

数字图像处理HW5：频域滤波

- 林汉宁
- 自动化52
- 2150504042
- 2019.4.2 本次工程使用python3.6 与opencv3，采用OOP方法编程实现所有要求

低通频域滤波器

任务要求

频域低通滤波器：设计低通滤波器包括 butterworth and Gaussian (选择合适的半径，计算功率谱比),平滑测试图像test1和2;分析各自优缺点；

任务实现

我使用python numpy库根据高斯噪声与butterworth定义完成实现。

结果展示

以test1为例

- D0=10：



- 高斯：

◦ 巴特沃斯：



• D0=20:

◦ 高斯：



◦ 巴特沃斯：



• D0=40:

◦ 高斯：



◦ 巴特沃斯：



• D0=100:

◦ 高斯：



- 巴特沃斯-



结论：

D_0 越小，两低通滤波器的平滑效果越明显，即滤波后图像越模糊。同时，功率谱比也随 D_0 减小而变小。在 D_0 相同时，高斯低通滤波器和二阶巴特沃斯滤波器的平滑效果没有明显的区别，只是二阶巴特沃斯滤波器似乎比高斯滤波器平滑程度更高一点点，高斯低通滤波器的功率谱比小于二阶巴特沃斯低通滤波器的功率谱比。

频域高通滤波器

要求：

频域高通滤波器：设计高通滤波器包括butterworth and Gaussian，在频域增强边缘。选择半径和计算功率谱比，测试图像test3,4：分析各自优缺点；

实现

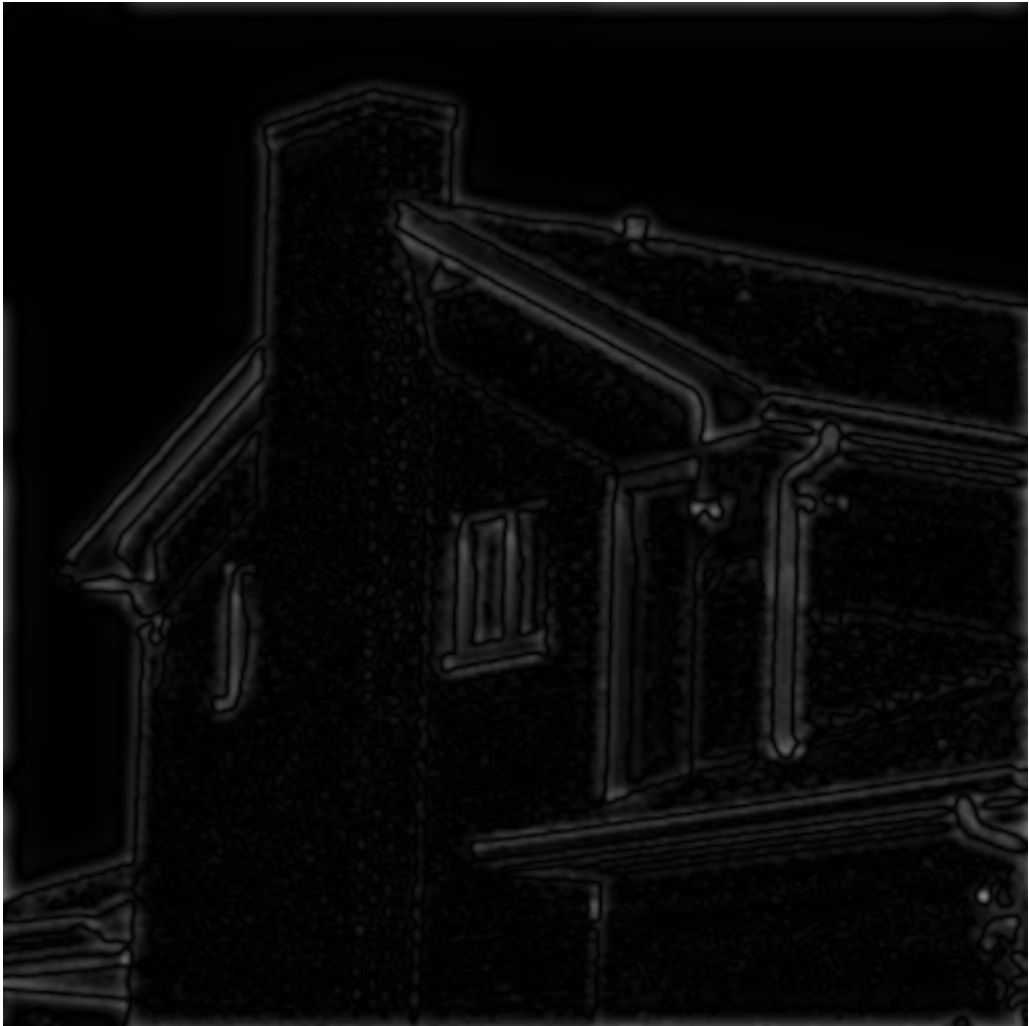
我使用python numpy库根据高斯噪声与butterworth定义完成实现。

结果展示

以test4为例

- $D_0=10$

○ 高斯：

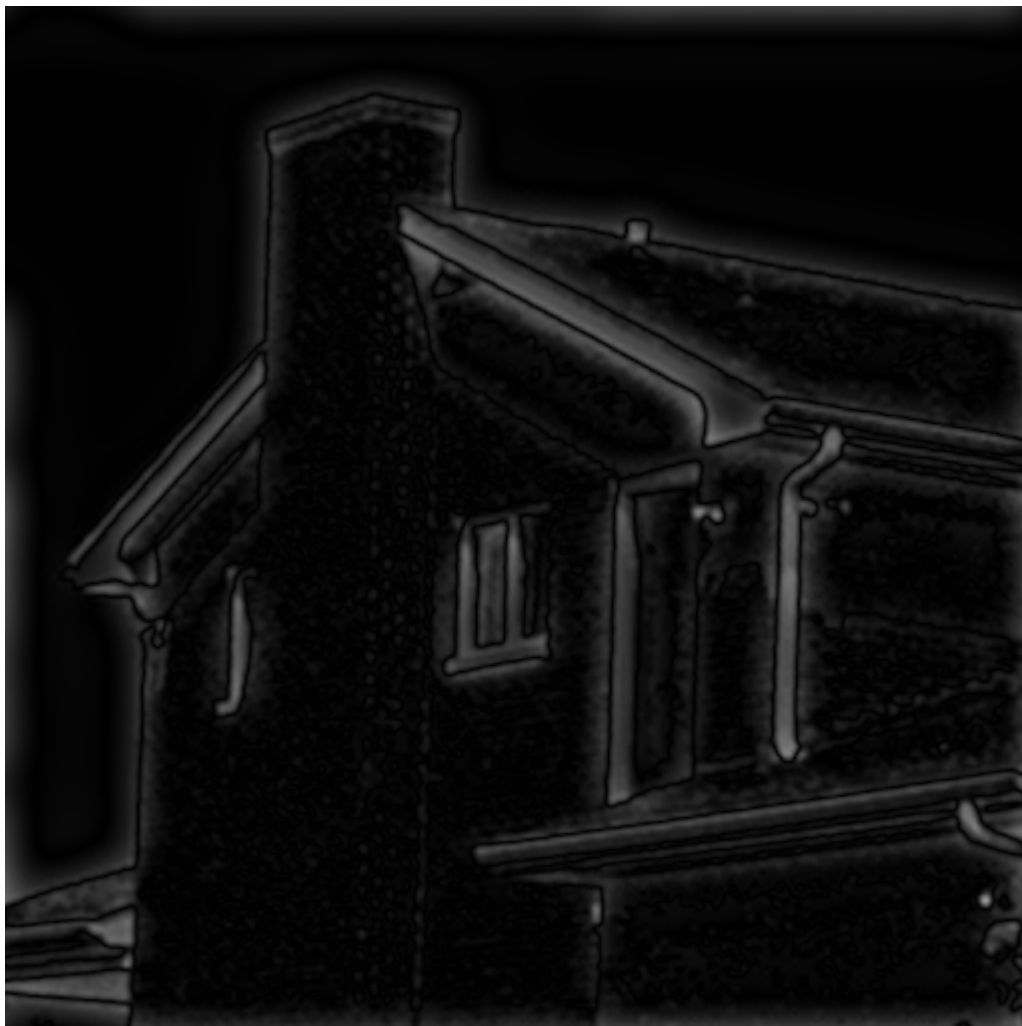


◦ 巴特沃斯：

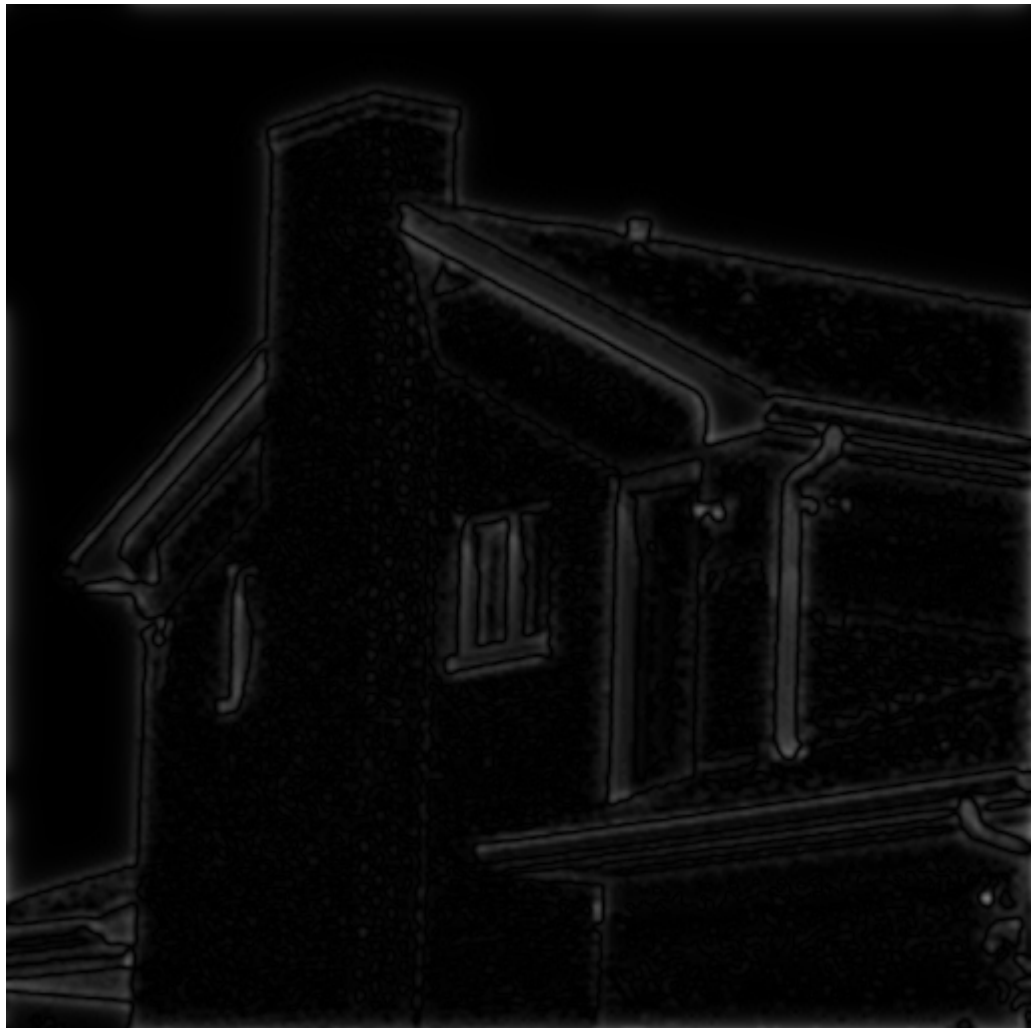


- $D0=20:n$

○ 高斯：

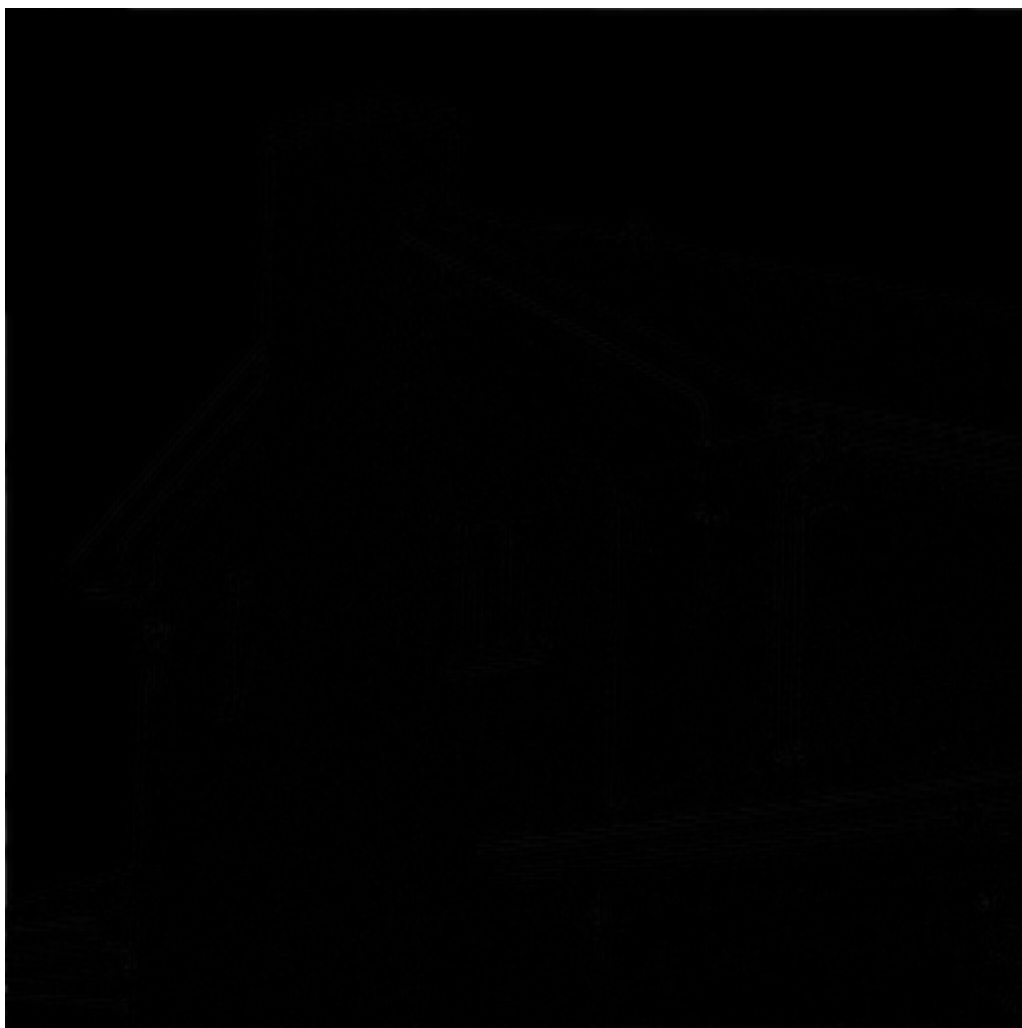


◦ 巴特沃斯：



• $D0=100$:

○ 高斯：



- 巴特沃斯：



结论：

D_0 相同时，高斯高通滤波器和二阶巴特沃斯高通滤波器的增强效果相似，都可以提取出图像边缘。 D_0 越小，两高通滤波器的增强的边缘越多，即滤波后图像边缘细节越多。同时，功率谱比也随 D_0 增大而变小。 D_0 相同时，高斯高通滤波器的功率谱比小于二阶巴特沃斯高通滤波器的功率谱比。

其他高通滤波器

要求

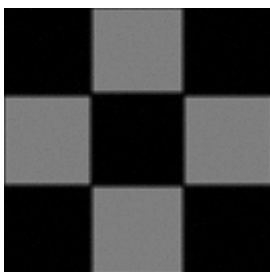
其他高通滤波器：拉普拉斯和Unmask，对测试图像test3,4滤波；分析各自优缺点；

实现

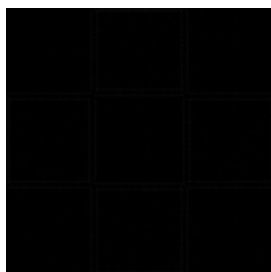
我使用python numpy库根据高斯噪声与butterworth定义完成实现。

结果展示

Unmask Test3-



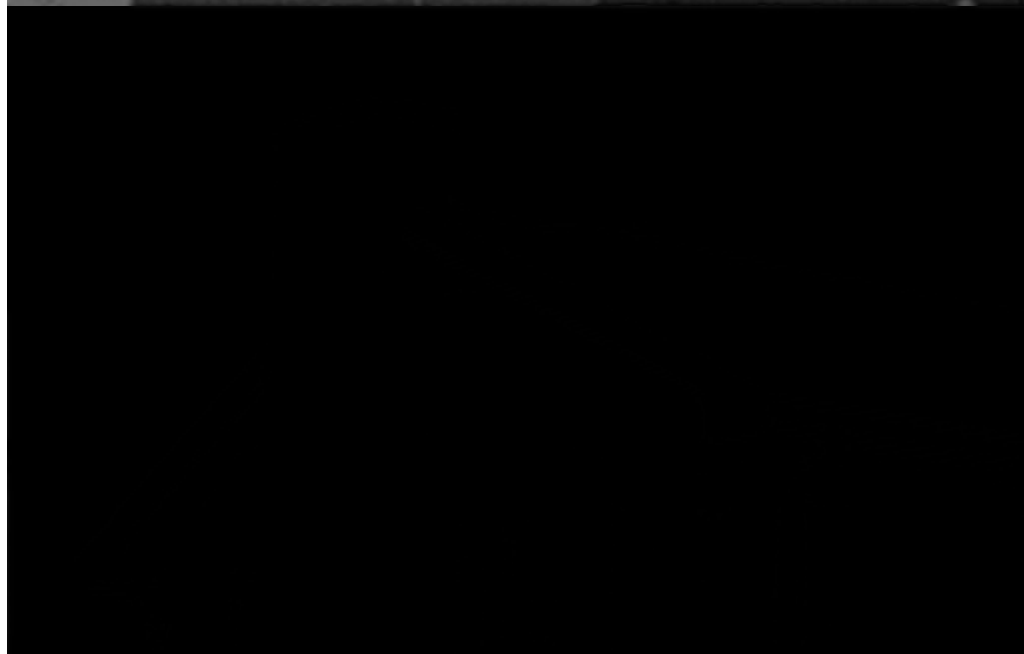
Laplace Test3-



Unmask Test4-



Laplace Test4-





结论：

在频域中拉普拉斯算子和钝化模板均可提取图像边缘。空域滤波与频域滤波都可以实现，对图像进行空域的处理时，每次只使用部分像素的性质。而进行频域处理时，是根据全局性质进行滤波处理。且频域处理时，频域高低对应图像变化的强弱，因此在某些方面频域处理更加直观。使用频域处理更为灵活，且更能体现图像本来的特征，空域原理简单更易懂。