

数字图像处理

第一次作业

姓名：何佳

班级：自动化 64

学号：2160700002

提交日期：2019.4.2

摘要：本次作业主要内容为频率域滤波，包括了 butterworth 低通、Gaussian 低通、butterworth 高通、Gaussian 高通、拉普拉斯高通和 Unsharp masking 高通，最后分析并讨论了空域低通高通滤波与频域低通和高通的关系。

一、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；

1.问题分析

1.1 频率域滤波

频率域滤波由修改一幅图像的傅里叶变换然后计算其反变换得到处理后的结果组成。由此，若给定一幅大小为 $M \times N$ 的数字图像 $f(x,y)$ ，则我们感兴趣的基本滤波公式有如下形式：

$$g(x,y) = \mathfrak{F}^{-1}\{H(u,v)F(u,v)\}$$

其中， $F(u,v)$ 是输入图像 $f(x,y)$ 的 DFT, $H(u,v)$ 是滤波函数(也简称为滤波器，或者滤波传递函数), $g(x,y)$ 是滤波后的(输出)图像。函数 F , H 和 g 是大小与输入图像相同的 $M \times N$ 阵列。

频率域滤波的步骤可以总结如下：

- 1) 给定一幅大小为 $M \times N$ 的输入图像 $f(x,y)$ ，根据 $P \geq 2M-1$ 和 $Q \geq 2N-1$ 的原则，得到填充参数 P 和 Q 。典型地，我们选择 $P = 2M$ 和 $Q = 2N$ 。
- 2) 对 $f(x,y)$ 添加必要数量的 0，形成大小为 $P \times Q$ 的填充后的图像 $fp(x,y)$ 。
- 3) 用 $(-1)^{x+y}$ 乘以 $fp(x,y)$ 移到其变换的中心。
- 4) 计算来自步骤 3 的图像的 DFT，得到 $F(u,v)$ 。
- 5) 生成一个实的、对称的滤波函数 $H(u,v)$ ，其大小为 $P \times Q$ ，中心在 $(P/2, Q/2)$ 处。用阵列相乘形成乘积 $G(u,v) = H(u,v)F(u,v)$ ；即 $G(i,k) = H(i,k)F(i,k)$ 。
- 6) 得到处理后的图像：

$$g_p(x,y) = \{\text{real}[\mathfrak{F}^{-1}[G(u,v)]]\}(-1)^{x+y}$$

其中，为忽略由于计算不准确导致的寄生复分量，选择了实部，下标 p 指出我们处理的是填充后的阵列。

- 7) 通过从 $g_p(x,y)$ 的左上象限提取 $M \times N$ 区域，得到最终处理结果 $g(x,y)$ 。

1.2 零填充

在 1.1 所述的步骤中，包括了对原始图像 $f(x,y)$ 进行零填充的操作步骤。这样做的原因是：我们处理的是二维的离散图像阵列，二维傅里叶变换及其反变换在 u 方向和 v 方向上均具有无限周期性，因此会导致卷积的缠绕错误。因此，根据奈奎斯特采样定理，我们需对 $M \times N$ 的原始图像阵列进行零填充，填充后图像大小为 $P \times Q$ ，且要求 $P \geq 2M-1$ 和 $Q \geq 2N-1$ 。一般来说，DFT 算法对偶数尺寸的阵列执行较快，因此最好选择 P 和 Q 为满足上面方程的最小偶整数。

1.3 butterworth 低通滤波器(BLPF)

截止频率位于距原点 D_0 处的 n 阶布特沃斯低通滤波器(BLPF)的传递函数定义为：

$$H(u,v) = \frac{1}{1 + [D(u,v) / D_0]^{2n}}$$

其中，

$$D(u, v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$

与 ILPF (理想低通滤波器) 不同, BLPF 传递函数并没有在通过频率和滤除频率之间给出明显截止的尖锐的不连续性。对于具有平滑传递函数的滤波器, 可在这样一点上定义截止频率, 即使得 $H(u, v)$ 下降到其最大值的某个百分比点, 这里定义截止频率点是当 $D(u, v) = D_0$ 时的点 (即 $H(u, v)$ 从其最大值 1 下降为 50%)。

1.4 Gaussian 滤波器

一维高斯低通滤波器(GLPF)的二维形式由下式给出:

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

上式中, $D(u, v)$ 是距频率矩形中心的距离。 D_0 是关于中心的扩展度的度量, 即截止频率。当 $D(u, v) = D_0$ 时, GLPF 下降到其最大值的 0.607 处。

另外, GLPF 的傅里叶反变换也是高斯的。这意味着通过对 $H(u, v)$ 进行 IDFT 得到的空间高斯滤波器将没有振铃现象。

1.5 功率谱比

功率谱定义为:

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

其中, R 和 I 分别是 $F(u, v)$ 的实部和虚部, 并且所有的计算直接对离散变量 $u=0, 1, 2, \dots, P-1$, $v=0, 1, 2, \dots, Q-1$ 进行。因此 $P(u, v)$ 是大小为 $M \times N$ 的阵列。实函数的傅里叶变换是共轭对称的, 这表明谱是关于原点偶对称的。如果 DFT 已被中心化, 那么原点位于频率矩形中心处, 半径为 D_0 的圆包含 $\alpha\%$ 的功率, 其中 α 的计算方式为:

$$\alpha = 100 \left[\sum_u \sum_v P(u, v) / P_T \right]$$

上式中,

$$P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u, v)$$

2. 编程思路

根据 1.3 和 1.4 中对 butterworth 和 Gaussian 低通滤波器的定义, 明确滤波器频域 $H(u, v)$; 根据 1.1 所述的频率域滤波步骤, 首先将原图像补零填充至 $P \times Q$, 然后采用 DFT 计算其傅里叶变换, 得到 $F(u, v)$, 在频率域将 $F(u, v)$ 和 $H(u, v)$ 进行对应点相乘, 并将结果通过 IDFT 变回空间域, 即得到最后的滤波结果。

根据 1.5 中的功率谱比计算方式, 分别对 $F(u, v)$ 和 $G(u, v)$ 每一个 (u, v) 处的值进行遍历, 计算其功率谱分量并求和, 最后两者做商得到功率谱比。

相关的 MATLAB 函数有:

- 1) fft2 函数: 用于数字图像的二维傅立叶变换;
- 2) ifft2 函数: 用于数字图像的二维傅立叶反变换;
- 3) fftshift 函数: 用于将变换后的图象频谱中心从矩阵的原点移到矩阵的中心。

4) imcrop 函数: imcrop(I,[a,b,c,d])表示将图像 I 从(a,b)处 (以左上角为 (0,0)) 开始, 裁剪 c×d 大小的图像。

3.处理结果

3.1 采用 butterworth 低通滤波进行平滑

1) test1.pgm

选取 N=2 阶 butterworth 低通滤波器的不同半径 D0, 得到的滤波结果如下:



N=2, D0=75 butterworth 低通滤波后的功率谱比为 0.9824=98.24%.

test1.pgm butterworth低通滤波器 N=2 D0=50最终结果



N=2, D0=50 butterworth 低通滤波后的功率谱比为 $0.9695=96.95\%$.

test1.pgm butterworth低通滤波器 N=2 D0=100最终结果



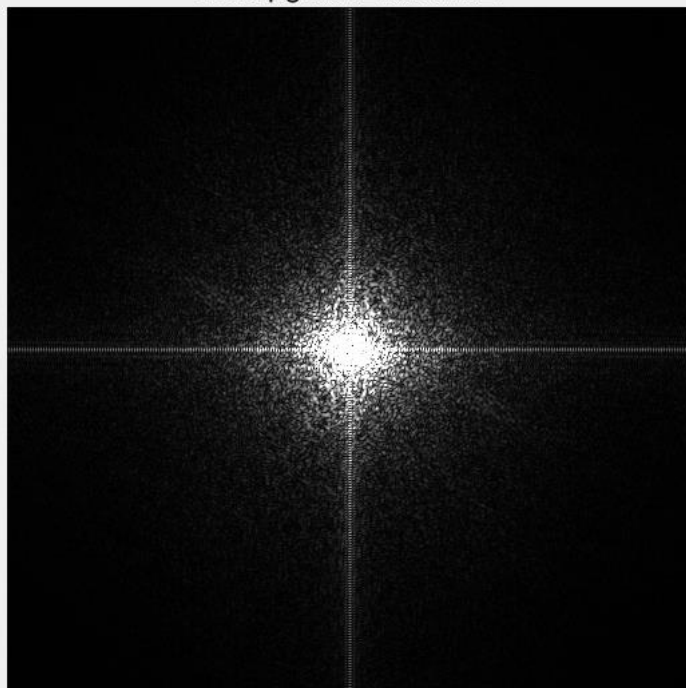
N=2, D0=100 butterworth 低通滤波后的功率谱比为 $0.9886=98.86\%$.

下面以 N=2, D0=75 为例, 给出在滤波过程中生成的中间图像:

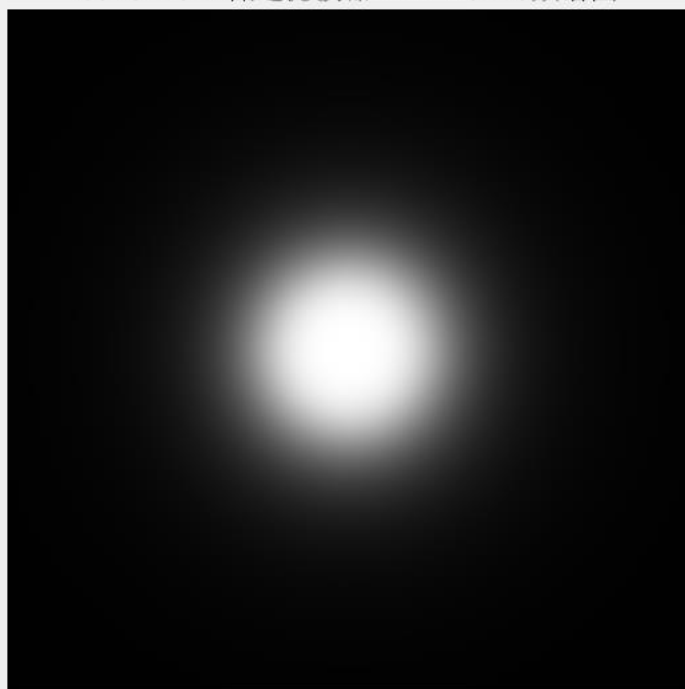
test1.pgm零填充后



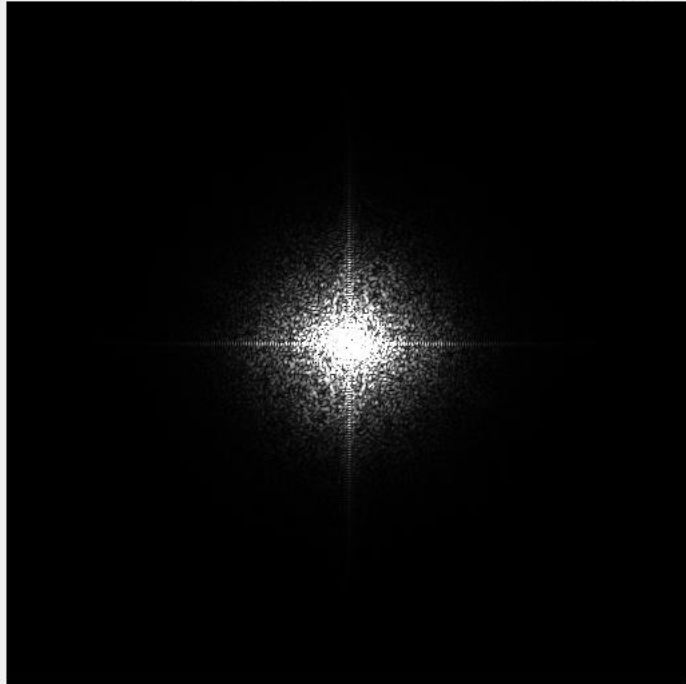
test1.pgm原图像频谱图



butterworth低通滤波器 $N=2$ $D_0=75$ 频谱图



butterworth低通滤波器 $N=2$ $D_0=75$ 滤波后频谱图



test1.pgm butterworth低通滤波器 $N=2$ $D_0=75$



最后裁剪左上角 $M \times N$ 部分的图像得到最终结果。

2) test2.tif

选取 $N=2$ 阶 butterworth 低通滤波器的不同半径 D_0 ，得到的滤波结果如下：

test2.tif原图像



test2.tif butterworth低通滤波器 N=2 D0=75最终结果



N=2, D0=75 butterworth 低通滤波后的功率谱比为 0.9824=98.24%.

test2.tif butterworth低通滤波器 N=2 D0=50最终结果



N=2, D0=50 butterworth 低通滤波后的功率谱比为 $0.9747=97.47\%$.

test2.tif butterworth低通滤波器 N=2 D0=100最终结果



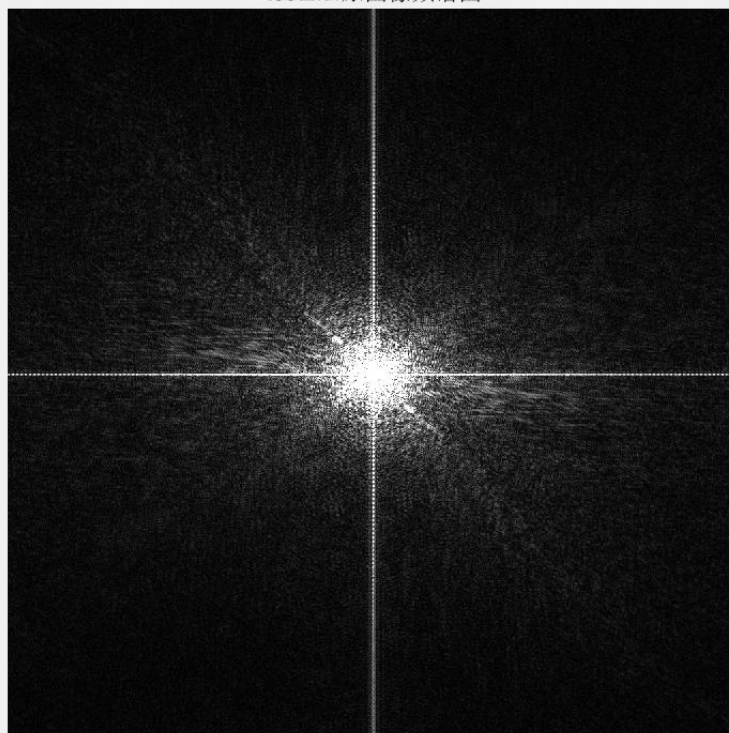
N=2, D0=100 butterworth 低通滤波后的功率谱比为 $0.9864=98.64\%$.

下面以 N=2, D0=75 为例, 给出在滤波过程中生成的中间图像:

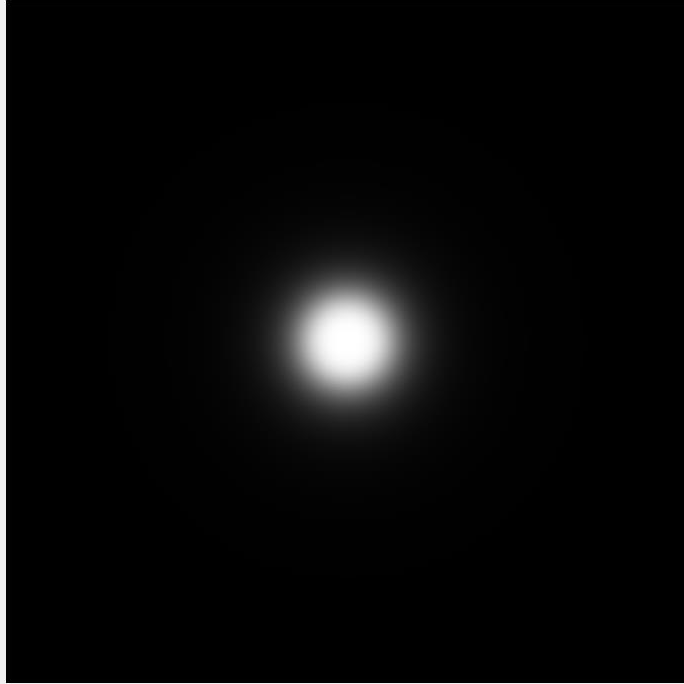
test2.tif零填充后



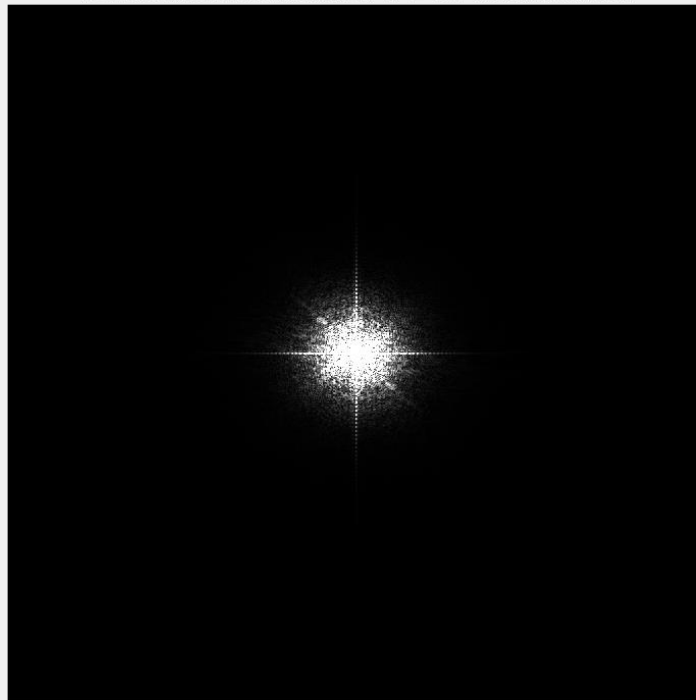
test2.tif原图像频谱图

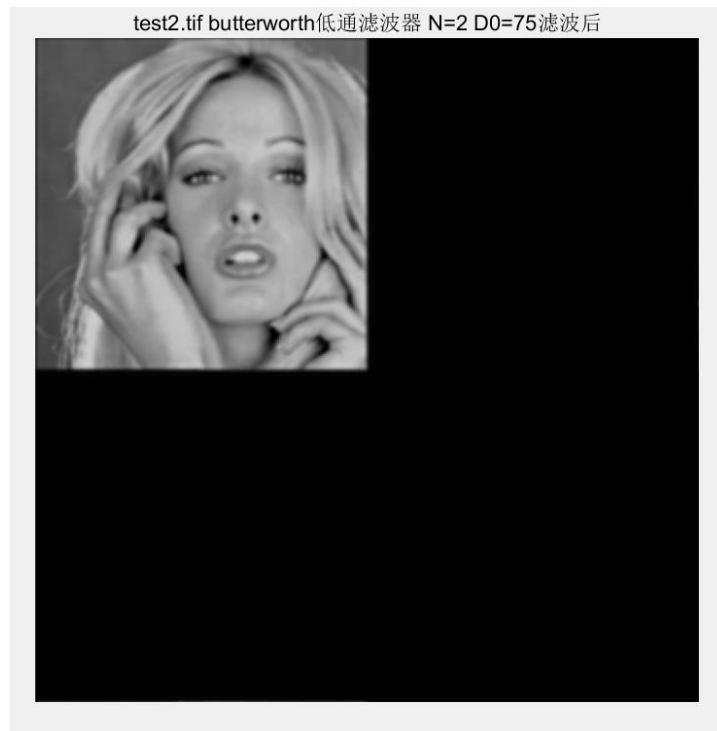


butterworth低通滤波器 N=2 D0=75频谱图



test2.tifbutterworth低通滤波器 N=2 D0=75滤波后频谱图





最后裁剪左上角 $M \times N$ 部分的图像得到最终结果。

3.2 采用高斯低通滤波器进行平滑

1) test1.pgm

分别采用不同半径 D_0 的高斯低通滤波器对 test1 进行处理：



test1.pgm 高斯滤波器 D0=75最终结果



D0=75 时，功率谱比为：0.9763=97.63%

test1.pgm 高斯低通滤波器 D0=50最终结果



D0=50 时，功率谱比为：0.9614=96.14%

test1.pgm 高斯低通滤波器 D0=100最终结果



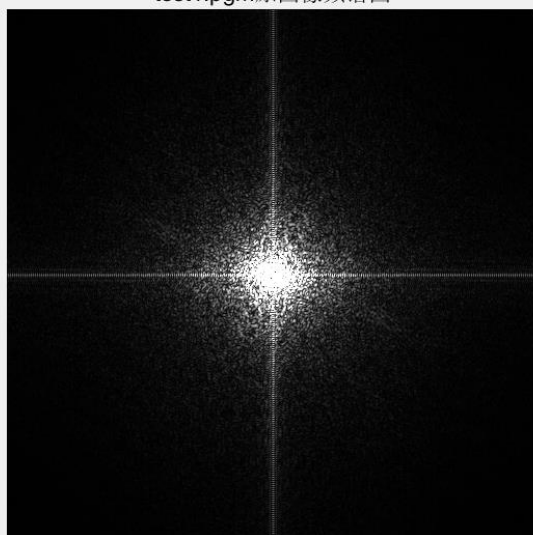
D0=100 时，功率谱比为：0.9837=98.37%

下面以 D0=75 高斯低通滤波为例，给出在滤波过程中生成的中间图像：

test1.pgm零填充后



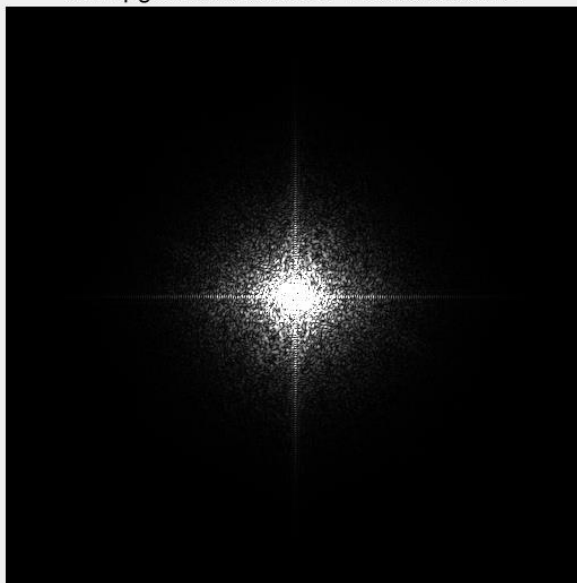
test1.pgm原图像频谱图



高斯滤波器 $D0=75$ 频谱图



test1.pgm 高斯滤波器 $D0=75$ 滤波后频谱图

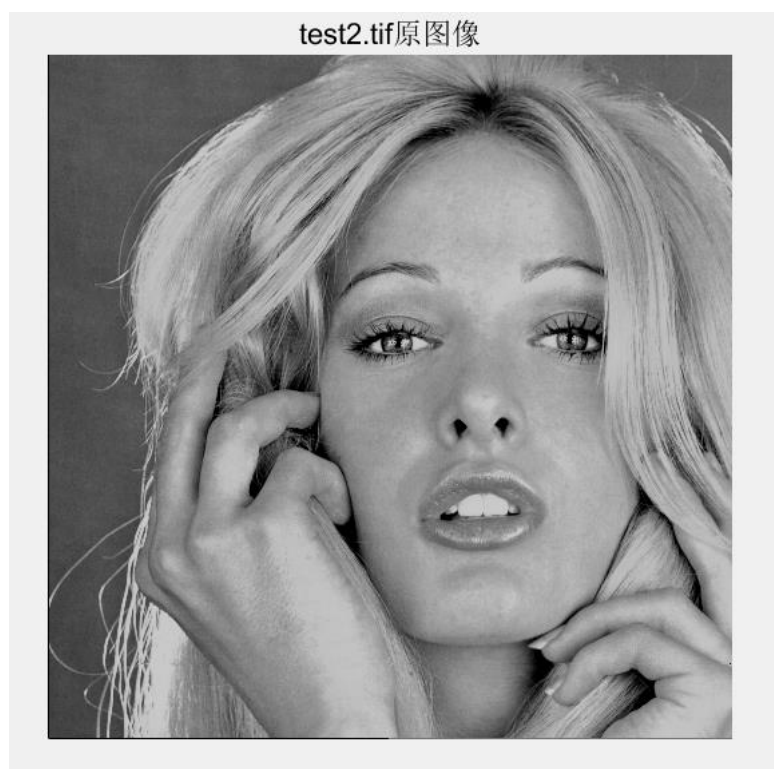




最后裁剪左上角 $M \times N$ 部分的图像得到最终结果。

2) test2.tif

采用不同 $D0$ 的高斯低通滤波器对 test2.tif 进行滤波的结果如下：



test2.tif 高斯滤波器 D0=75最终结果



D0=75 时，功率谱比为 $0.9789=97.89\%$.

test2.tif 高斯低通滤波器 D0=50最终结果



D0=50 时，功率谱比为 $0.9699=96.99\%$.

test2.tif 高斯低通滤波器 $D_0=100$ 最终结果



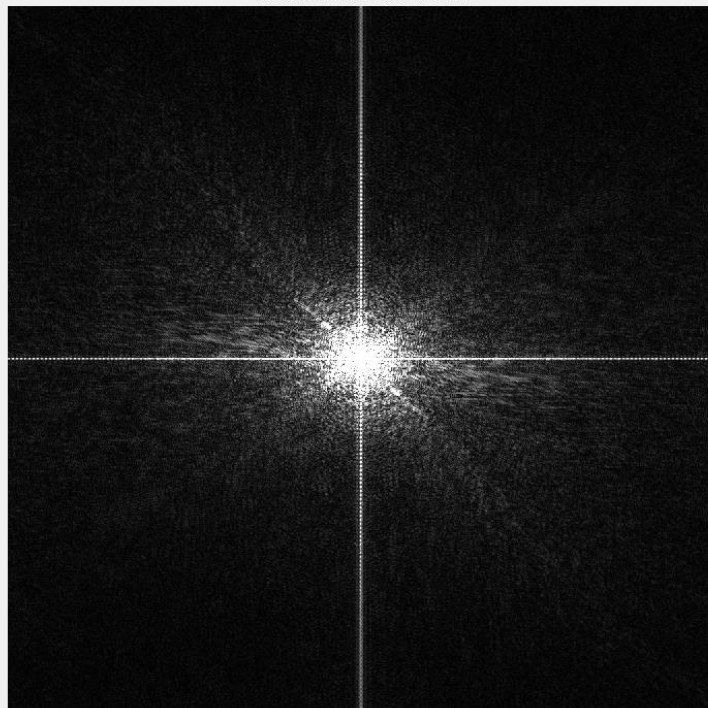
$D_0=100$ 时，功率谱比为 $0.9789=97.89\%$.

下面以 $D_0=75$ 的高斯低通滤波器为例，给出滤波过程中生成的中间图像：

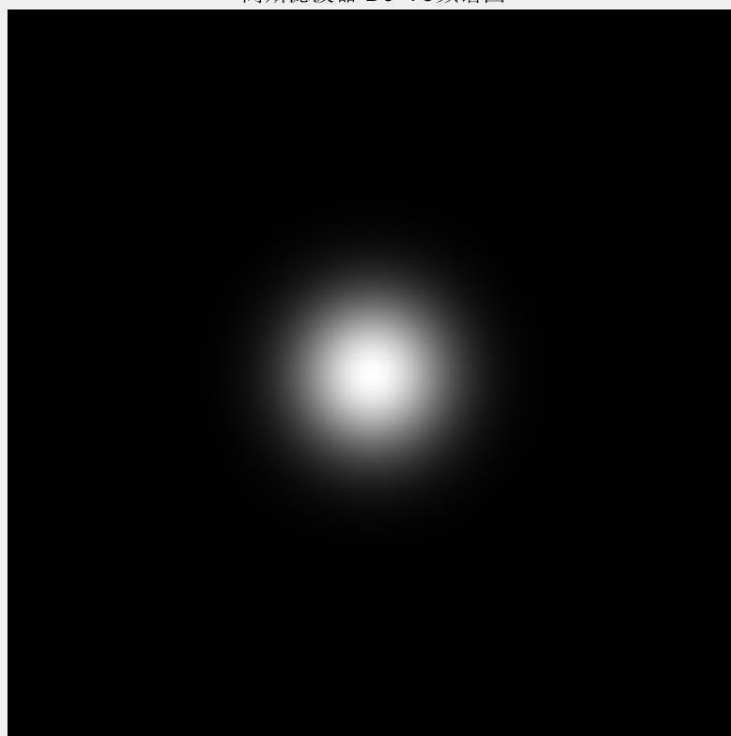
test2.tif零填充后

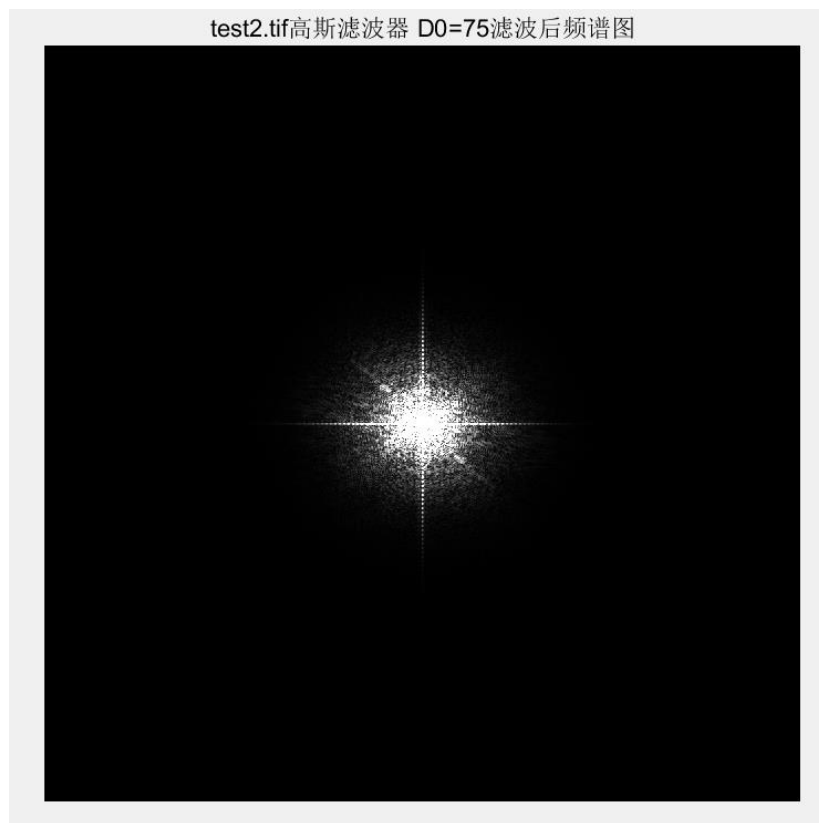


test2.tif原图像频谱图



高斯滤波器 $D_0=75$ 频谱图





最后裁剪左上角 $M \times N$ 部分的图像得到最终结果。

4.结果分析与讨论

上述结果中，不论采用的是 butterworth 低通滤波器还是高斯低通滤波器，都有滤波半

径 D_0 越小，滤波后的图像越模糊，功率谱比越小的结论。这是因为随着 D_0 的减小，频域低通截止频率越小，更多的高频信息被滤除，图像更加模糊，而图像整体功率得到衰减的部分也越多，因此相对于原图像，功率谱比也越小。

另外，对于 D_0 相同的二阶 butterworth 低通滤波器和高斯低通滤波器，在 D_0 较小时，前者的滤波结果比后者清晰一些，功率谱比略大，说明此时前者频谱函数高于后者。当 D_0 增大到某一值时，两者滤波效果完全相同，此时两个滤波器的二维频谱函数相交。当 D_0 更大时，前者的得到的滤波结果比后者模糊一些，功率谱比略小，说明此时前者频谱函数低于后者。

二、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；

1.问题分析

1.1 使用频率域高通滤波器锐化图像

图像的边缘和其他灰度的急剧变化与高频分量有关，高通滤波器衰减傅里叶变换中的低频分量而不会扰乱高频信息，所以图像的锐化可在频率域通过高通滤波来实现。

一个高通滤波器是从给定的低通滤波器用下式得到：

$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$

其中， $H_{LP}(u, v)$ 是低通滤波器的传递函数。也就是说，被低通滤波器衰减的频率能通过高通滤波器，反之亦然。

进行频率域高通滤波的步骤与低通滤波基本一致，包括对原图像进行零填充并平移至变换中心、傅里叶变换、与滤波器频谱函数在频域相乘、傅里叶反变换以及最后裁剪图像。另外，功率谱的定义及功率谱比的计算方式与前述相同。

1.2 butterworth 高通滤波器(BHPL)

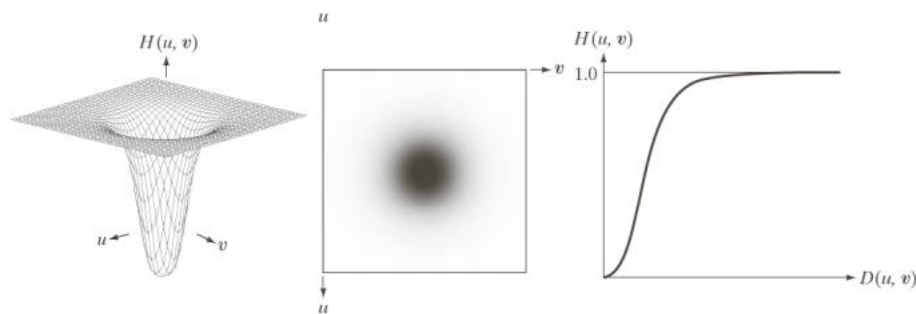
截止频率为 D_0 的 n 阶 butterworth 高通滤波器（BHPF）定义为：

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

其中

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

上式中， D_0 为截止频率点，在该点， $H(u, v)$ 下降至最大值的 50%。



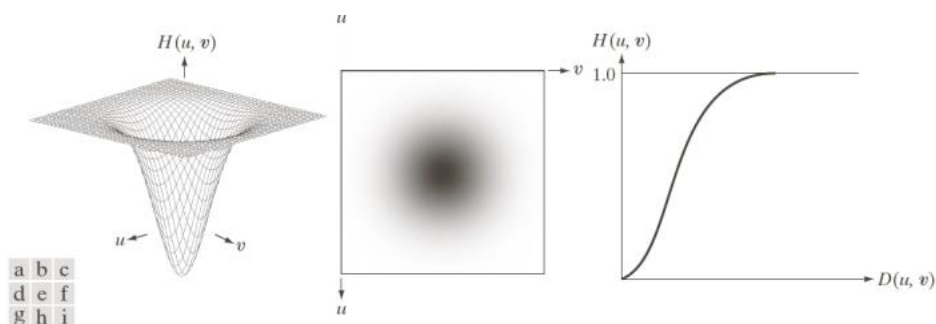
1.3 Gaussian 高通滤波器

截止频率处在距频率矩形中心距离为 D_0 的高斯高通滤波器 (GHPF) 的传递函数由下式给出:

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$

其中 $D(u, v)$ 是距离频率域矩形中心的距离。 D_0 是截止频率, 当 $D(u, v) = D_0$ 时, GHPF 下降到其最大值的 0.607 处。 $D(u, v)$ 由下式给出:

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$



2.编程实现

其编程思路同频率域低通滤波, 即首先进行零填充, 然后对图像矩阵的每一个元素乘以 $(-1)^{x+y}$, 使其在频域平移至变换中心, 然后对处理后的图像阵列进行 DFT 并在频域与滤波器传递函数 $H(u, v)$ 进行相乘, 接着对相乘后的结果进行 IDFT, 然后将该高通滤波后的图像通过一个比例因子 K 与原图像相加, 所得即为高通锐化结果, 最后裁剪掉零填充的部分, 得到高通滤波后的图像。

功率谱的计算仍然是根据功率谱比计算方式, 分别对 $F(u, v)$ 和 $G(u, v)$ 每一个 (u, v) 处的值进行遍历, 计算其功率谱分量并求和, 最后两者做商得到功率谱比。

使用到的 Matlab 函数有:

- 1) fft2 函数: 用于数字图像的二维傅立叶变换;
- 2) ifft2 函数: 用于数字图像的二维傅立叶反变换;
- 3) fftshift 函数: 用于将变换后的图象频谱中心从矩阵的原点移到矩阵的中心。
- 4) imcrop 函数: `imcrop(I,[a,b,c,d])` 表示将图像 I 从 (a, b) 处 (以左上角为 $(0, 0)$) 开始, 裁剪 $c \times d$ 大小的图像。

3.处理结果

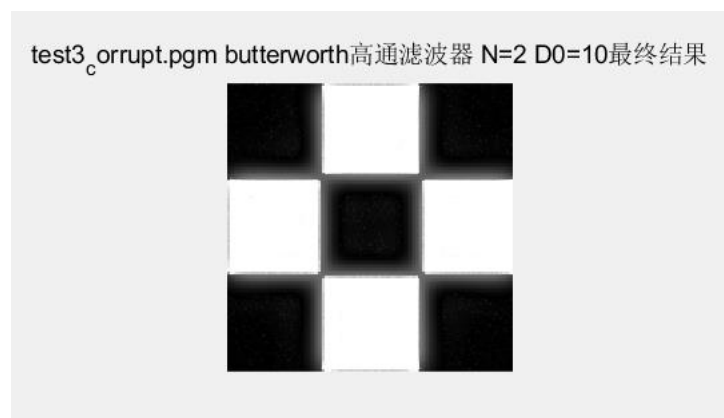
3.1 用 butterworth 高通滤波器进行锐化

1) test3_corrupt.pgm

采用 $N=2$ 的 butterworth 高通滤波器进行高通滤波, 通过分别设置不同的滤波半径 D_0 , 得到如下不同结果:

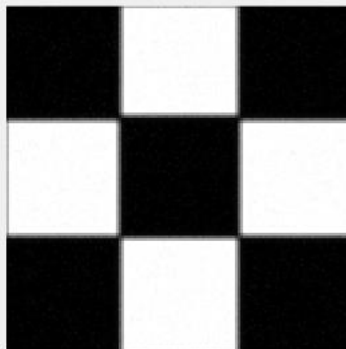


$N=2$, $D_0=20$ 时的功率谱比为: $0.0260=2.60\%$



$N=2$, $D_0=10$ 时的功率谱比为: $0.0770=7.70\%$

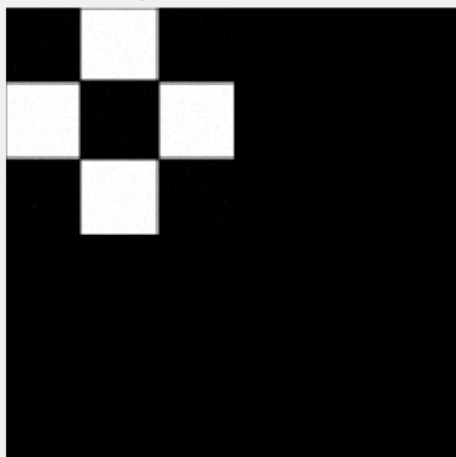
test3_c_orrupt.pgm butterworth高通滤波器 N=2 D0=100最终结果



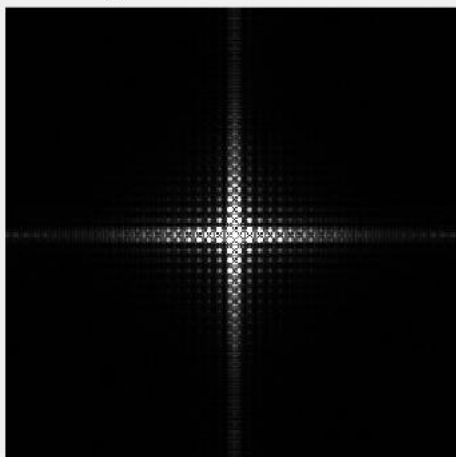
N=2, D0=100 时的功率谱比为: $4.1523 \times 10^{-4} = 0.04\%$

下面以 N=2, D=20 的 butterworth 高通滤波器, 锐化比例因子 K=5 为例给出滤波过程中生成的中间图像:

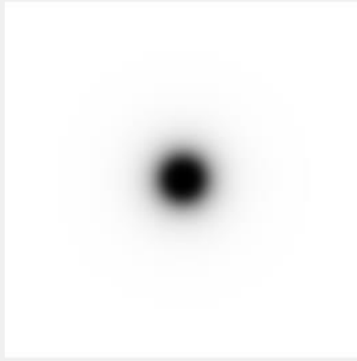
test3_c_orrupt.pgm零填充后



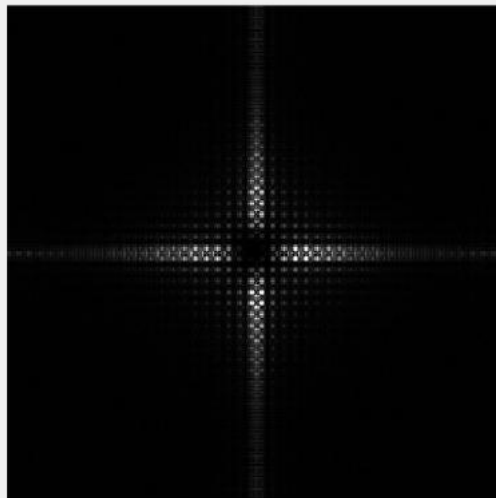
test3_c_orrupt.pgm原图像频谱图



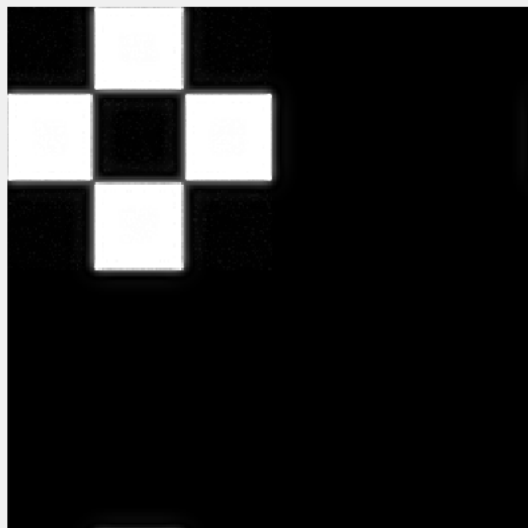
butterworth高通滤波器 N=2 D0=20 频谱图



test3_corrupt.pgm butterworth高通滤波器 N=2 D0=20 滤波后频谱图



test3_corrupt.pgm butterworth高通滤波器 N=2 D0=20 滤波后



最后裁剪左上角 $M \times N$ 部分的图像即得最终结果。

2) test4 copy.bmp

采用 $N=2$ 的 butterworth 高通滤波器进行高通滤波，分别选取 $D_0=40,80,120$ 的滤波半径进行滤波，结果如下：

test4 copy.bmp原图像



test4 copy.bmp butterworth高通滤波器 $N=2$ $D_0=80$ 最终结果



$N=2$, $D_0=80$ 时的功率谱比为: $0.0073=0.73\%$

test4 copy.bmp butterworth高通滤波器 N=2 D0=40最终结果



N=2, D0=40 时的功率谱比为: $0.0190=1.90\%$

test4 copy.bmp butterworth高通滤波器 N=2 D0=120最终结果



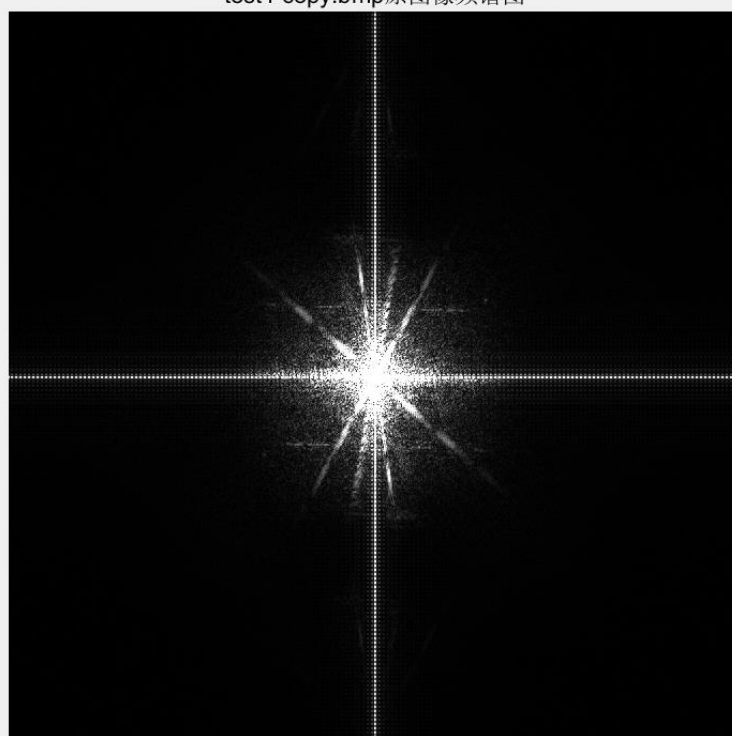
N=2, D0=120 时的功率谱比为: $0.0040=0.40\%$

下面以使用 N=2, D0=80 的 butterworth 高通滤波器为例给出滤波过程中生成的中间图像:

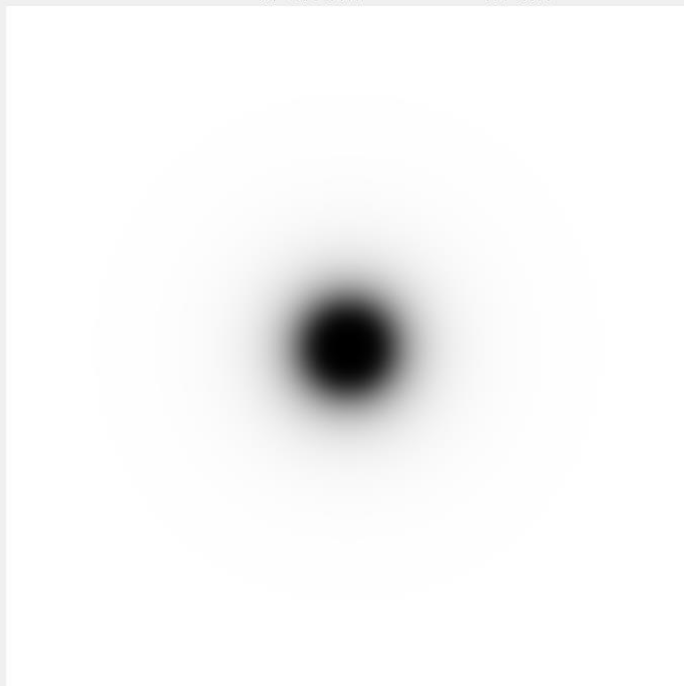
test4 copy.bmp零填充后



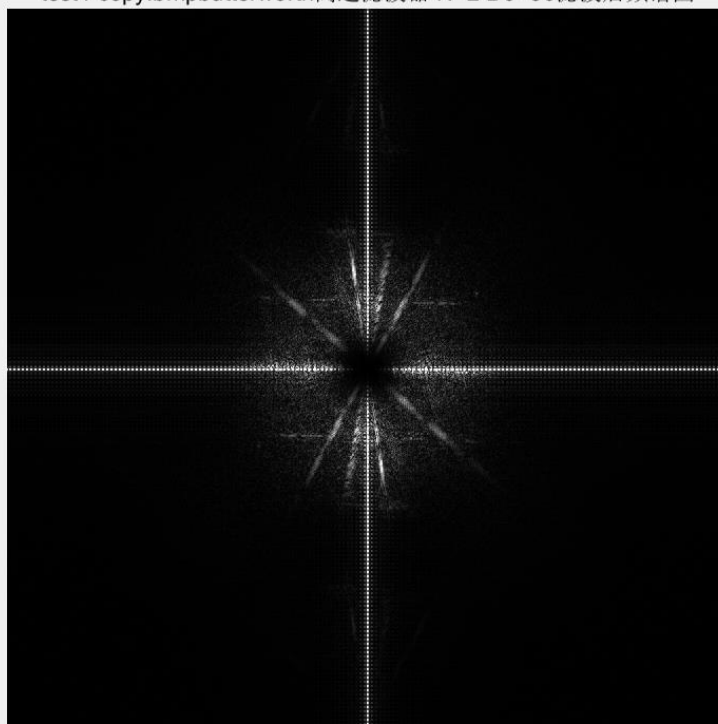
test4 copy.bmp原图像频谱图

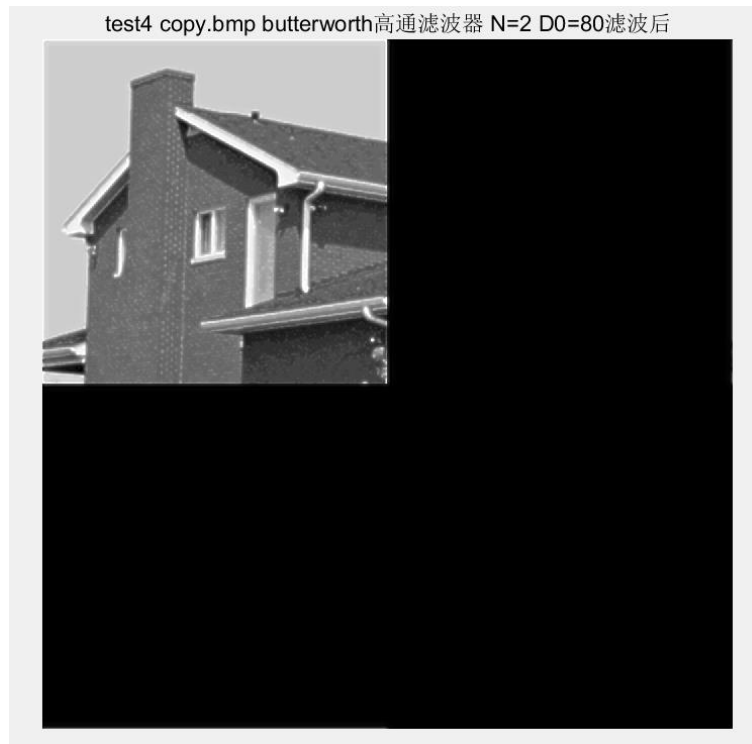


butterworth高通滤波器 N=2 D0=80频谱图



test4 copy.bmpbutterworth高通滤波器 N=2 D0=80滤波后频谱图



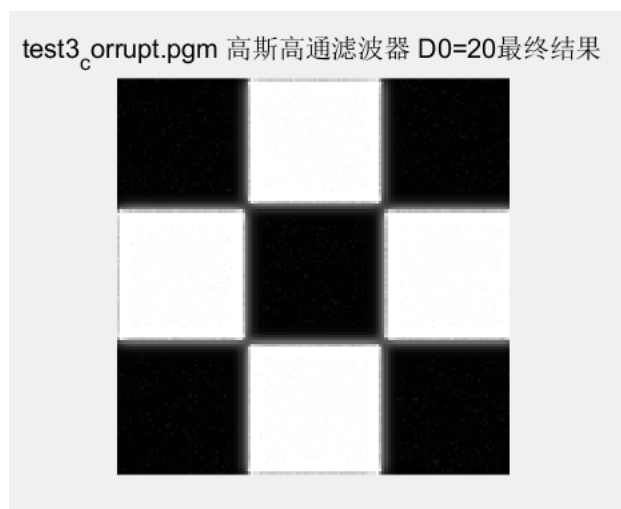
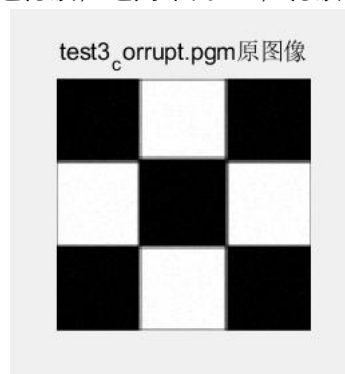


最后裁剪左上角 $M \times N$ 部分即得最终结果。

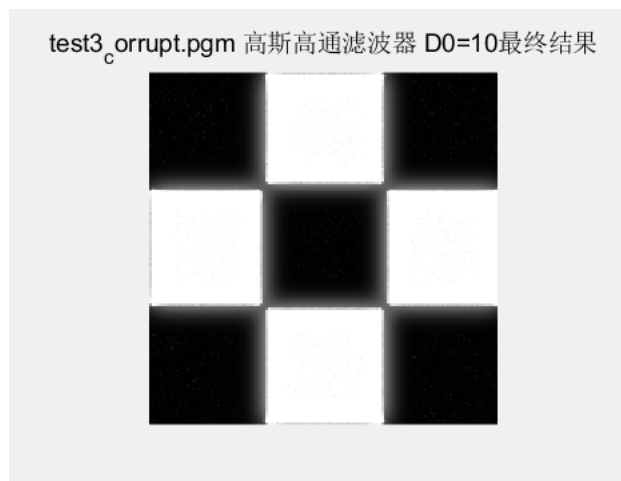
3.2 用高斯高通滤波器进行锐化

1) test3_corrupt.pgm

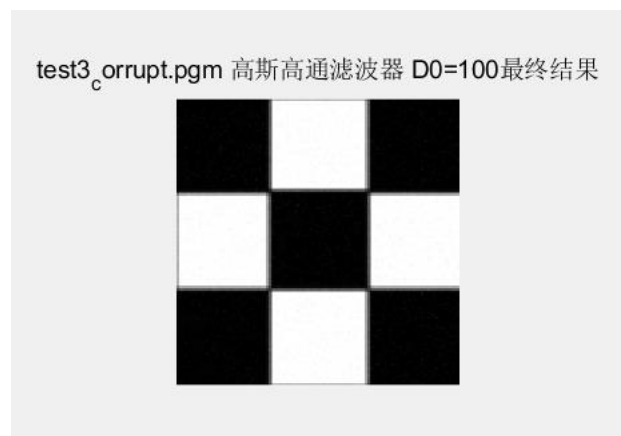
采用高斯高通滤波器进行高通滤波，选用不同 D_0 ，滤波结果如下：



D0=20 时的功率谱比为：0.0210=2.10%

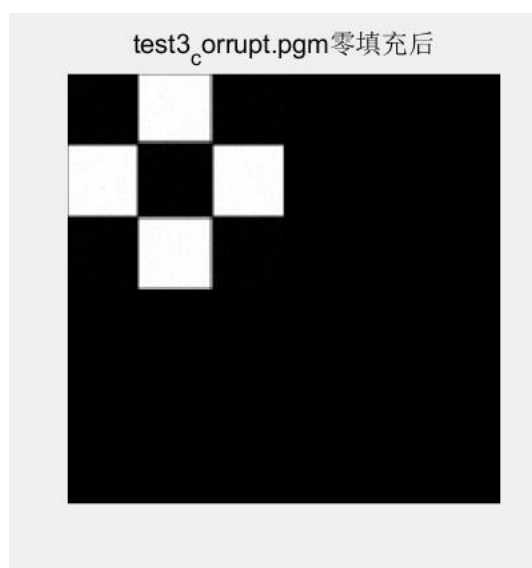


D0=10 时的功率谱比为：0.0650=6.50%

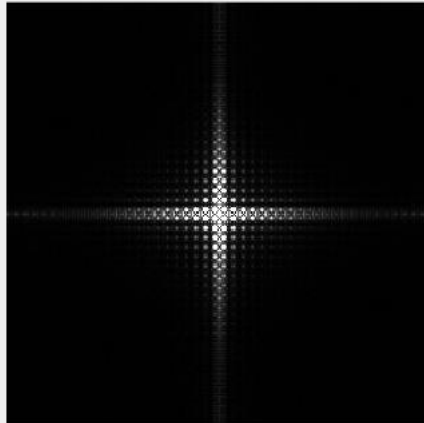


D0=100 时的功率谱比为：3.9583e-04=0.03%

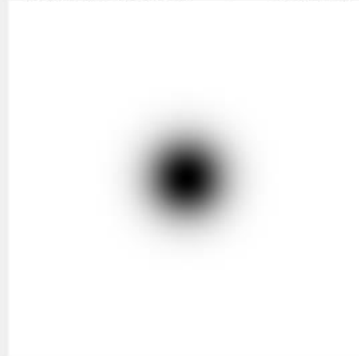
下面以 D=20 的高斯高通滤波器为例给出滤波过程中生成的中间图像：



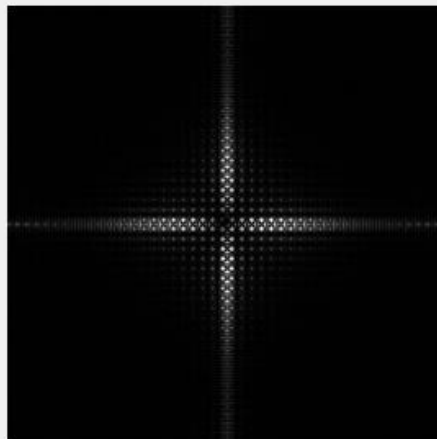
test3_orrupt.pgm原图像频谱图



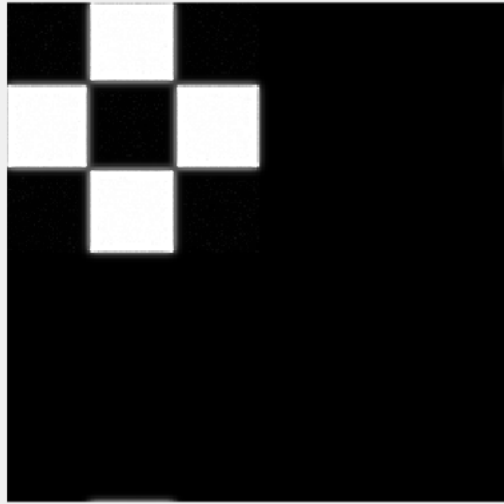
高斯高通滤波器 $D_0=20$ 频谱图



test3_orrupt.pgm高斯高通滤波器 $D_0=20$ 滤波后频谱图



test3_orrupt.pgm 高斯高通滤波器 $D_0=20$ 滤波后



2) test4 copy.bmp

采用高斯高通滤波器进行高通滤波，选用不同的滤波半径 D_0 ，结果如下：

test4 copy.bmp原图像



test4 copy.bmp 高斯高通滤波器 D0=80最终结果



D0=80 时的功率谱比为: $0.0062=0.62\%$

test4 copy.bmp 高斯高通滤波器 D0=40最终结果



D0=40 时的功率谱比为: $0.0160=1.60\%$

test4 copy.bmp 高斯高通滤波器 $D_0=120$ 最终结果



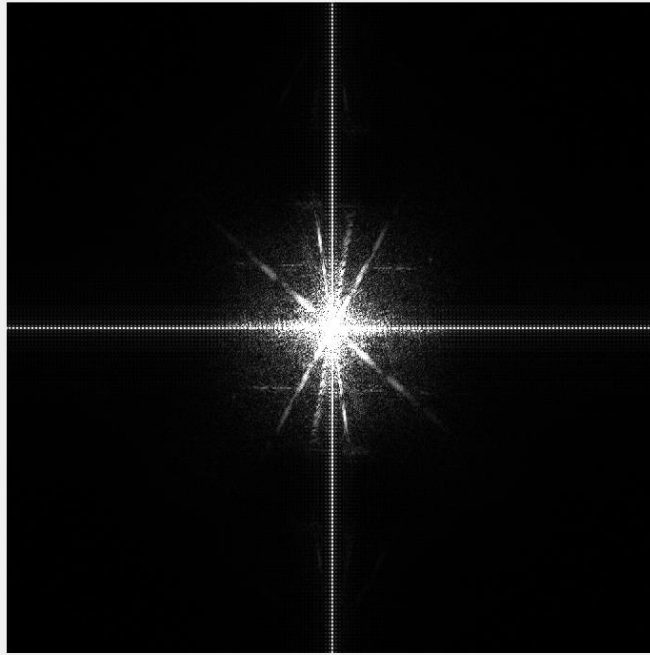
$D_0=120$ 时的功率谱比为: $0.0034=0.34\%$

下面以 $D=80$ 的高斯高通滤波器为例给出滤波过程中生成的中间图像:

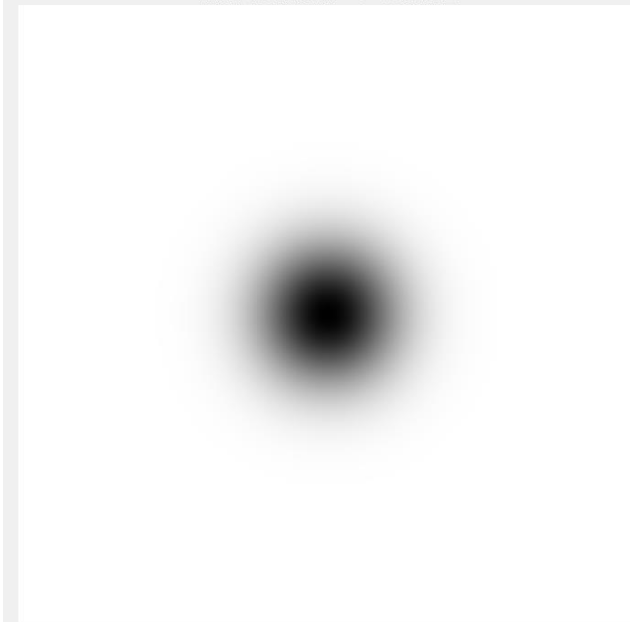
test4 copy.bmp零填充后

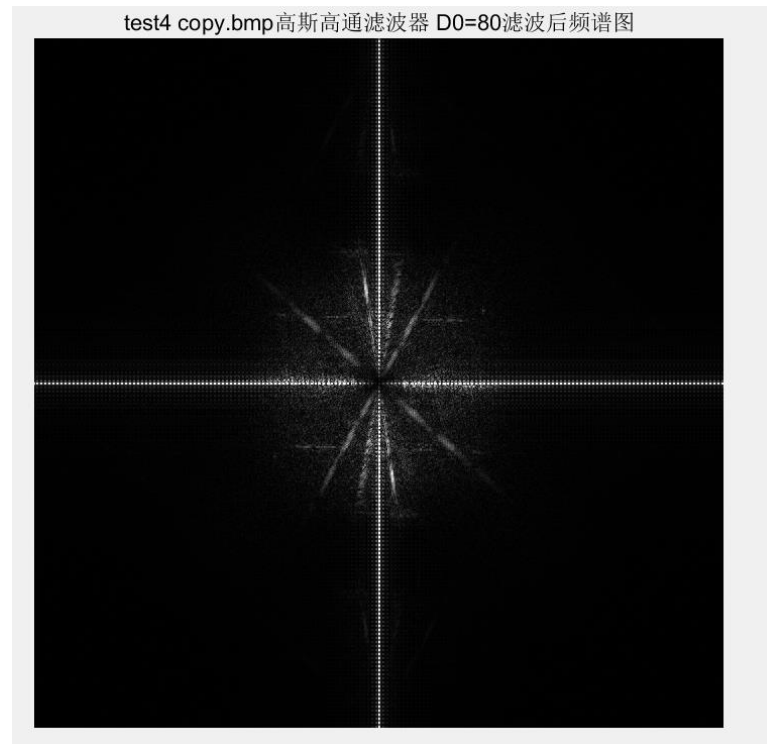


test4 copy.bmp原图像频谱图



高斯高通滤波器 $D_0=80$ 频谱图





最后裁剪左上角的 $M \times N$ 区域即得最终处理结果。

4.结果分析与讨论

上述结果中，不论采用的是 butterworth 高通滤波器还是高斯高通滤波器，都有滤波半径 D_0 越小，处理后的图像边缘增强越明显，功率谱比越大的结论。这是因为随着 D_0 的减

小，频域高通截止频率越小，高频信息保留更多，滤波后的边缘更加明显，因而叠加在原图像后的增强效果更明显，另外因为高频部分的边缘保留更多，因此相对于原图像，功率谱比也越大。

对比二阶 butterworth 高通滤波器和高斯高通滤波器，两个滤波器达到的图像增强效果基本一致。

三、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；比较并讨论空域低通高通滤波 (Project3) 与频域低通和高通的关系；

1. 问题分析

1.1 频率域滤波

频率域滤波原理及步骤与之前相同。

1.2 频率域的拉普拉斯算子

拉普拉斯算子关于频率矩形的中心可使用如下滤波器在频率域实现：

$$\begin{aligned} H(u, v) &= -4\pi^2 \left[(u - P/2)^2 + (v - Q/2)^2 \right] \\ &= -4\pi^2 D^2(u, v) \end{aligned}$$

其中， $D(u, v)$ 是距离函数，表达式如下：

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

然后，拉普拉斯图像由下式得到：

$$\nabla^2 f(x, y) = \mathfrak{F}^{-1} \{ H(u, v) F(u, v) \}$$

其中， $F(u, v)$ 是 $f(x, y)$ 的傅里叶变换。增强可使用下式实现：

$$g(x, y) = f(x, y) + c \nabla^2 f(x, y) \quad c = -1$$

这里， $c = -1$ ，因为 $H(u, v)$ 是负的。由于计算 $\nabla^2 f(x, y)$ 会引入已标定系数的 DFT，这些系数的幅度与 f 的最大值相比可能要大几个量级。这样， f 与其拉普拉斯间的差别一定会被带入可比的范围。处理该问题的最容易的方法是在计算 $f(x, y)$ 的 DFT 之前将 $f(x, y)$ 的值归一化到区间 $[0, 1]$ 内，并用它的最大值除 $\nabla^2 f(x, y)$ ，从而将它带到近似区间 $[-1, 1]$ 内，然后利用上式进行计算。

在频率域，上式可写为：

$$\begin{aligned}
g(x, y) &= \mathfrak{F}^{-1} \{ F(u, v) - H(u, v)F(u, v) \} \\
&= \mathfrak{F}^{-1} \{ [1 - H(u, v)] F(u, v) \} \\
&= \mathfrak{F}^{-1} \left\{ \left[1 + 4\pi^2 D^2(u, v) \right] F(u, v) \right\}
\end{aligned}$$

1.3 Unsharp masking 钝化模板

钝化模板定义如下：

$$g_{mask}(x, y) = f(x, y) - f_{LP}(x, y)$$

其中

$$f_{LP}(x, y) = \mathfrak{F}^{-1} [H_{LP}(u, v)F(u, v)]$$

其中， $H_{LP}(u, v)$ 是一个低通滤波器， $F(u, v)$ 是 $f(x, y)$ 的傅里叶变换。这里， $f_{LP}(x, y)$ 是平滑后的图像。然后：

$$g(x, y) = f(x, y) + k * g_{mask}(x, y)$$

上式中， $K=1$ 时为钝化模板， $K>1$ 时为高提升模板。

2.编程实现

2.1 拉普拉斯模板高通滤波

采用如下所示的拉普拉斯模板进行高通滤波：

1	1	1
1	-8	1
1	1	1

将原图像与 laplace 算子进行卷积。卷积运算的 matlab 函数用法如下：

`C=conv2(A,B,shape);`

其中，A:输入图像，B:卷积核，C：输出图像

假设输入图像 A 大小为 $m_a \times n_a$ ，卷积核 B 大小为 $m_b \times n_b$ ，则：当 $shape=full$ 时，返回全部二维卷积结果，即返回 C 的大小为 $(m_a+m_b-1) \times (n_a+n_b-1)$ ； $shape=same$ 时，返回与 A 同样大小的卷积中心部分； $shape=valid$ 时，不考虑边界补零，即只要有边界补出的零参与运算的都舍去，返回 C 的大小为 $(m_a-m_b+1) \times (n_a-n_b+1)$

2.2 Unsharp masking 高通滤波

钝化模板选用高斯低通滤波器，编程思路与问题一和问题二基本一致，在高斯低通滤波之后将图像按照：

$$g_{mask}(x, y) = f(x, y) - f_{LP}(x, y)$$

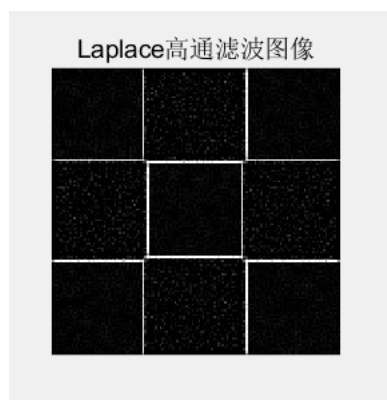
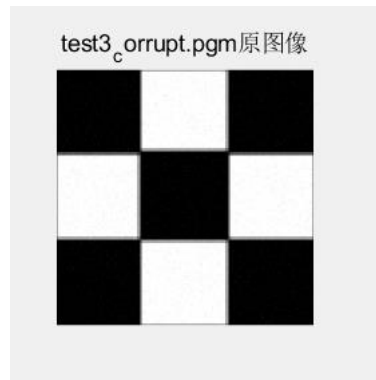
和

$$g(x, y) = f(x, y) + k * g_{mask}(x, y)$$

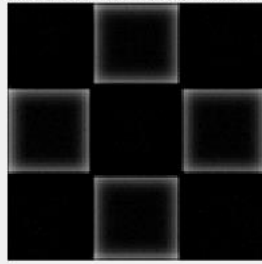
进行处理，输出即为 Unsharp masking 增强后的图像。上式中 K 取 1.

3.处理结果

1) test3_corrupt.pgm



Unsharp Masking高通滤波图像



Unsharp Masking锐化增强后的图像

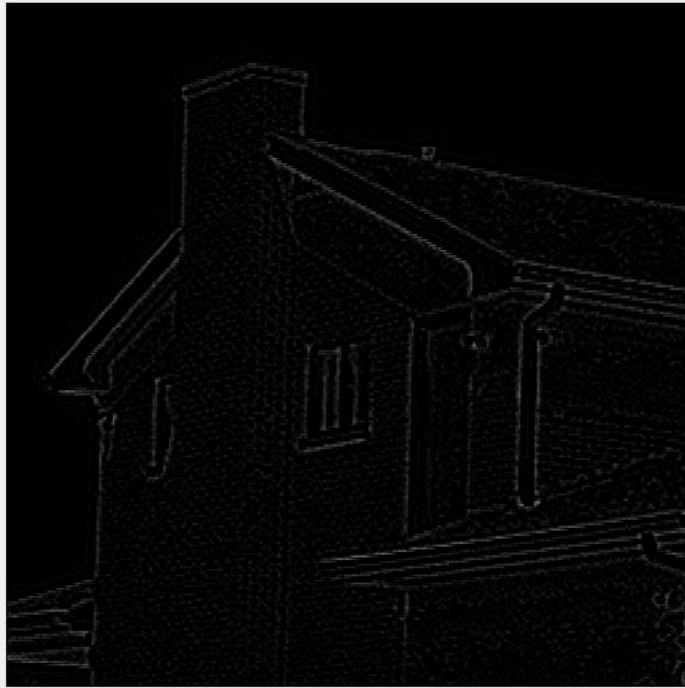


2) test4 copy.bmp

test4 copy.bmp原图像



Laplace高通滤波图像



Laplace锐化增强后的图像



Unsharp Masking高通滤波图像



Unsharp Masking锐化增强后的图像



4.结果分析与讨论

4.1 拉普拉斯和 Unmask 各自优缺点

上述结果分别采用 $Laplace = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ 的拉普拉斯模板和以滤波半径 $D0=10$ 的高斯低通滤波器为钝化模板的 Unsharp Masking 的结果，可以看出，前者滤波后保留的细节信息更多，因此增强后的图像看起来有重影，而后者则是越高频的部分保留越多，但细节保

留较差，因此增强后的结果中，边缘处的亮度很高，其他部分基本与原图一致。

4.2 空域低通高通滤波与频域低通和高通的关系

空域图像矩阵经过傅里叶变换进入频域，频域图像矩阵经过傅里叶反变换进入空域。另外，卷积定理阐明了空域和频域的转换关系，即空域的卷积等于频域的相乘，反之亦然。因此，空域滤波可转换到频域进行处理，反正亦然。对于图像处理来说，对于相同处理目的，空域与频域滤波的结果往往是相同的，但在频域中分析图像的频率成分与图像的视觉效果间的对应关系更直观。空域滤波在具体实现上和硬件设计上也有一定优点。例如，空域技术中无论使用点操作还是模板操作，每次都只是基于部分像素的性质，而频域技术每次都利用图像中所有像素的数据，具有全局性，有可能更好地体现图像的整体特性，如整体对比度和平均灰度值等。

参考文献

<https://blog.csdn.net/celerychen2009/article/details/38852105>

http://blog.sina.com.cn/s/blog_6f57a7150100o9vo.html

源代码见附件 txt 文件.