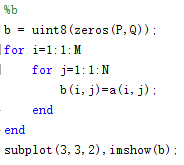
作业二

(南京大学 计算机科学与技术系, 南京 210093)

# Assignment 1

**a到b：**

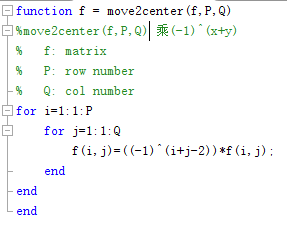
设置填充参数P=2M和Q=2N，然后填充0得到大小为P\*Q的·新图像b：

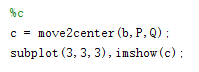


**b到c：**

将b乘(-1)^(x+y)：

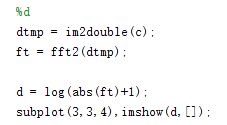
实现为matlab函数move2center：





**c到d：**

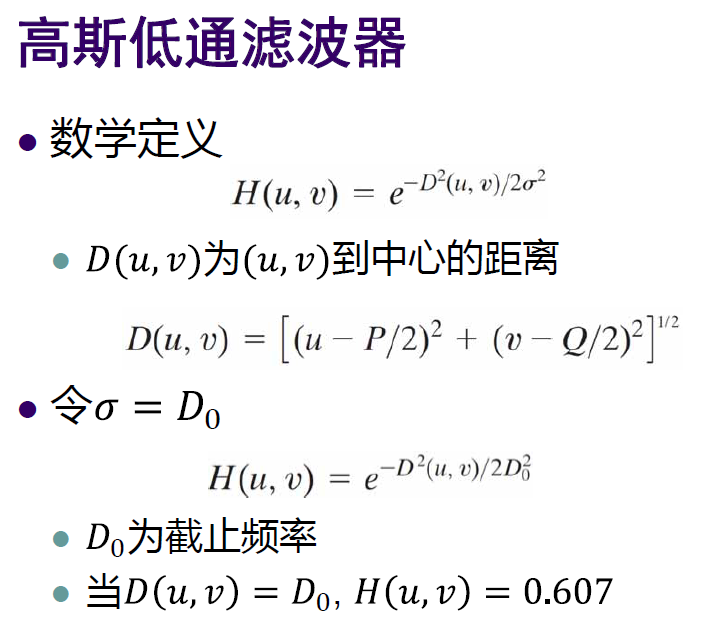
对得到的c进行二维离散傅里叶变换：



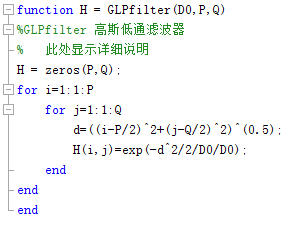
这里使用abs函数对复数矩阵ft的各个元素取模，并使用对数函数对数值范围进行缩小，因为变换后的ft中最大值和最小值有几个数量级的差距，直接显示会使有些细节“丢失”，使用对数函数可以缩小范围，增强图像显示效果。

**d到e：**

生成高斯低通滤波器：



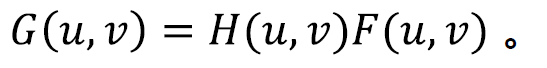
根据数学式实现高斯低通滤波器的matlab生成函数GLPfilter：

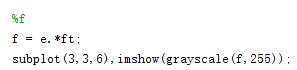


**e到f：**

使用高斯低通滤波器对傅里叶变换后的ft进行滤波：

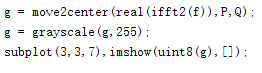
阵列相乘：





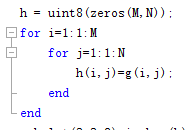
**f到g：**

对f进行反傅里叶变换，然后乘(-1)^(x+y)：



**g到h：**

对生成的g进行裁剪：

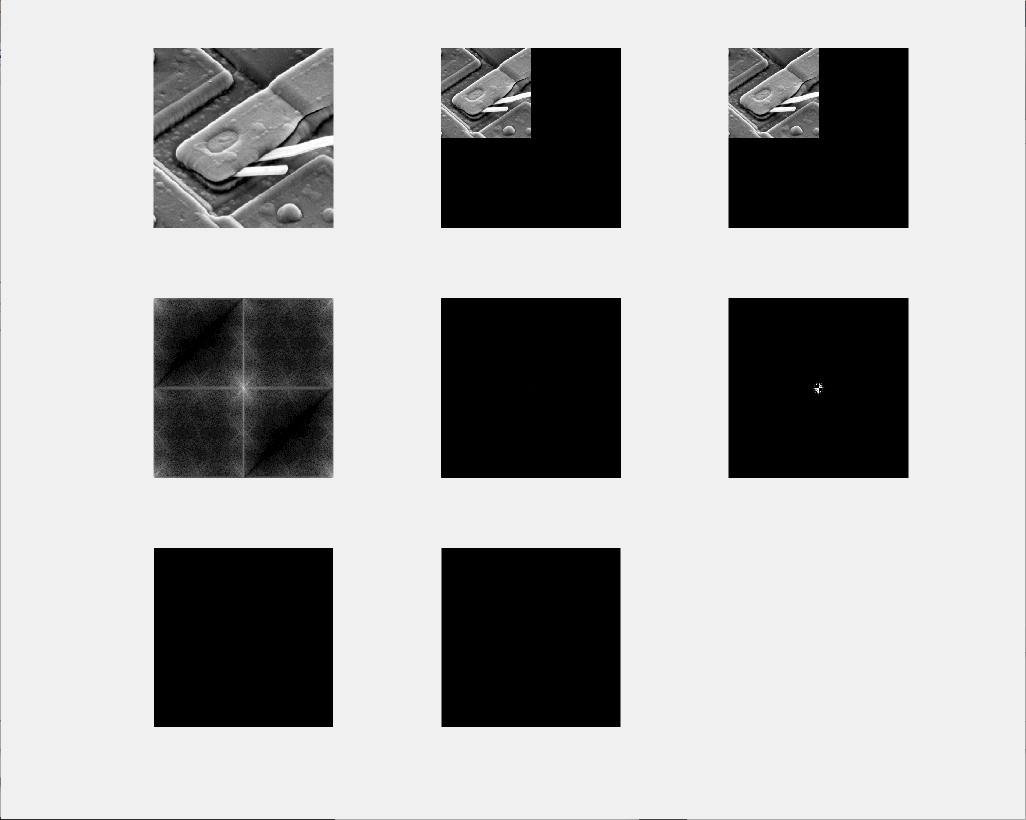


Grayscale函数：

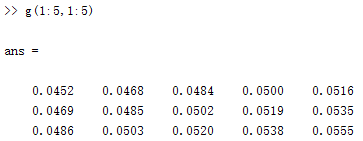
浮点数与整数转换的截断误差问题：

在没有使用grayscale函数的时候：

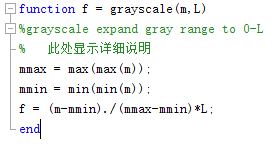
显示结果如图：



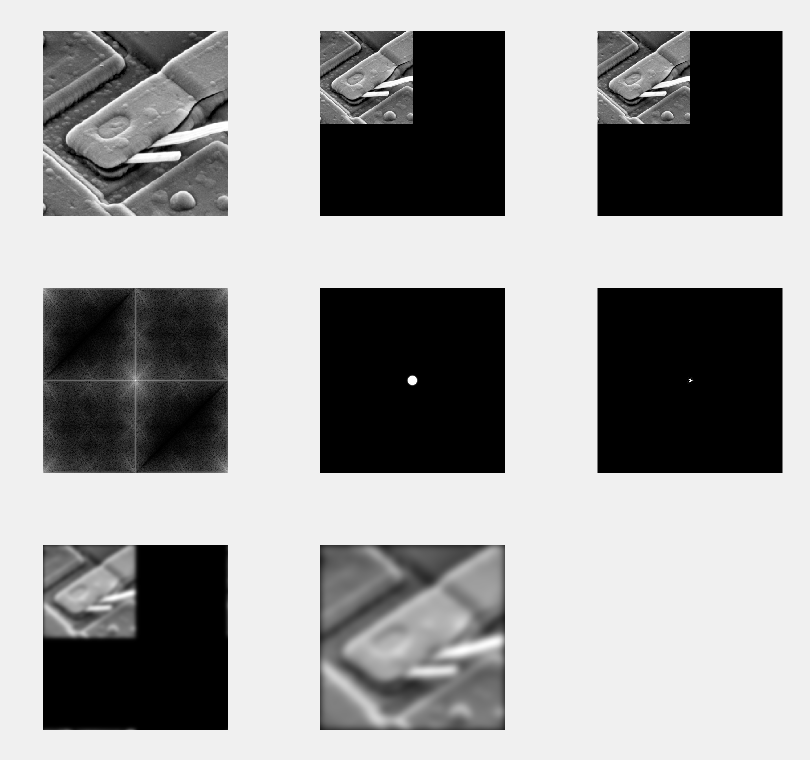
e，f, g, h都是全黑。后来查看g的值：



可以看到都是小于1的小数，转为uint8就转为0，直接使用imshow(uint8(g))进行显示就是黑色。所以使用grayscale函数将灰度范围转为0-255：



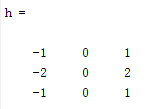
最后结果如图：



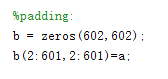
# Assignment 2

**空间域滤波：(2\_1.m)**

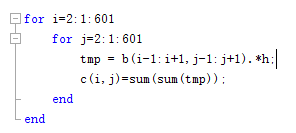
使用3\*3的Sobel算子：



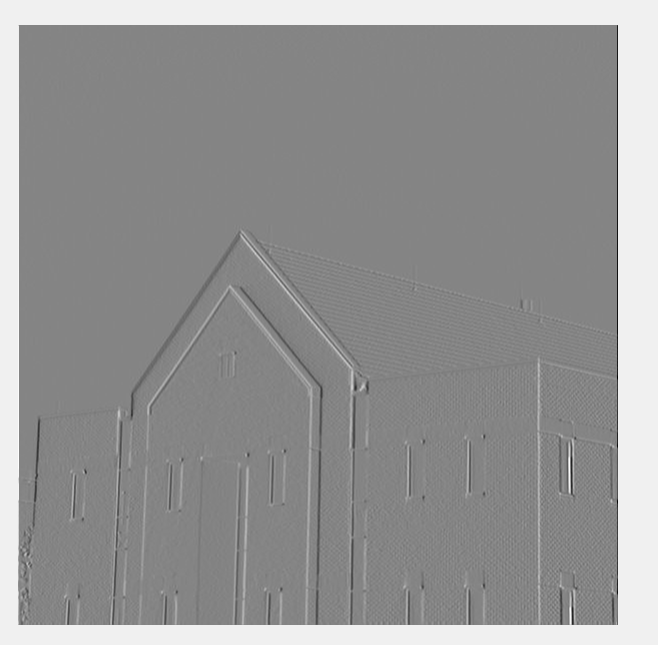
填充0：



计算：



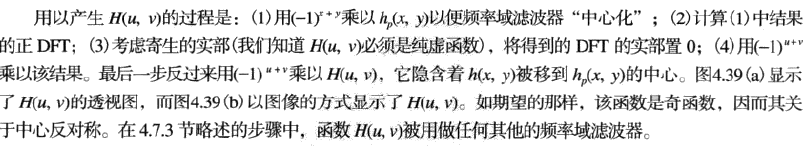
最后得到空间域滤波结果：



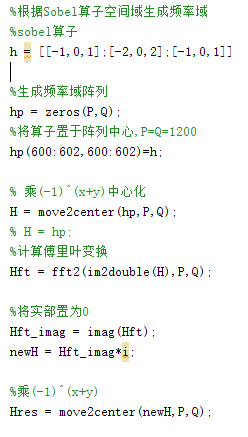
**频率域滤波：**

根据前面3\*3的空间域Sobel算子产生频率域滤波器：

参考教材P168的方法：



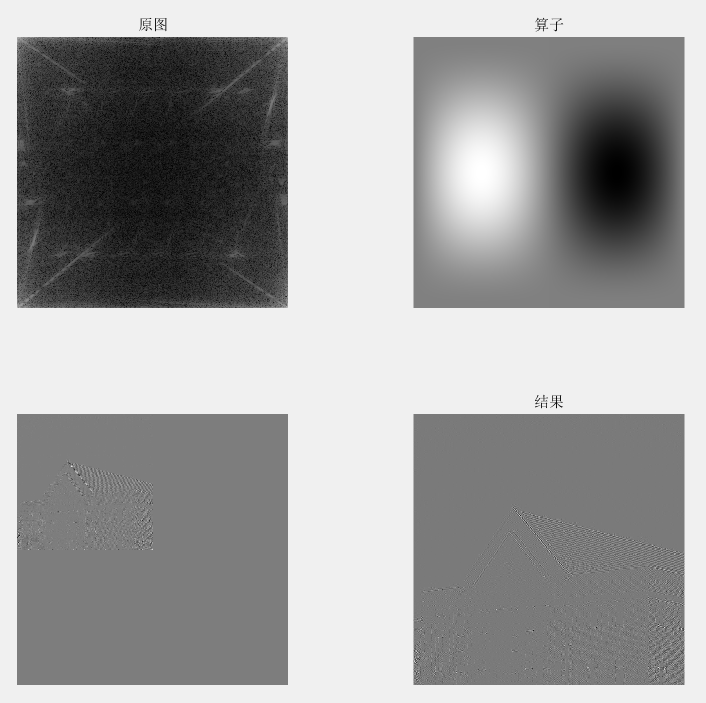
设计Matlab代码如下：



最终生成频率域滤波器Hres；

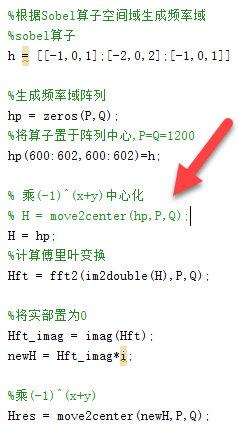
注意在该过程中分别在傅里叶变换前后乘(-1)的x+y次方(move2center函数)以使其中心化；

使用上面的代码，最终产生的结果如图：



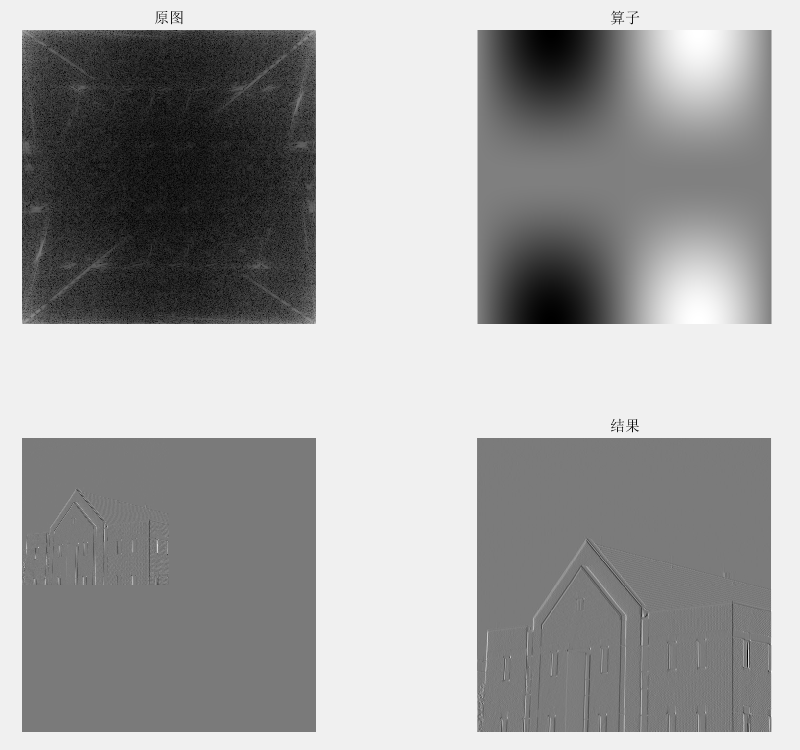
可以看到产生的算子频率域图片与教材以及老师ppt上面的算子图片一致，但是产生的结果房子的轮廓并不明显。

后来我不断改动，取消第一次乘(-1)的x+y次方：

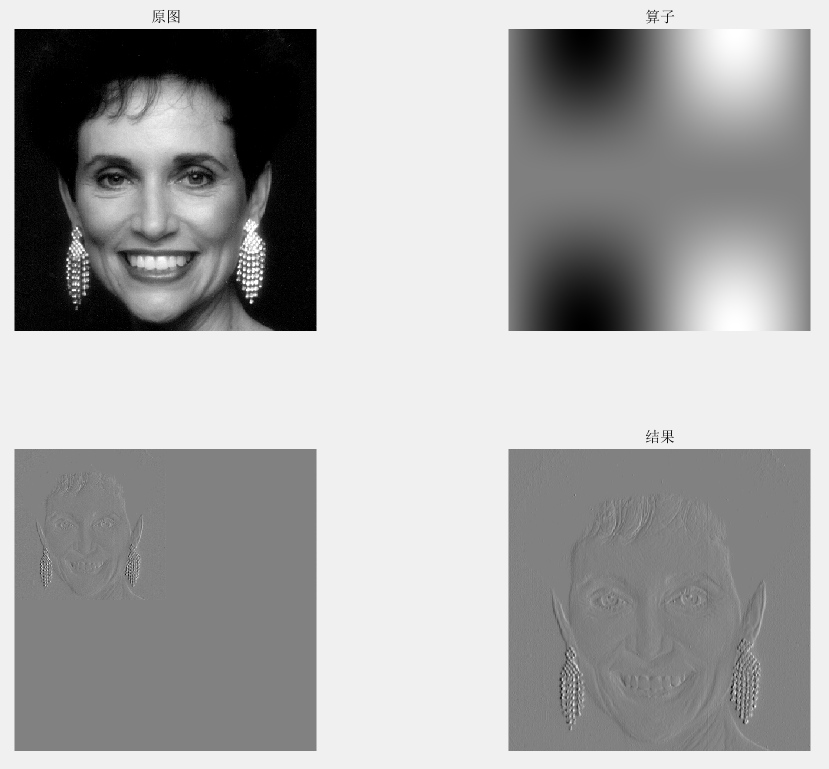


箭头所示，取消第一次乘(-1)的(x+y)次方。

产生的结果：



很明显，取消第一次乘负一的x+y次方后，算子频率域图片改变了，但结果却正确了，结果与空间域滤波一致。后来我又使用了assignment4的图片测试：

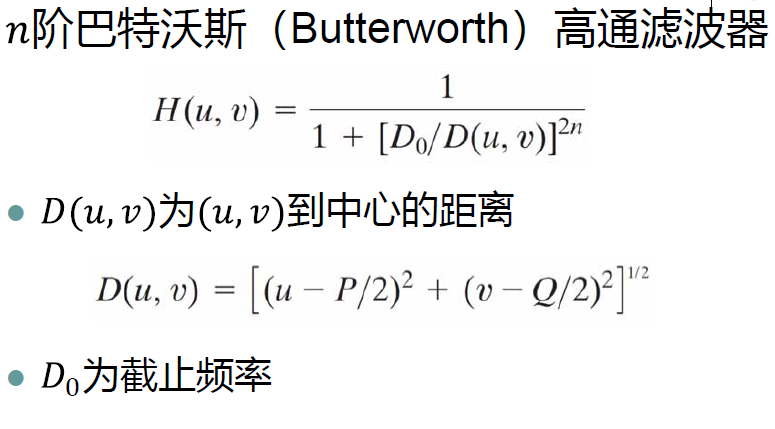


可以看到这种情况下仍然得到正确结果。

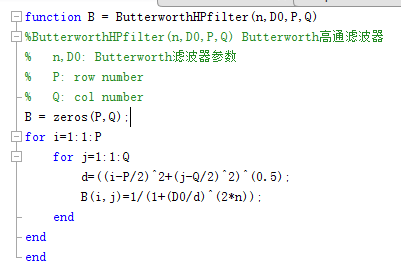
所以最终我没有按照教材的算法计算，取消了第一次乘(-1)的(x+y)次方。

# Assignment 3

使用巴特沃斯高通滤波器：



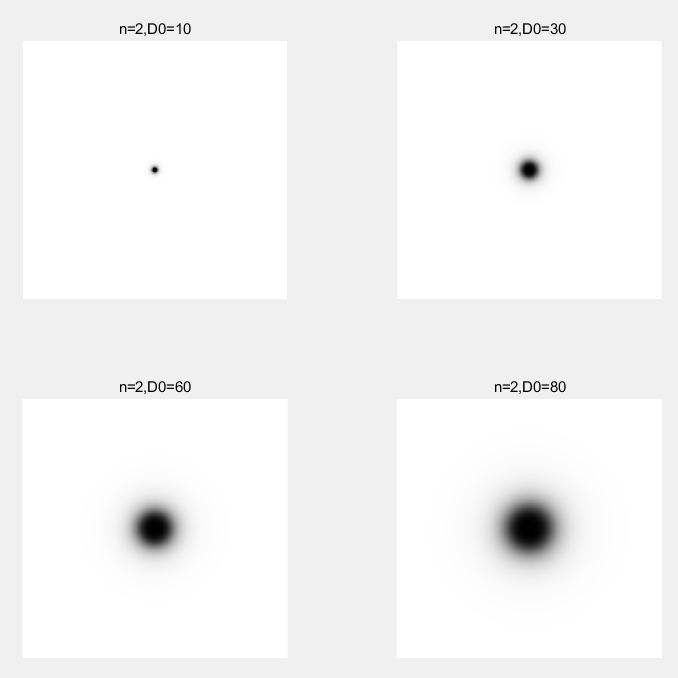
根据数学定义实现matlab函数：



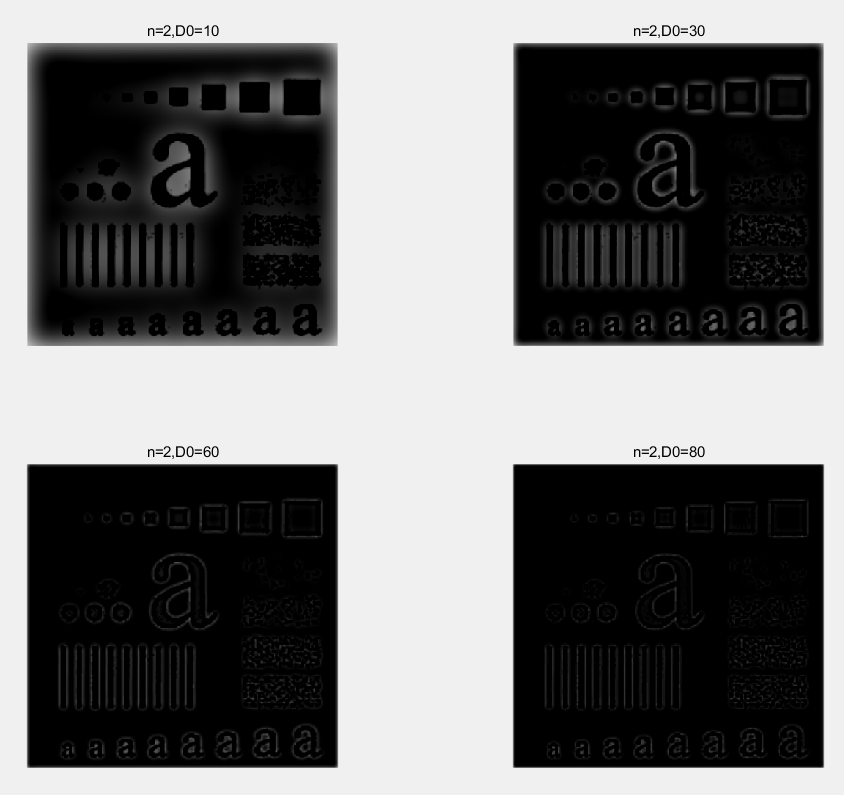
设定不同的参数，使用巴特沃斯高通滤波器对图像滤波；

结果如图：

二阶巴特沃斯高通滤波器，选择不同的D0值，图像：



滤波结果：



可以看到，D0的值越大，滤波后的图像边缘越平滑，但图像也更暗。

# Assignment 4

美颜算法：

算法来自于<https://www.cnblogs.com/Imageshop/p/4709710.html>；

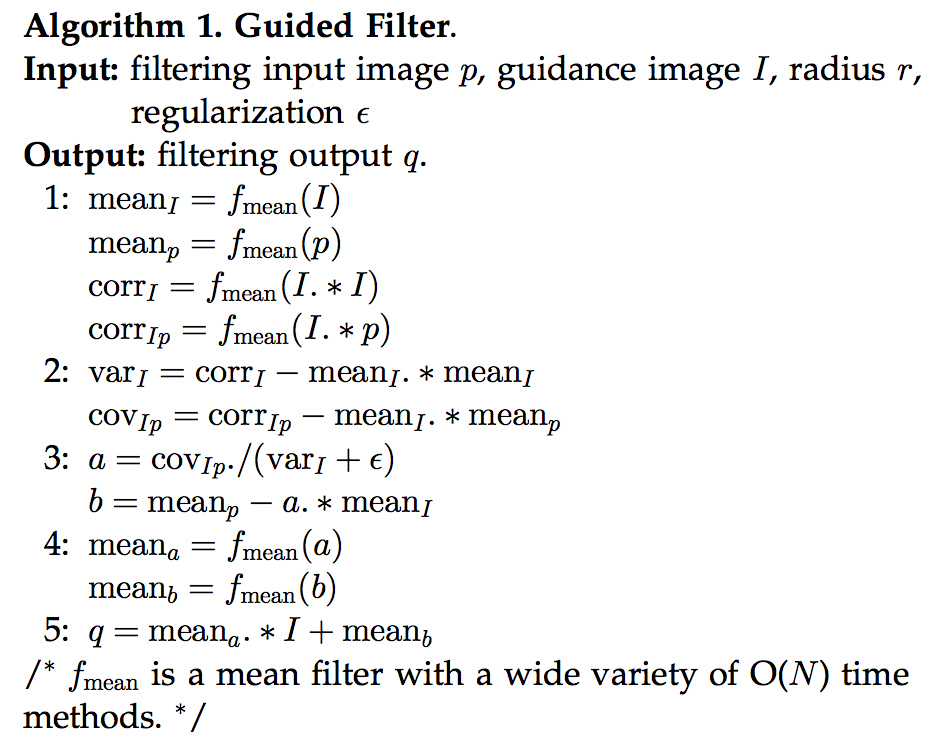
最终算法公式为：



EPFFilter为保持边缘的平滑滤波器；这里使用导向滤波器；Opacity为不透明度；

**导向滤波：**通过一张引导图，对初始图像p（输入图像）进行滤波处理，使得最后的输出图像大体上与初始图像P相似，但是纹理部分与引导图相似。当引导图与输入图 p 为同一个图像的时候，导向滤波的效果与双边滤波的效果类似，可以保持边缘。

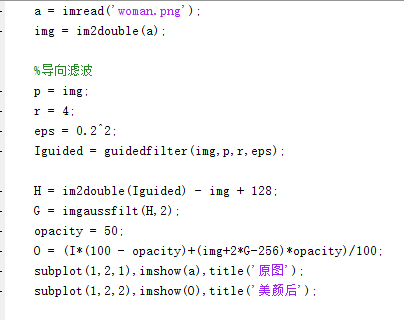
算法如图：



代码实现见guidedfilter.m和boxfilter.m两个文件，两个文件的代码来自网络。

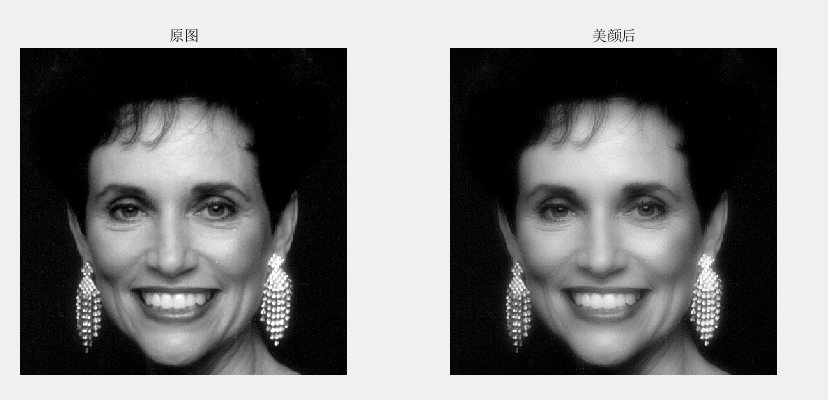
美颜算法实现：(不透明度Opacity取50%，高斯模糊半径取2)

beautify\_gray.m：



效果：

使用beautify\_gray.m



对RGB图像，分别对三个通道进行美颜再合并（beautify\_rgb.m），效果如图：

