# 西安交通大学

# 数字图像处理作业报告

## 作业 5：频域滤波器

摘要：本次报告首先简要介绍了频域滤波器的基本概念及原理，然后展示了通过 Matlab 编程实现以下功能的具体过程：（1）设计 Butterworth 与 Gaussian 低通滤波器，选择合适半径，平滑测试图像并计算功率谱比；（2）设计 Butterworth 与 Gaussian 高通滤波器，选择合适半径，在频域增强边缘并计算功率谱比；（3）实现 Laplace 和 Unmask 高通滤波器，对测试图像滤波。最后，根据实验结果，对空域与频域的低高通滤波进行了对比分析。

关键词：频域滤波，MATLAB

姓　名：胡　　欣　　盈

班　级：自　动　化　9 4

学　号：2 1 9 4 3 2 3 1 7 6

提交日期：2022 年 3 月 23 日

# 目　录

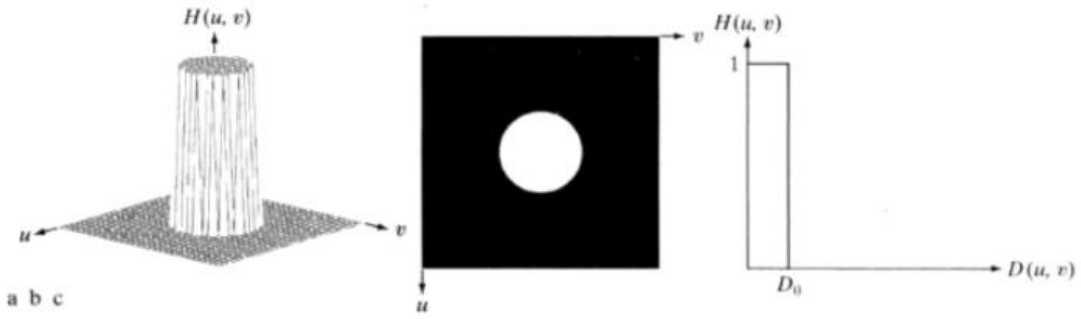
## 一、基本概念及原理

### （1）理想低通滤波器

理想低通滤波器在以原点为圆心、$D_0$为半径的园内，通过所有的频率，而在圆外截断所有的频率。（圆心的频率最低，为变换的直流分量）。函数如下：

$$H(s,v)=\begin{cases}1, D(u,v)\le D_0\\ 0, D(u,v)>D_0\end{cases}$$

根据理想低通滤波器的函数画出其图像如下，可以看出，理想低通滤波器的过渡非常急剧。



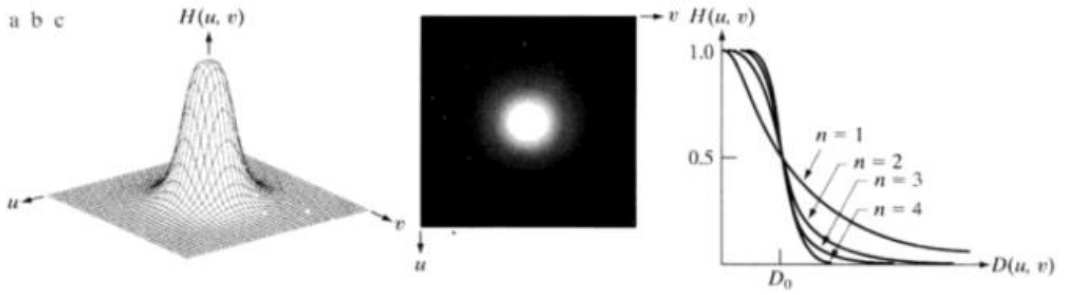

(a)一个理想低通滤波器变换函数的透视图；(b)以图像形式显示的滤波器；(c)滤波器径向横截面

### （2）Butterworth 低通滤波器

Butterworth 低通滤波器的函数表达式如下，其中 n 称为其阶数::

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

根据函数画出其图像如下：从图中，我们可以看出它的过渡没有理想低通滤波器那么剧烈。同时阶数越高，过渡越剧烈，越像理想低通滤波器；当阶数越低时，过渡越平缓，越像 Gaussian 低通滤波器。



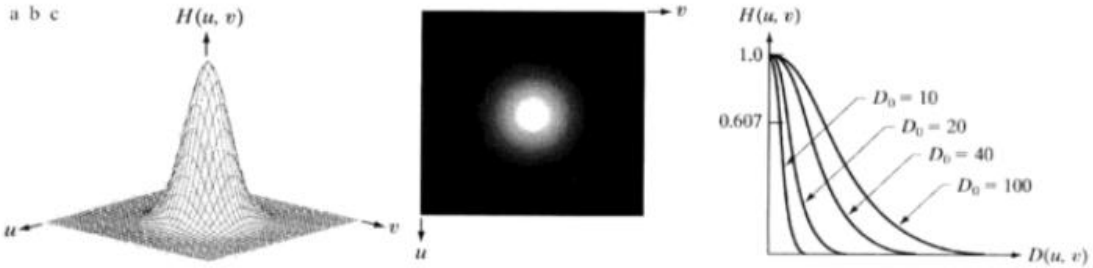

(a) 布特沃斯低通滤波器传递函数的透视图；(b) 显示为图像的滤波器；(c) 阶数为 1 到 4 的滤波器的径向剖面

## （3）Gaussian 低通滤波器

Gaussian 低通滤波器的函数表达式如下：

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

根据函数画出其图像如下：




(a) 一个 GLPF 传递函数的透视图；(b) 显示为图像的滤波器；(c) 不同 $D_0$ 值的滤波器径向剖面图

## （4）高通滤波器

高通滤波与低通滤波正好相反，是频域图像的高频部分通过而抑制低频部分。在图像中图像的边缘对应高频分量，因此高通滤波的效果是图像锐化。通常，我们用 1 减去一种低通滤波器，就可以得到相应的高通滤波器，反之亦然。

同样最简单的高通滤波器是理想高通滤波器。通过设置一个频率阈值，将高于该阈值的频率部分通过，而低于阈值的低频部分设置为 0。

对于 Butterworth 滤波器，它的函数表达式除了可以用 1 减去其低通滤波的函数表达式之外，还可以直接将原低通表达式中的 D0 和 D(u,v)分子分母上下颠倒即可。

### （5）Laplace 高通滤波器

频域的 Laplace 算子可以由如下的滤波器实现：

$$H(u,v) = -4\pi^2(u^2+v^2)$$

前提是 F(u,v)的原点在进行图像变换之前已经通过执行运算 $f(x,y)(-1)^{x+y}$ 中心化了，使得变换中心(u,v)=(0,0)就是频率矩形的中点,否则

$$H(u,v) = -4\pi^2[(u-\frac{M}{2})^2+(v-\frac{N}{2})^2]$$

### （6）Unmask 高通滤波器

钝化模板由如下的表达式给出：

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

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

最后图像由下式给出,当 k=1 时，为钝化模板；k>1 时，为高频提升滤波器。

$$g(x,y) = f(x,y) + k\cdot g_{max}(x,y)$$

## 二、频域低通滤波器

题目要求设计低通滤波器包括 Butterworth 和 Gaussian，并选择合适的半径，计算功率谱比,平滑测试图像 test1 和 test2;分析各自优缺点。

在 MATLAB 中，首先用 imread()将图像读入，再使用 mat2gray()

函数将 unit8 转化为 double，为下面的处理做铺垫。根据前述原理构建函数 Butterworth_low (Img_in,D0,n) 和 Gaussian_low (Img_in,D0)。

通过参数为 D0=200,n=5 的 Butterworth 低通滤波器分别对两幅图片进行滤波，Butterworth 低通滤波下的 test1 与 test2 功率谱比分别为 0.9984 和 0.9942，第一幅图像的功率谱比要稍大一些，结果如图 2-1、2-2 所示，可以看出，两幅图像均变得更加光滑。


图 2-1~2-2 D0=200,n=5 的 Butterworth 低通滤波

通过参数为 D0=100 的 Gaussian 低通滤波器分别对两幅图像进行滤波，低通滤波下的 test1 与 test2 功率谱比分别为 0.9839 和 0.9839，结果如图 2-3、2-4 所示，可以看出，两幅图像均变得光滑，其滤波效果和 Butterworth 滤波器基本相当，且 Gaussian 滤波略胜一筹。

但是两者在截止频率 D0 相同时，由于在过渡带处存在差异，所以功率谱比有所不同，当 Butterworth 低通滤波器的 n 越大时，过渡带越陡峭，功率谱比与 Gaussian 滤波器的差异也会越大。同时，当将两个滤波器的 D0 减小时，图像会越来越模糊，功率谱也会越来越

小，即滤波后包含的低频分量越来越少。

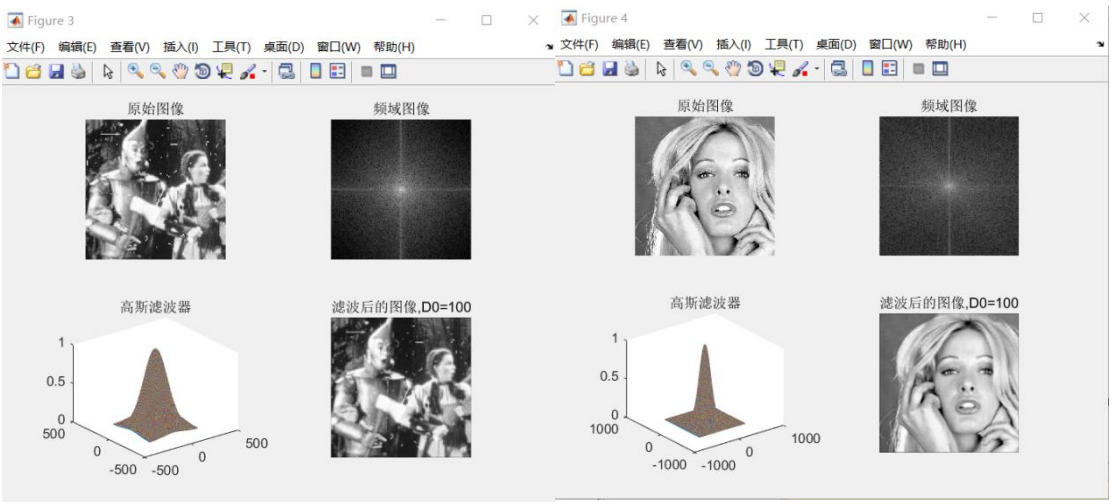



图 2-3~2-4 D0=100 的 Gaussian 低通滤波

## 三、频域高通滤波器

本题要求设计高通滤波器包括 Butterworth 和 Gaussian，在频域增强边缘。选择半径和计算功率谱比，测试图像 test3 与 test4，并分析各自优缺点。

在 MATLAB 中，首先用 imread()将图像读入，再使用 mat2gray()函数将 unit8 转化为 double，为下面的处理做铺垫。根据前述原理构建函数 Butterworth_high (Img_in,D0,n) 和 Gaussian_high (Img_in,D0)

通过参数为 D0=20,n=5 的 Butterworth 高通滤波器分别对两幅图像进行滤波，Butterworth 低通滤波下的 test3 与 test4 功率谱比分别为 0.0296 和 0.0157,结果如图 3-1、3-2 所示。可以看到，通过高通滤波将图像的低频分量滤去，将高频分量保留提取出来，所以可以用来做图像的边缘提取。

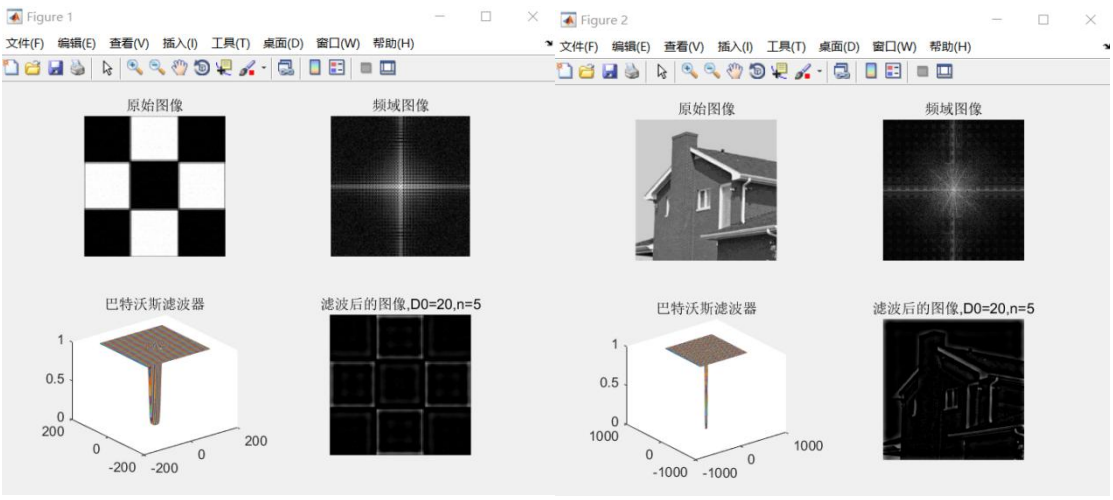



图 3-1~3-2 D0=20,n=5 的 Butterworth 高通滤波

通过参数为 D0=20 的 Gaussian 高通滤波器，分别对两幅图像进行高通滤波，Gaussian 高通滤波下的 test3 与 test4 功率谱比分别为 0.0208 和 0.0365，结果如图 3-3、3-4 所示。由于滤去了低频分量，保留了高频分量，所以提取出了图像的边缘。

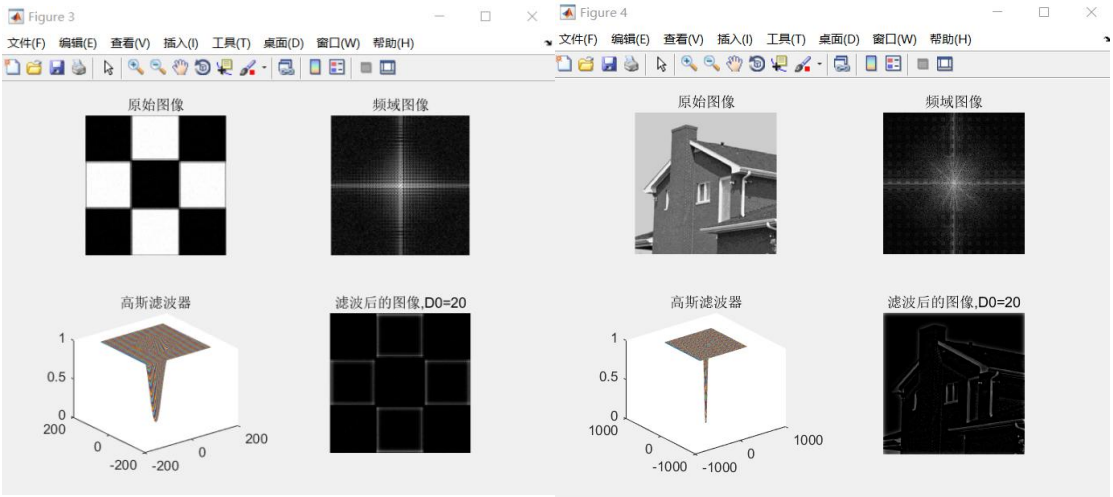



图 3-3~3-4 D0=20 的 Gaussian 高通滤波

对比两种滤波器可以看出，两者效果基本相当，Gaussian 滤波略优一些。同时，如果调整程序中的参数可以发现，当截止频率 D0 增加时，两种滤波器所得到的图像边缘越来越清晰，但是当 D0 到一定程度时，由于滤去能量过多，所以图像会整体呈现黑色。且高通滤

波器在滤波时，会将直流分量一起滤除，导致图像变暗。

## 四、其他高通滤波器

本题要求使用 Laplace 和 Unmask 高通滤波器，对测试图像 test3,4 滤波并分析各自优缺点，同时比较并讨论空域低通高通滤波与频域低通和高通的关系。根据前述原理构建函数Laplacian(Img_in)与 Unmask(Img_in,D0,n)。

通过 Laplace 高通滤波分别对两幅图像进行处理，结果如图 4-1、4-2 所示，可以看出滤波器的边缘增强效果，但这个效果与之前的两个高通滤波器所得到的效果并不相同，Laplace 得到的滤波后图像有明显的线条感，视觉效果不佳。

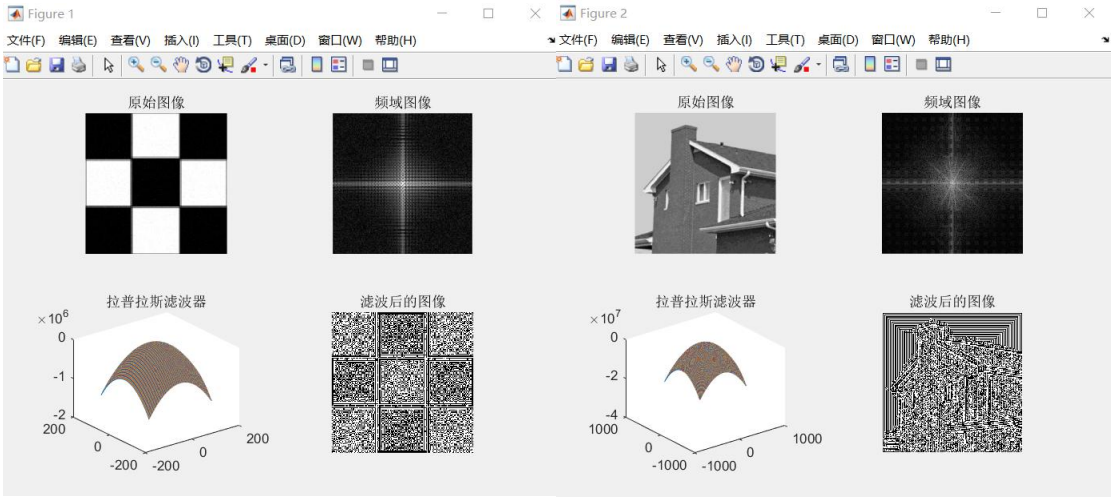



图 4-1~4-2 Laplace 高通滤波

通过 Unmask 高通滤波器分别对两幅图像进行高通滤波，结果如图 4-3、4-4 所示，经过仔细观察还是可以发现其边缘增强效果的。这个效果又不同与前述的几种高通滤波器的效果，让图像边缘的对比度有了一定的提升，总体来说图像更加清晰了。

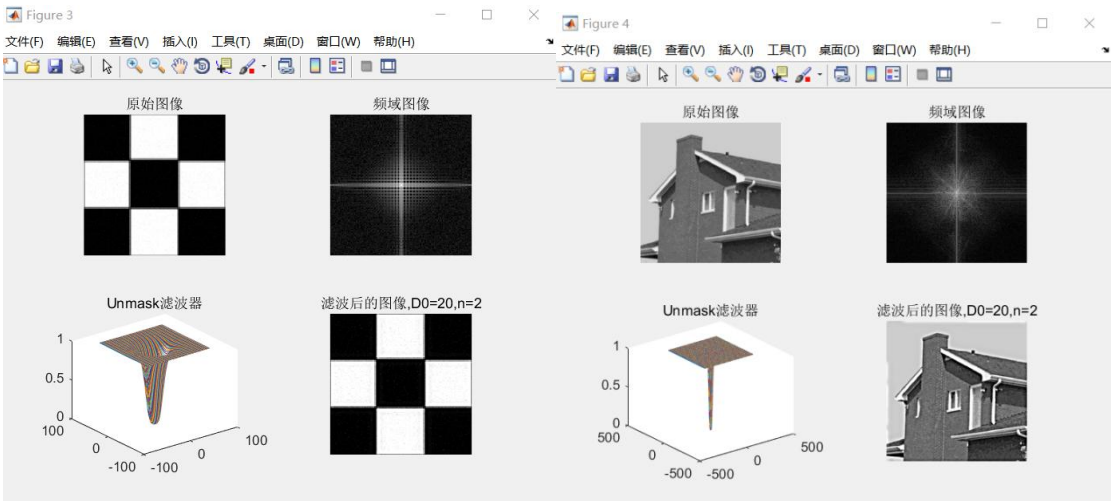



图 4-3~4-4 D0=20,n=2 的 Unmask 高通滤波

空域滤波定义为滤波函数与输入图像进行卷积，而频域滤波则定义为滤波函数与输入图像的傅里叶变换进行相乘。两者的纽带则是卷积定理和傅里叶变换。频率域增强技术和空间域增强技术有密切的联系，一方面，许多空域增强技术可借助频域概念来分析和帮助设计；另一方面，许多空域增强技术也可以通过频域来实现，而频域增强也可以通过空域实现。空域主要包括平滑滤波和锐化滤波。平滑滤波是要滤除不规则的噪声或干扰的影响，从频域的角度来看，不规则的噪声具有较高的频率，所以可用具有低通能力的频域滤波器来滤除。

# 附录

## 附录 1：参考文献

[1]阮秋琦,阮宇智等.数字图像处理(第三版)(美)冈萨雷斯[M],2003,电子工业出版社

[2]门敬文•数字图像处理 MATLAB 版[M].2007.2,国防工业出版社.

[3]数字图像处理——虚宇宸轩-CSDN 博客

# 附录 2：源代码

## （1）频域低通滤波器

```
clear
clc

test1=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test1.pgm');
test2=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test2.tif');
test1 =mat2gray(test1);
test2 =mat2gray(test2);
%Task1
Butterworth_low(test1,200,5)
Butterworth_low(test2,200,5)
Gaussian_low(test1,100)
Gaussian_low(test2,100)

function p=Butterworth_low(Img_in,D0,n)
[M,N]=size(Img_in);    M=2*M;N=2*N;
u = -M/2:(M/2-1);   v = -N/2:(N/2-1);
[u,v] = meshgrid(u,v);
D = sqrt(u.^2+v.^2);
%设计滤波器：
H = 1./(1+(D./D0).^(2*n));
Img_fft= fft2(Img_in,size(H,1),size(H,2));
Img_fft_shift = fftshift(Img_fft);
Img_Butterworth = Img_fft_shift.*H;
Img_out = ifft2(ifftshift(Img_Butterworth));
Img_out = Img_out(1:size(Img_in,1),1:size(Img_in,2));
%Img_out =mat2gray(Img_out);
%输出图像：
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);plot3(u,v,H);title('Butterworth 滤波器');
subplot(2,2,4);imshow(Img_out);title(['滤波后的图像,D0=',num2str(D0),',n=',num2str(n)]);
%计算功率谱比：
S = 0;S1 = 0;
[P,Q] = size(Img_fft_shift);
for a = 1:P
    for b=1:Q
        S1 = S1+(abs(Img_Butterworth(a,b)))^2;
        Img_out = (abs(Img_fft_shift(a,b)))^2;
        S=S+Img_out;
```

```
    end
end
p = S1/S;
end

function p=Gaussian_low(Img_in,D0)
[M,N]=size(Img_in);    M=2*M;N=2*N;
u = -M/2:(M/2-1);   v = -N/2:(N/2-1);
[u,v] = meshgrid(u,v);
D = sqrt(u.^2+v.^2);
%设计滤波器:
H = exp(-D.^2/(2.*(D0.^2)));
Img_fft= fft2(Img_in,size(H,1),size(H,2));
Img_fft_shift = fftshift(Img_fft);
Img_Gaussian = Img_fft_shift.*H;
Img_out = ifft2(ifftshift(Img_Gaussian));
Img_out = Img_out(1:size(Img_in,1),1:size(Img_in,2));
%Img_out =mat2gray(Img_out);
%输出图像:
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);plot3(u,v,H);title('Gaussian 滤波器');
subplot(2,2,4);imshow(Img_out);title(['滤波后的图像,D0=',num2str(D0)]);
%计算功率谱比:
S = 0;S1 = 0;
[P,Q] = size(Img_fft_shift);
for a = 1:P
    for b=1:Q
        S1 = S1+(abs(Img_Gaussian(a,b)))^2;
        Img_out = (abs(Img_fft_shift(a,b)))^2;
        S=S+Img_out;
    end
end
p = S1/S;
end
```

## （2）频域高通滤波器

```
clear
clc

test3=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test3_corrupt.pgm');
```

```
test4=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test4.tif');

test3 =mat2gray(test3);
test4 =uint8(test4);
test4 =mat2gray(test4);
test4_1=test4(1:512,1:512);
%Task2
Butterworth_high(test3,20,5)
Butterworth_high(test4_1,20,5)
Gaussian_high(test3,20)
Gaussian_high(test4_1,20)

function p=Butterworth_high(Img_in,D0,n)
[M,N]=size(Img_in);    M=2*M;N=2*N;
u = -M/2:(M/2-1);   v = -N/2:(N/2-1);
[u,v] = meshgrid(u,v);
D = sqrt(u.^2+v.^2);
%设计滤波器:
H = 1./(1+(D0./D).^(2*n));
Img_fft= fft2(Img_in,size(H,1),size(H,2));
Img_fft_shift = fftshift(Img_fft);
Img_Butterworth = Img_fft_shift.*H;
Img_out = ifft2(ifftshift(Img_Butterworth));
Img_out = Img_out(1:size(Img_in,1),1:size(Img_in,2));
%输出图像:
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);plot3(u,v,H);title('Butterworth 滤波器');
subplot(2,2,4);imshow(Img_out);title(['滤波后的图像,D0=',num2str(D0), ',n=',num2str(n)]);
%计算功率谱比:
S = 0;S1 = 0;
[P,Q] = size(Img_fft_shift);
for a = 1:P
    for b=1:Q
        S1 = S1+(abs(Img_Butterworth(a,b)))^2;
        Img_out = (abs(Img_fft_shift(a,b)))^2;
        S=S+Img_out;
    end
end
p = S1/S;
end

```

```
function p=Gaussian_high(Img_in,D0)
[M,N]=size(Img_in);    M=2*M;N=2*N;
u = -M/2:(M/2-1);   v = -N/2:(N/2-1);
[u,v] = meshgrid(u,v);
D = sqrt(u.^2+v.^2);
%设计滤波器:
H = 1-exp(-D.^2/(2.*(D0.^2)));
Img_fft= fft2(Img_in,size(H,1),size(H,2));
Img_fft_shift = fftshift(Img_fft);
Img_Gaussian = Img_fft_shift.*H;
Img_out = ifft2(ifftshift(Img_Gaussian));
Img_out = Img_out(1:size(Img_in,1),1:size(Img_in,2));

%输出图像:
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);plot3(u,v,H);title('Gaussian 滤波器');
subplot(2,2,4);imshow(Img_out);title(['滤波后的图像,D0=',num2str(D0)]);
%计算功率谱比:
S = 0;S1 = 0;
[P,Q] = size(Img_fft_shift);
for a = 1:P
    for b=1:Q
        S1 = S1+(abs(Img_Gaussian(a,b)))^2;
        Img_out = (abs(Img_fft_shift(a,b)))^2;
        S=S+Img_out;
    end
end
p = S1/S;
end
```

## （3）其他高通滤波器

```
clear
clc
test3=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test3_corrupt.pgm');
test4=imread('C:\Users\LENOVO\Desktop\ImageProcessHomework\Homework5\test4.tif');

test3 =mat2gray(test3);
test4 =uint8(test4);
test4 =mat2gray(test4);
test4_1=test4(1:512,1:512);
```

```
%Task3
Laplacian(test3)
Laplacian(test4_1)
Unmask(test3,20,2)
Unmask(test4_1,20,2)

function Laplacian(Img_in)
[M,N]=size(Img_in);    M=2*M;N=2*N;
u = -M/2:(M/2-1);   v = -N/2:(N/2-1);
[u,v] = meshgrid(u,v);
D = sqrt(u.^2+v.^2);
%设计滤波器:
H = -4.*pi.^2*(u.^2+v.^2);
Img_fft= fft2(Img_in,size(H,1),size(H,2));
Img_fft_shift = fftshift(Img_fft);
Img_Laplacian = Img_fft_shift.*H;
Img_out = ifft2(ifftshift(Img_Laplacian));
Img_out = Img_out(1:size(Img_in,1),1:size(Img_in,2));
%输出图像:
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);plot3(u,v,H);title('Laplace 滤波器');
subplot(2,2,4);imshow(Img_out);title('滤波后的图像');
end

function Unmask(Img_in,D0,n)
Img_fft_shift=fftshift(fft2(Img_in));
[M,N]=size(Img_fft_shift);
for i=1:M
    for j=1:N
        d=sqrt((i-fix(M/2))^2+(j-fix(N/2))^2);
        if d==0
            H(i,j)=0;
        else
            H(i,j)=1/(1+0.414*(D0/d)^(2*n));
        end
        Img_out(i,j)=(1+H(i,j))*Img_fft_shift(i,j);
    end
end
Img_out=real(ifft2(ifftshift(Img_out)));
%输出图像:
figure;
subplot(2,2,1);imshow(Img_in);title('原始图像');
```

```
subplot(2,2,2);imshow(log(1+abs(Img_fft_shift)),[]);title('频域图像');
subplot(2,2,3);
u=-M/2:(M/2-1);v=-N/2:(N/2-1);
[u,v]=meshgrid(u,v);plot3(u,v,H);title('Unmask 滤波器');
subplot(2,2,4);imshow(Img_out);title(['滤波后的图像,D0=',num2str(D0),',n=',num2str(n)]);
end
```