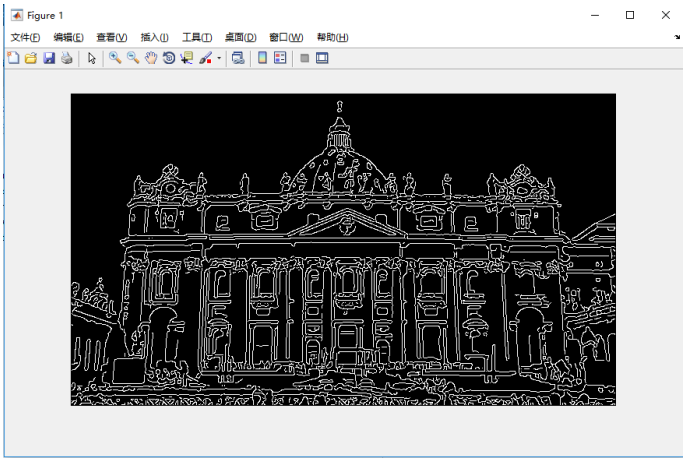

D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0898559 0.22464
Original


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0898559 0.22464
GrayScaled


D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0898559 0.22464
Gaussian Blur

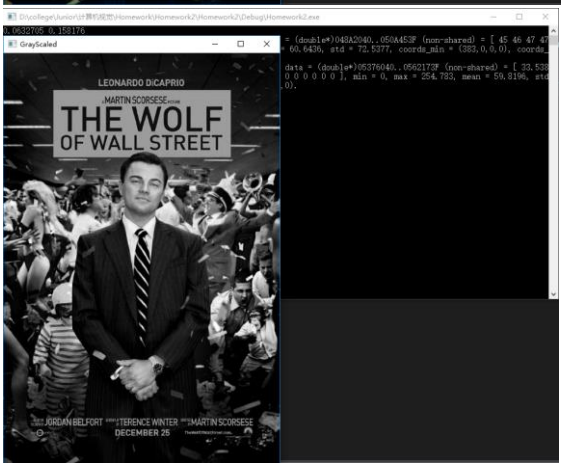

D:\college\Junior\计算机视觉\Homework\Homework2\Homework2\Debug\Homework2.exe
0.0898559 0.22464
Sobel Filtered

MATLAB 中 Canny 算子对照效果：

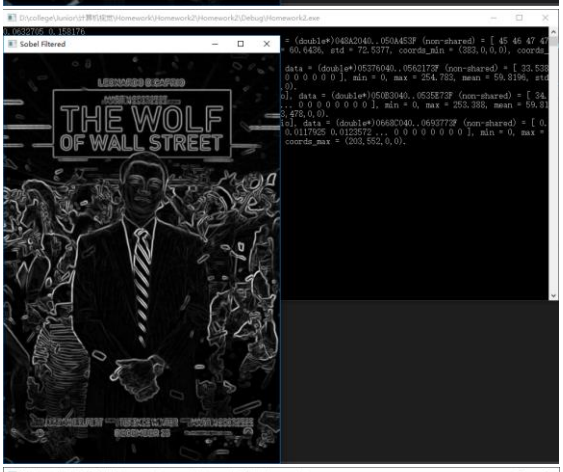


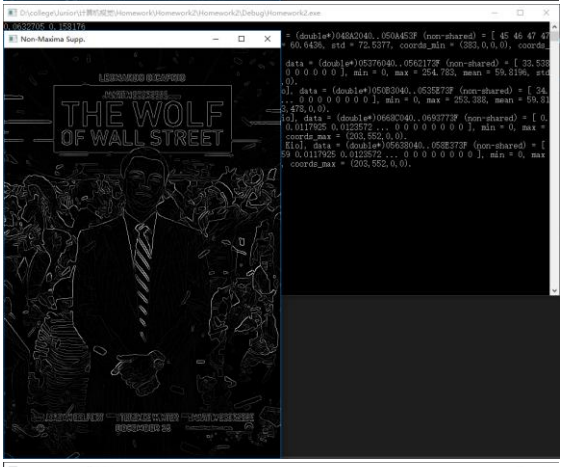
4. twows.bmp

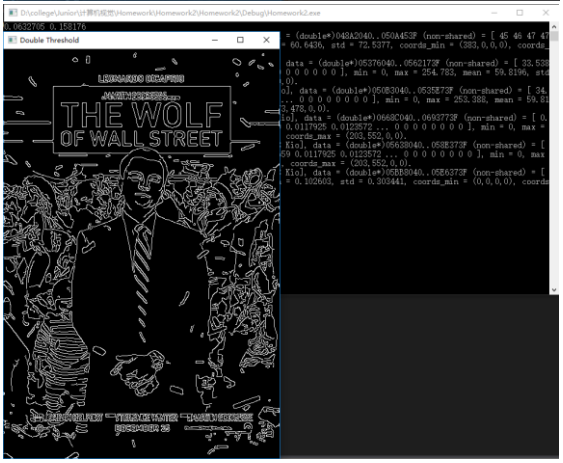
实验效果（逐步显示）：

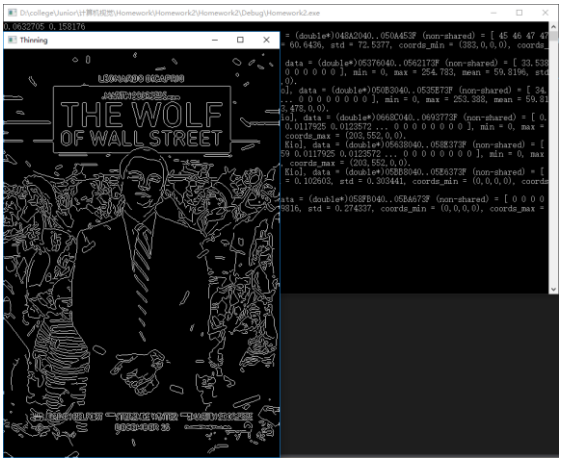



Gaussian Blur
LEONARDO DICAPRIO
MARTIN SCORSESE
THE WOLF
OF WALL STREET
JORDAN BELFORT
TERENCE WINTER
MARTIN SCORSESE
DECEMBER 25

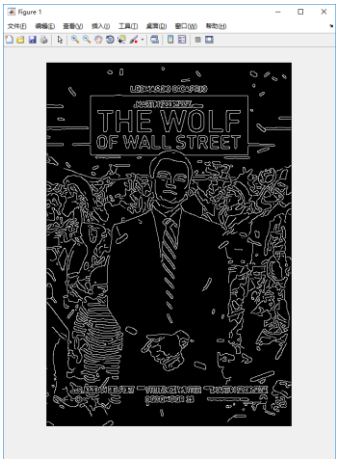

Sobel Filtered
THE WOLF
OF WALL STREET


Non-Maxima Supp.
THE WOLF
OF WALL STREET


Double Threshold
THE WOLF
OF WALL STREET



MATLAB 中 Canny 算子对照效果：



六. 分析与评价

总的来说，基本上较好地实现了 Canny 算子检测边缘，检测出来的边缘的质量也较好。在做的过程中按照之前老倪的课件一步一步做，也没有发现什么明显的问题。

具体的细节层面上的东西可能与网上资料、MATLAB 代码有部分出入。例如一些网上资料的非极大值抑制部分采用了插值的方法估计梯度方向目标点的梯度幅值 ;又例如 MATLAB 的细化操作是用的二值图像形态学的方法。

另外，写代码累，改别人的 SB 代码更累！这个 SB 的封装又加多了一层矩阵赋值操作，会导致额外的开销。