

数字图像处理第五次报告

学生姓名：任泽华

班级：自动化 71

学号：2171411498

提交日期：2020-4-14

摘要：

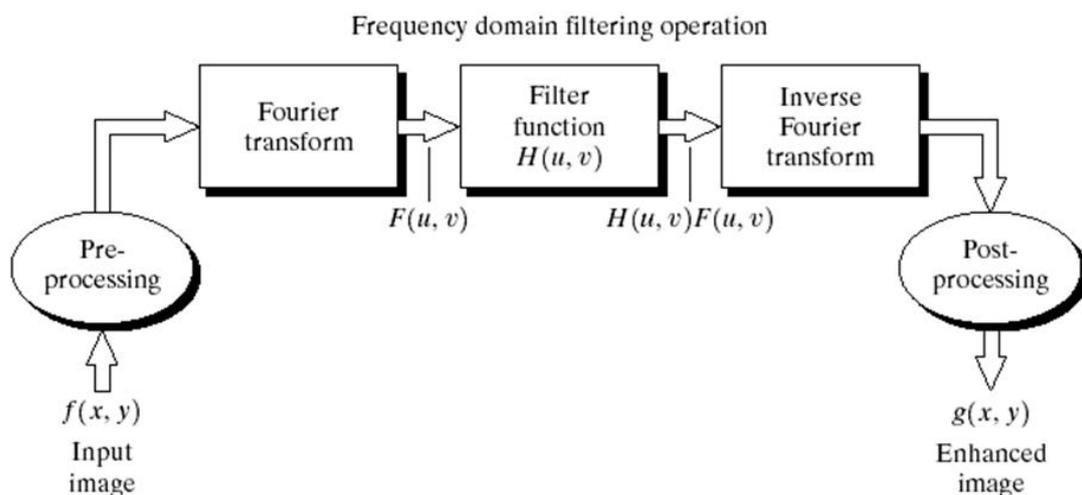
本报告主要工作：主要在频域对不同的图片分别进行了低通与高通滤波的操作，比较了理想滤波器、巴特沃斯滤波器和高斯滤波器三种滤波器的效果与优劣；采用频域滤波法对图片进行了拉普拉斯滤波和 unsharp mask 滤波，分析比较了两者的优劣，对于频域和空域的对应有了更深入的了解；最后分析比较了空域和频域的对应关系，分析了二者的关联性与区别应用。本报告软件运行环境为 MATLAB R2018b，所有代码均为自己编写。

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

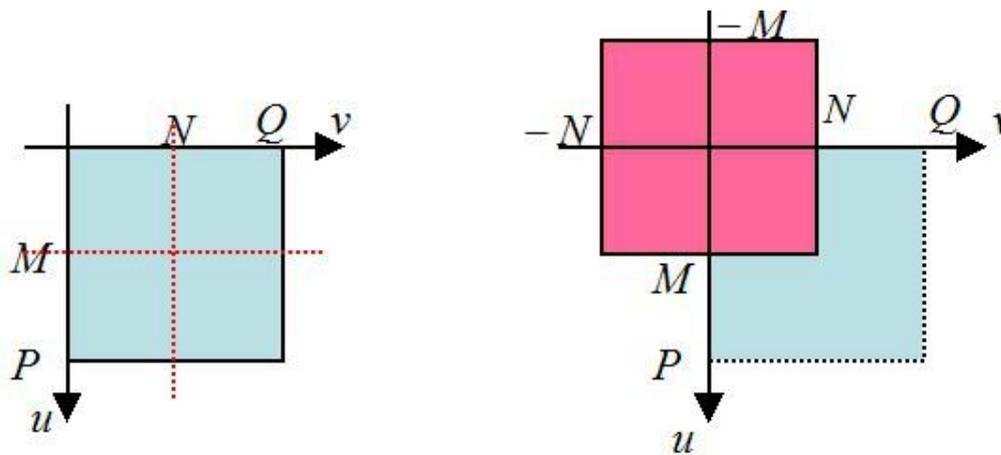
1. 低通滤波原理

(1) 频域低通滤波的主要原理为：

计算原图像的离散傅里叶变换（为保证质量，最好进行 padding，即填充 0 操作，并乘 $(-1)^{(x+y)}$ 保证零频点移到图像中央便于观察）。然后在频域进行低通滤波操作，对频域数值进行运算求得变换后的频域值。最后进行离散傅里叶反变换求得变换后的图片，只取左上角的 1/4 即可。注意最后图像还要再乘 $(-1)^{(x+y)}$ 才可以恢复。



变换流程图



零频点移到中心

(2) 关于功率谱比的定义：

功率谱图形每个像素值是图像 FFT 频域图像每个像素值的模的平方

$$P_f(u,v) = |F(u,v)|^2 \quad P_g(u,v) = |G(u,v)|^2$$

而功率谱的比即为变换后的功率谱比变换前的功率谱：

$$L = \frac{P_g(u,v)}{P_f(u,v)}$$

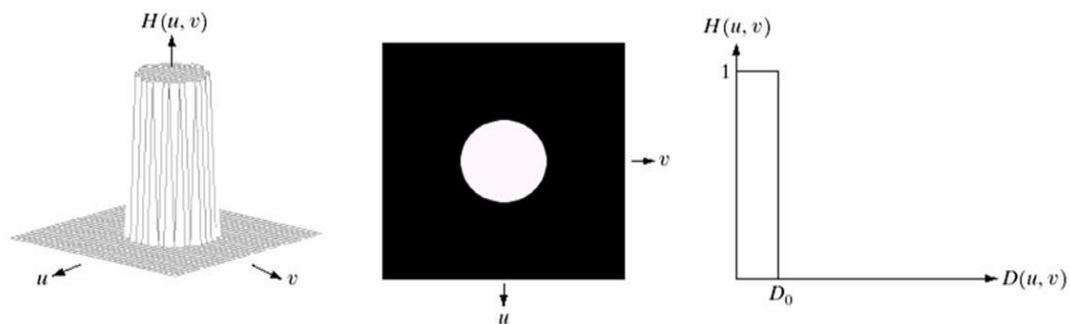
而对于某个具体的滤波器，其功率谱比即为对应的频域变换图像像素模的平方

$$L = |H(u,v)|^2$$

通过编写对应的计算程序即可计算出对应的功率谱比图形。

(3) 理想低通滤波器

理想低通滤波器思想比较朴素，对于频域分量是采取“一刀切”的策略，所以边缘效应比较明显：



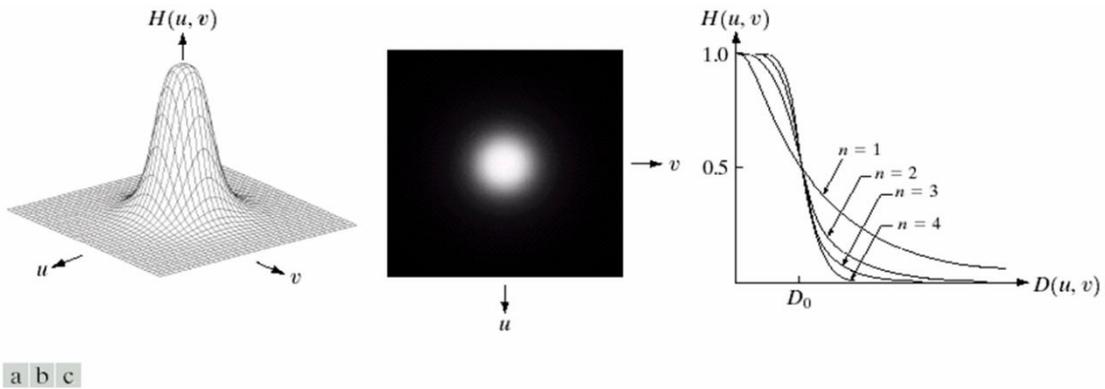
a b c

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

其中 $D(u,v)$ 是 (u, v) 点到中心的欧氏距离:

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

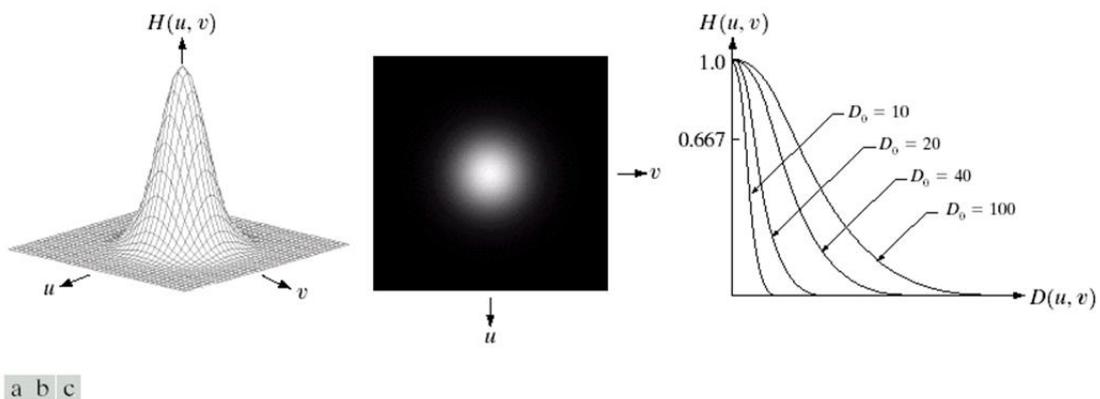
(4) 巴特沃斯低通滤波器



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

其中 $D(u,v)$ 也是 (u, v) 点到中心的欧氏距离, n 为巴特沃斯低通滤波器的阶次

(5) 高斯低通滤波器



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

其中 σ 取 D_0 时:

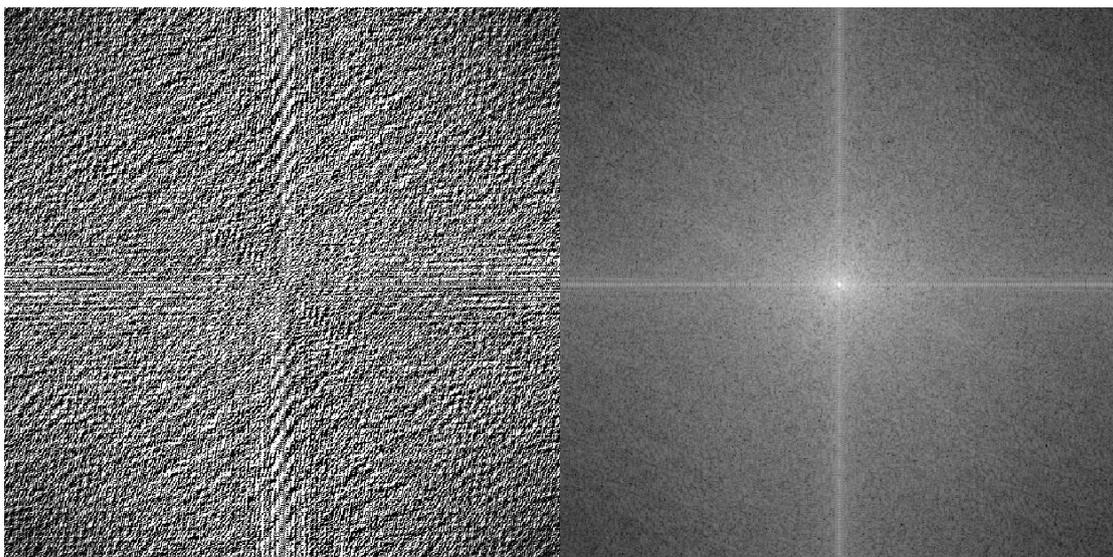
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

2. FFT 的显示



原图

1:1 显示

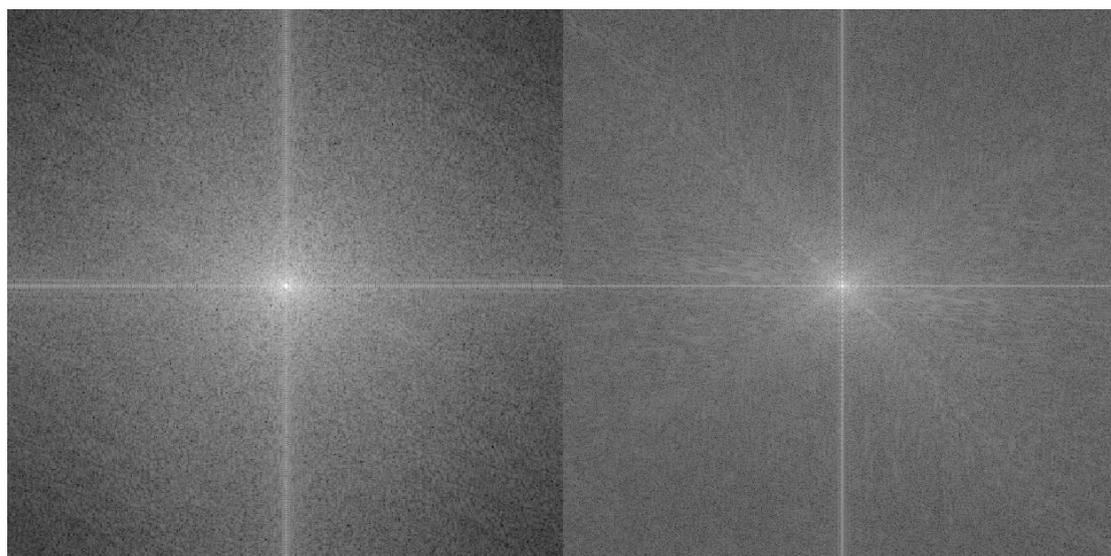


0-255 显示

log 显示

如图所示,将原本的 FFT 图像原封不动地显示出来,由于零频点过大,导致其他部分看不清楚,而只显示 0-255 范围的内容时虽然有部分细节信息,但是丢失了一部分重要的细节内容,最好的显示方式就是进行以 e 为底数的 \log 函数显示。下面的 FFT 图像都用这种方法显示。

下面是 test1 和 test2 对应的 FFT 显示图



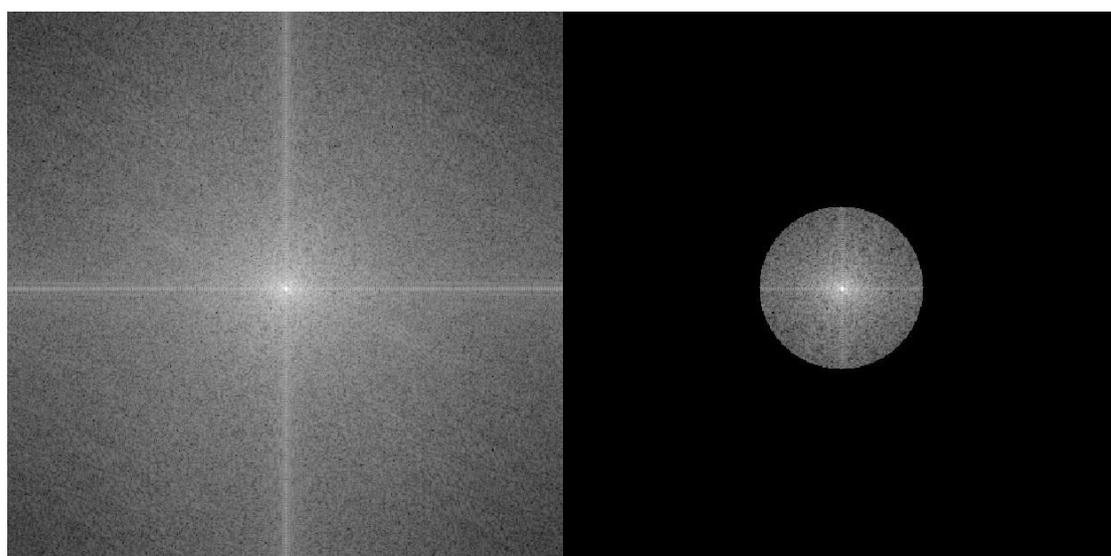
test1

test2

3. 理想低通滤波器

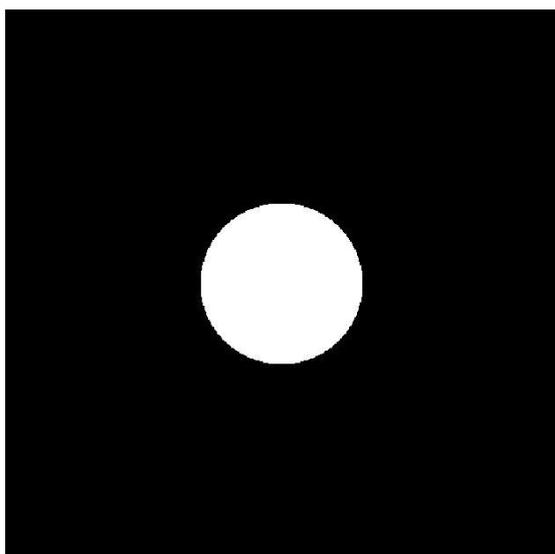
选择 $D_0=75$ 的理想低通滤波器

上面四幅为 test1，下面四幅为 test2



FFT

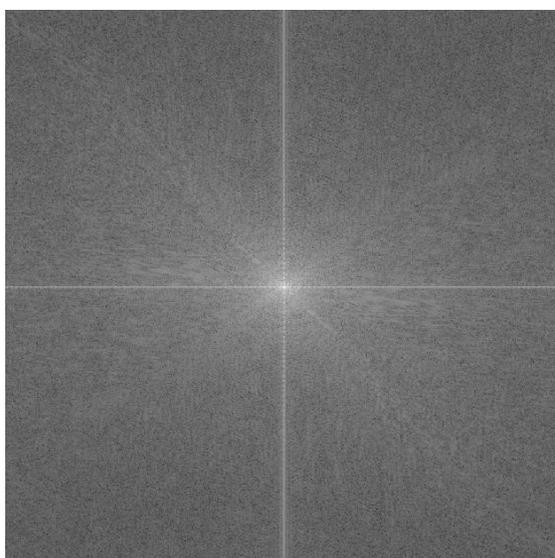
滤波 FFT



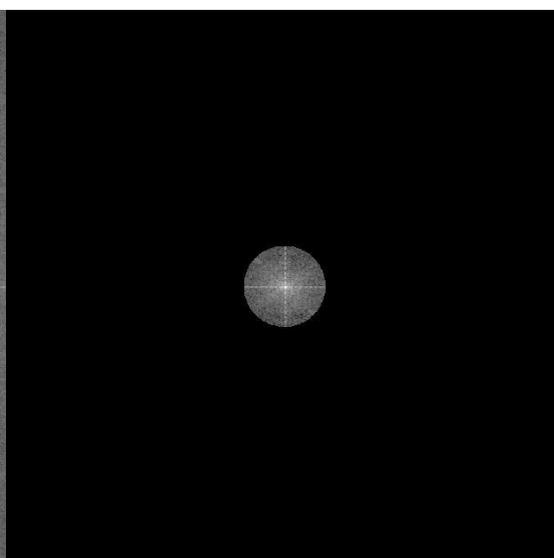
功率谱比



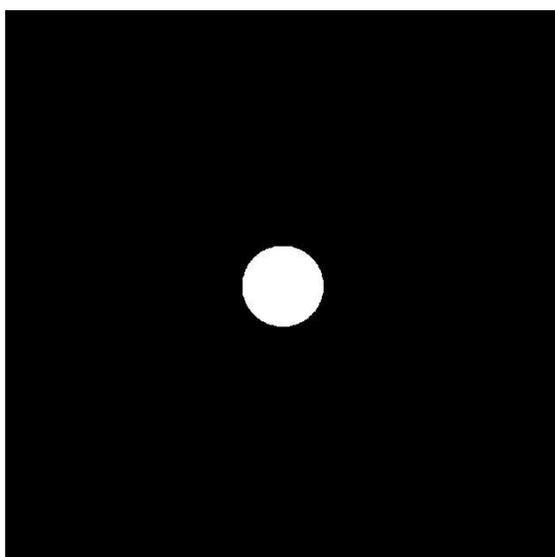
结果图



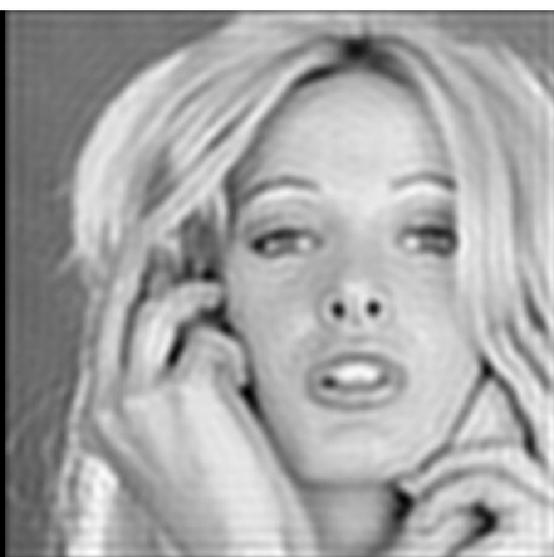
FFT



滤波 FFT



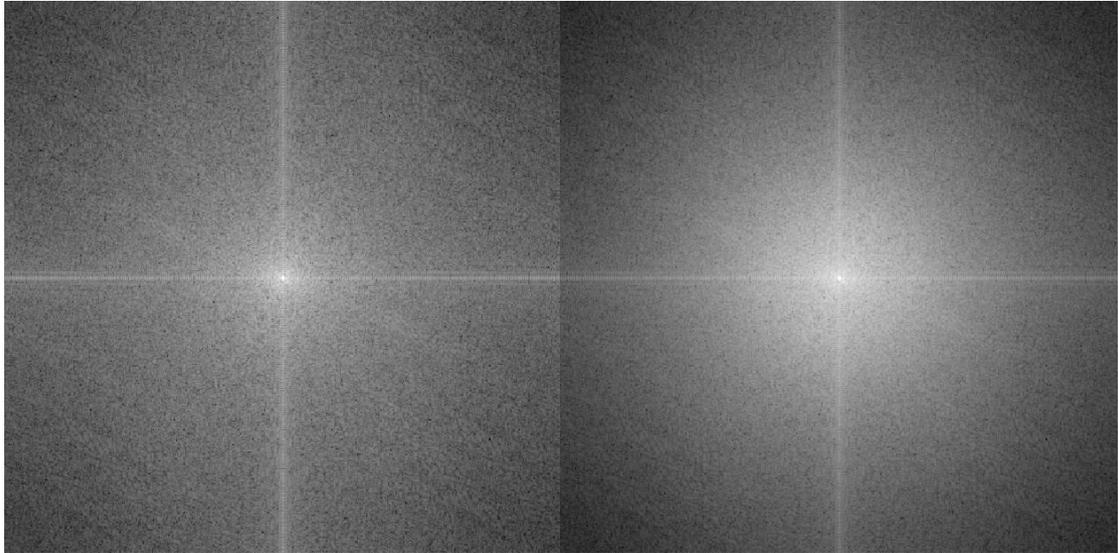
功率谱比



结果图

4. 巴特沃斯低通滤波器

选择 $D_0=75$ 的二阶巴特沃斯低通滤波器



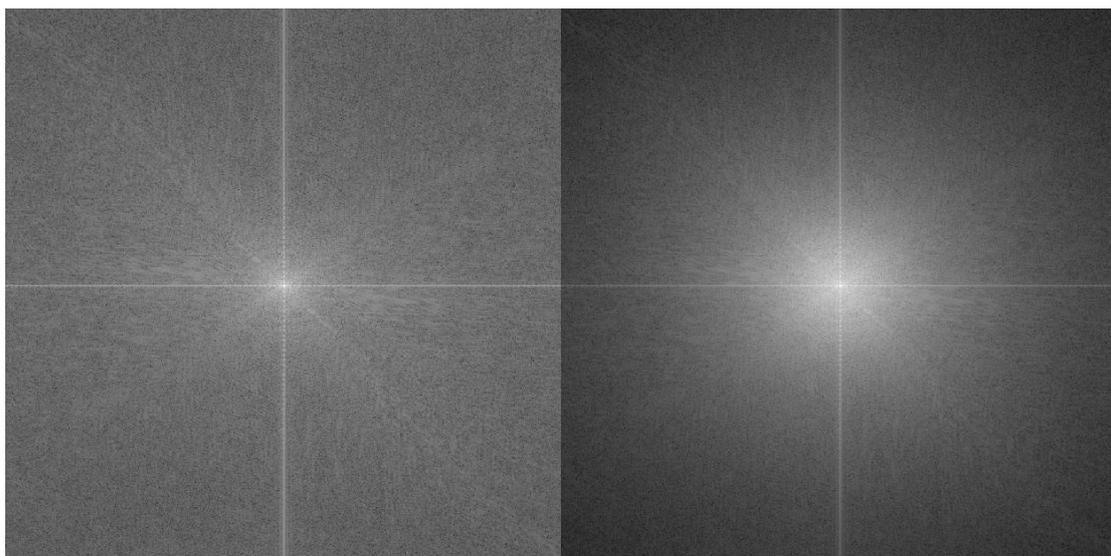
FFT

滤波 FFT



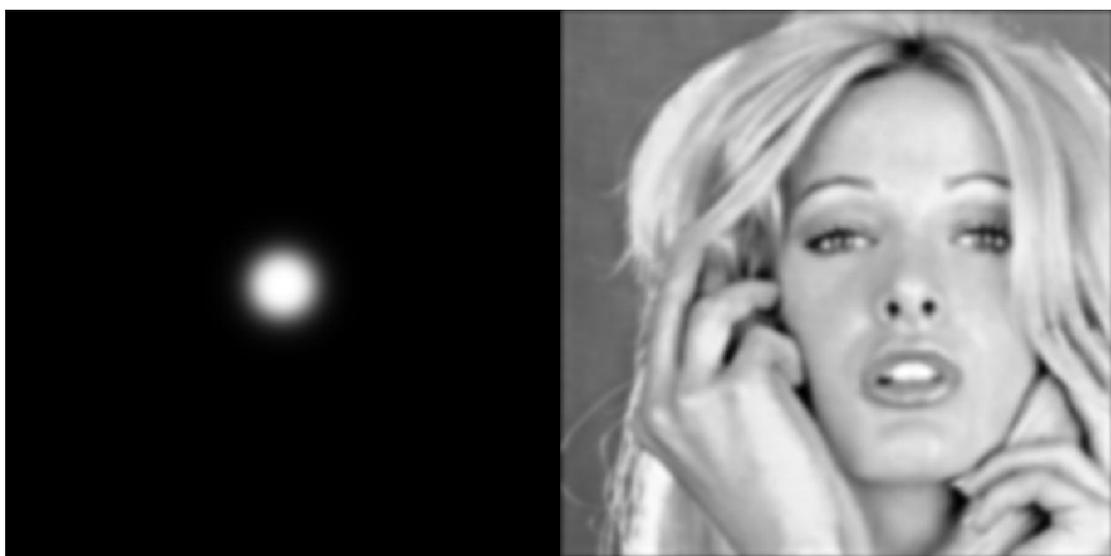
功率谱比

结果图



FFT

滤波 FFT

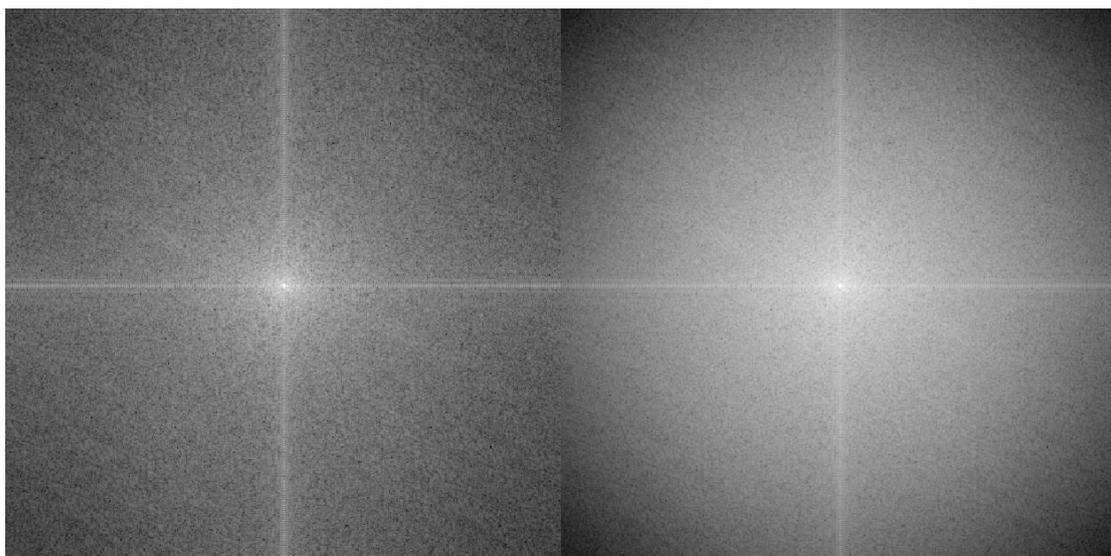


功率谱比

结果图

5. 高斯低通滤波器

选择 $D_0=75$ 的高斯低通滤波器



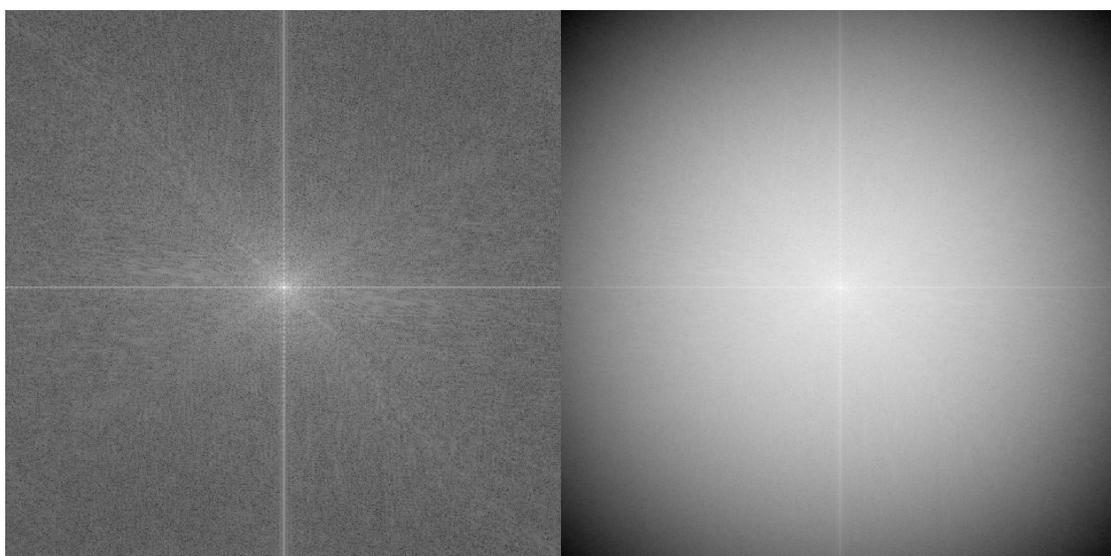
FFT

滤波 FFT



功率谱比

结果图



FFT

滤波 FFT



功率谱比

结果图

6. 三种滤波方式比较

从图像效果可以看出，理想低通的滤波效果最差，而且有很明显的振铃现象边缘被多次重复，所以在实际应用中不做考虑。

对比二阶巴特沃斯和高斯，高斯滤波器的图像更加平滑清晰，滤波效果也很不错，这是因为高斯滤波器是巴特沃斯阶数 n 趋近于无穷后的

结果，而 n 越大滤波效果越好，所以高斯会由于巴特沃斯的效果。



原图

理想低通



巴特沃斯



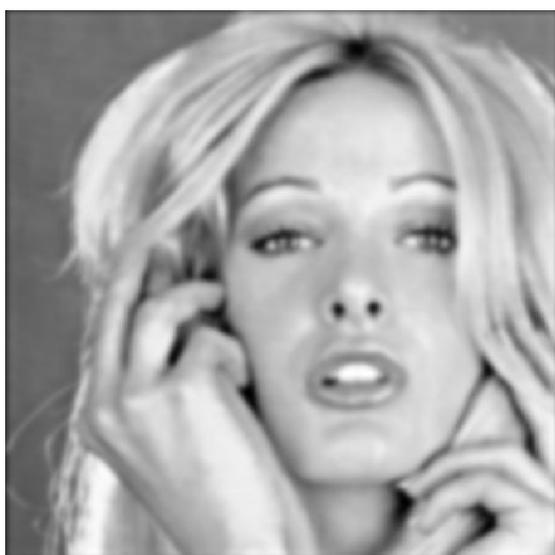
高斯低通



原图



理想低通



巴特沃斯



高斯低通

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. 高通滤波原理

(1) 理想高通

类比于前一问的低通，可以对应地给出高通滤波器的传递函数：

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

(2) 巴特沃斯高通

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

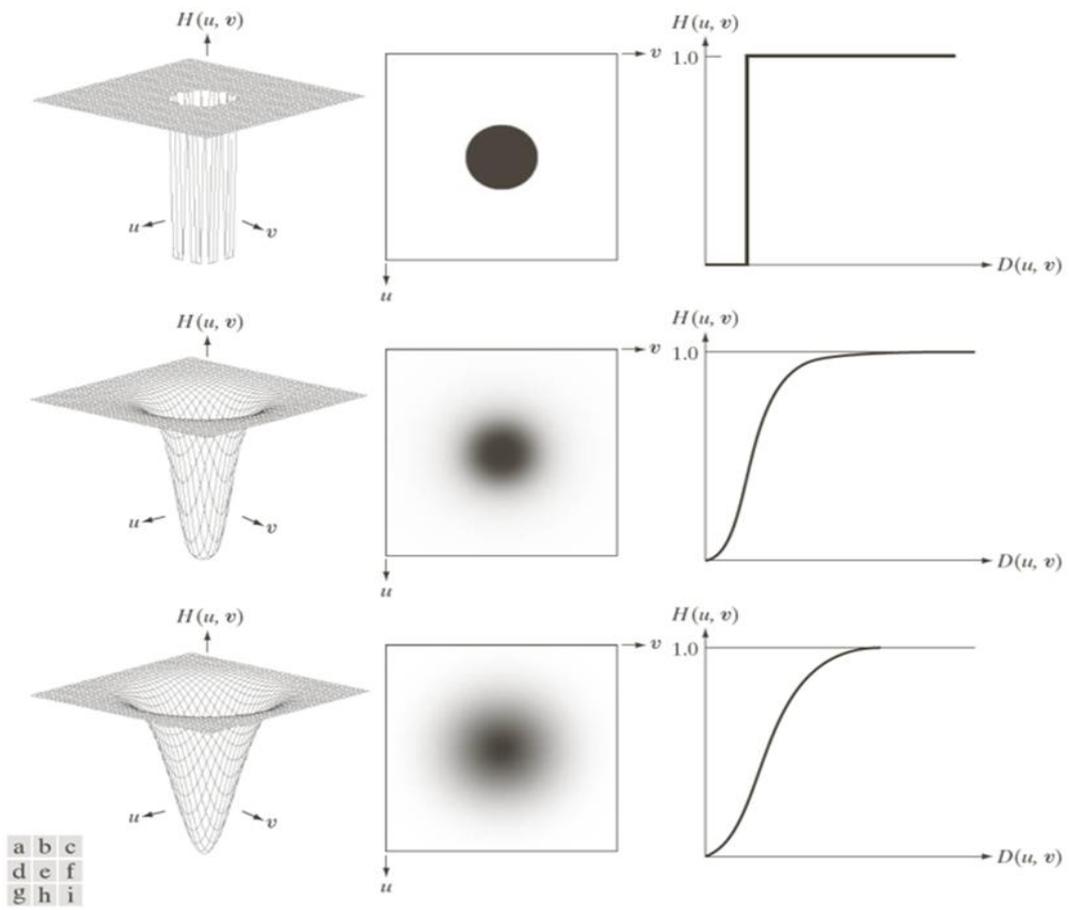
(3) 高斯高通

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2\sigma^2}}$$

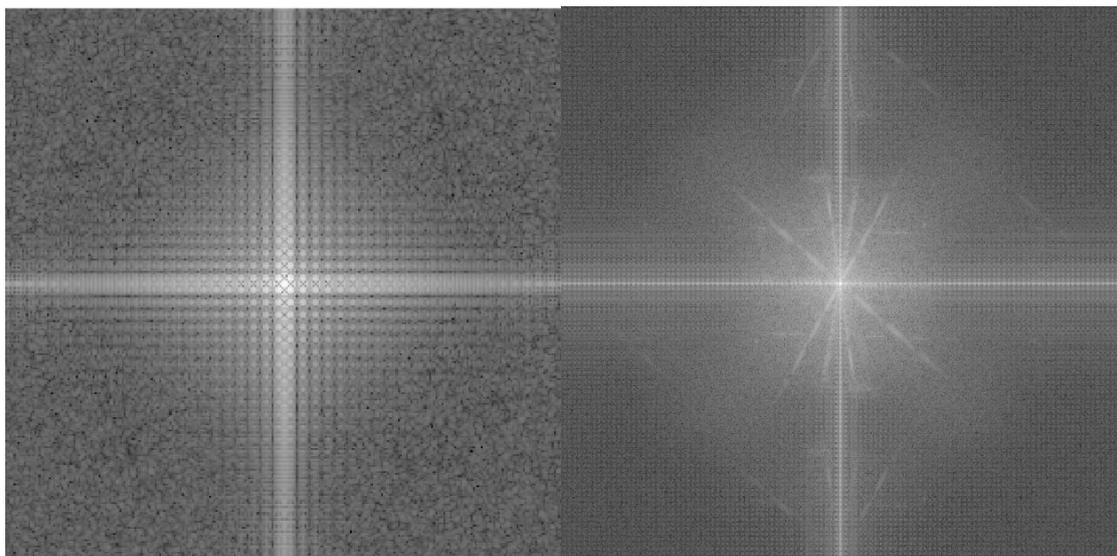
同样地，令 $\sigma = D_0$ ：

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

(4) 示意图



2. FFT 的显示

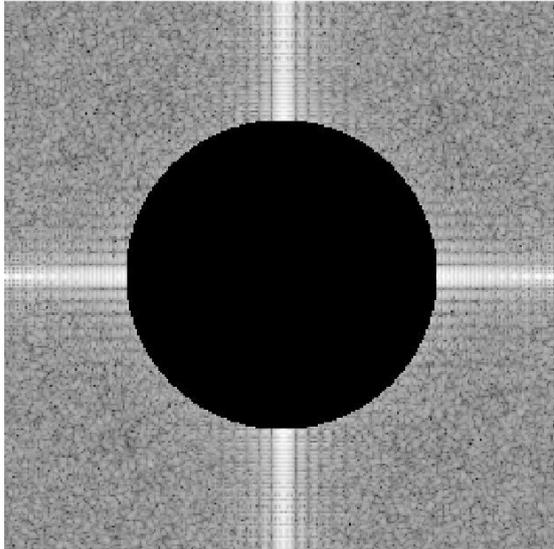


test3

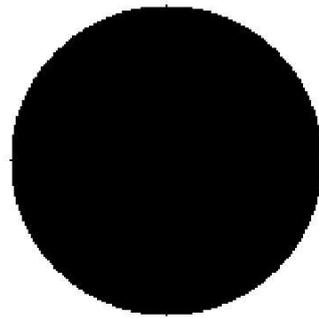
test4

3. 理想高通滤波器

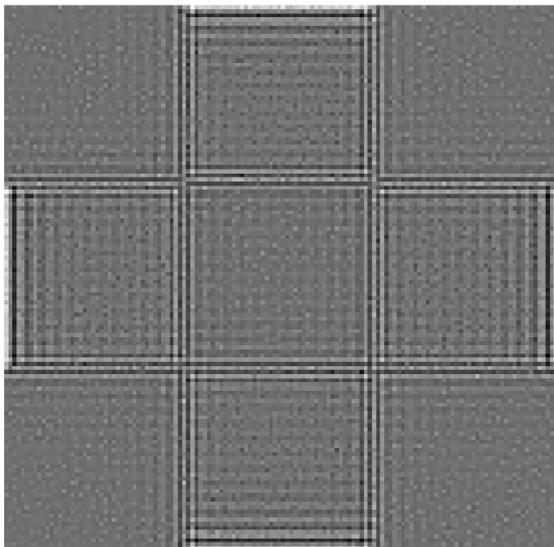
选择 $D_0=75$ 的理想高通滤波器



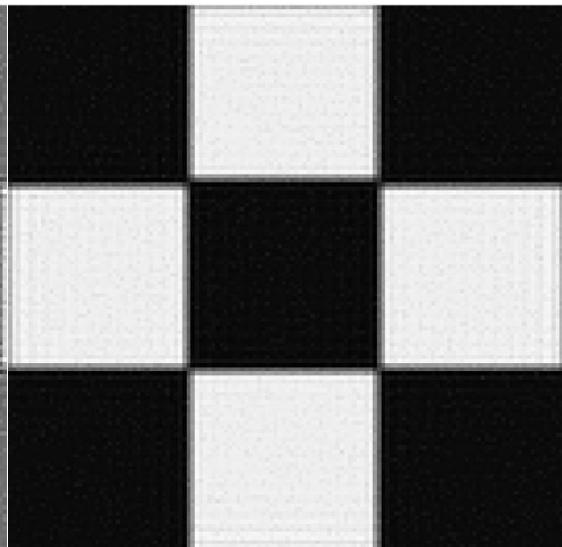
滤波 FFT



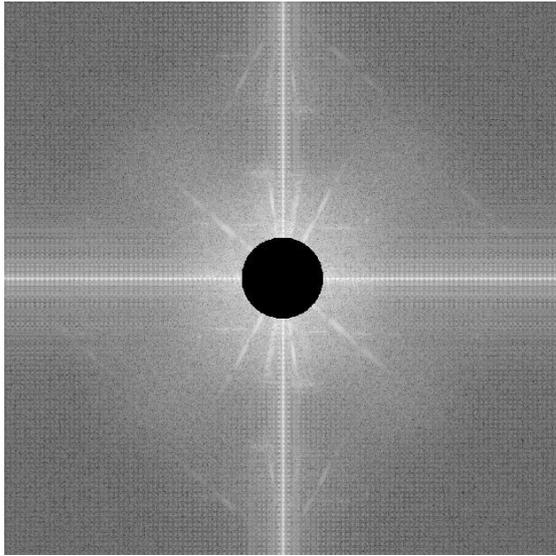
功率谱比



边缘图



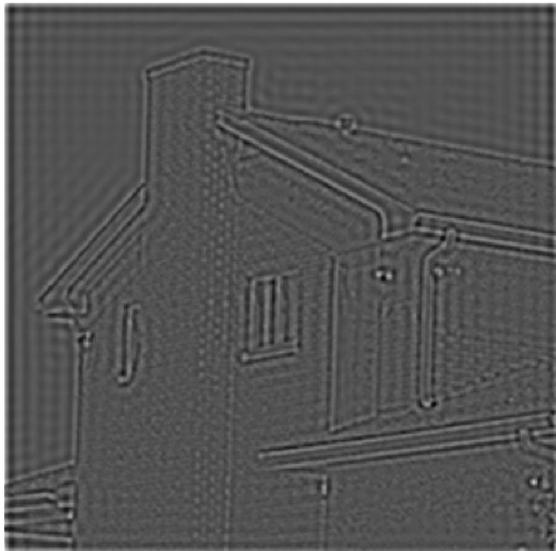
增强图



滤波 FFT



功率谱比



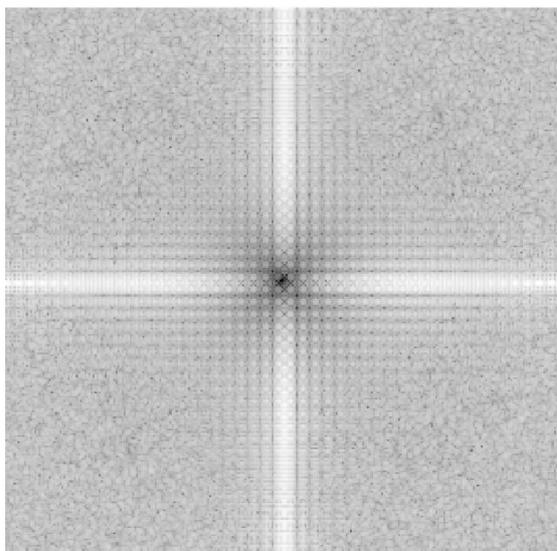
边缘图



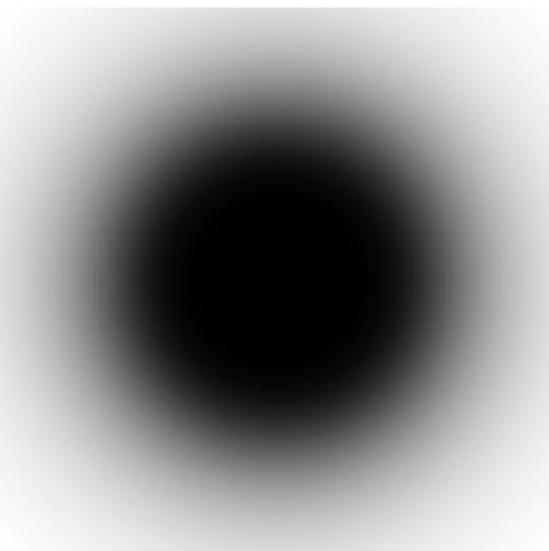
增强图

4. 巴特沃斯高通滤波器

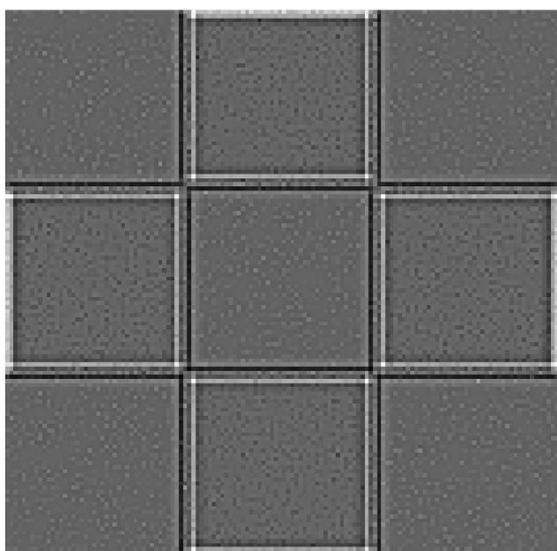
选择 $D_0=75$ 的二阶巴特沃斯高通滤波器



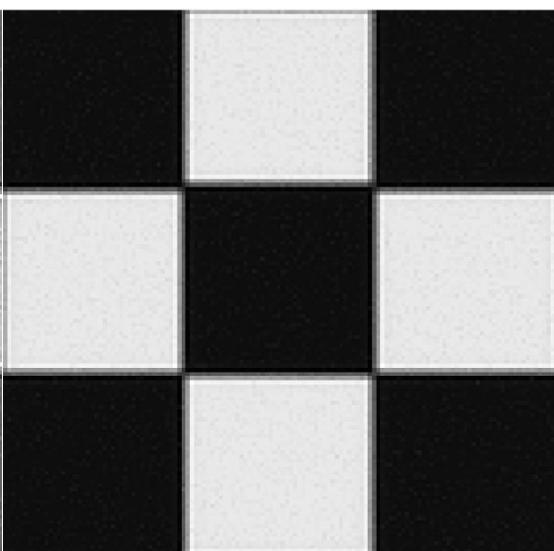
滤波 FFT



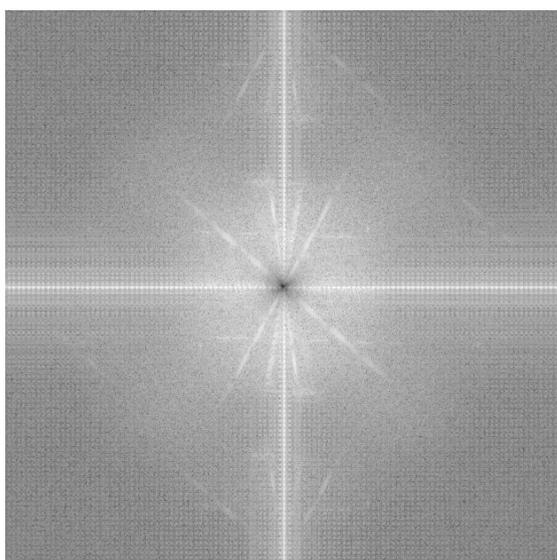
功率谱比



边缘图



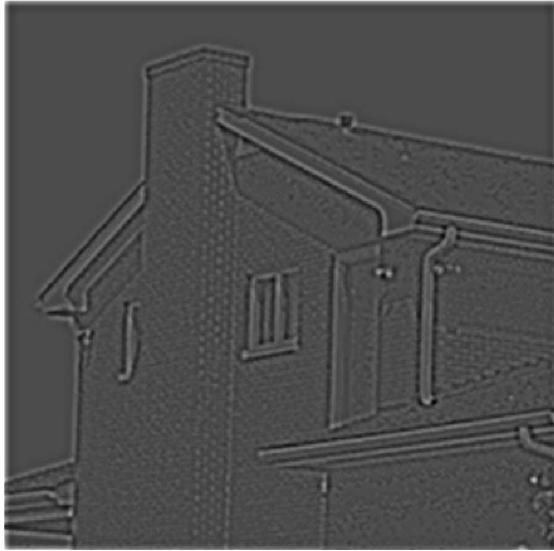
增强图



滤波 FFT



功率谱比



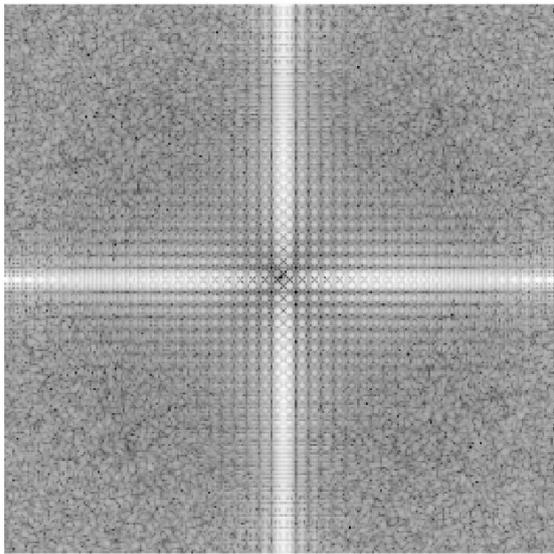
边缘图



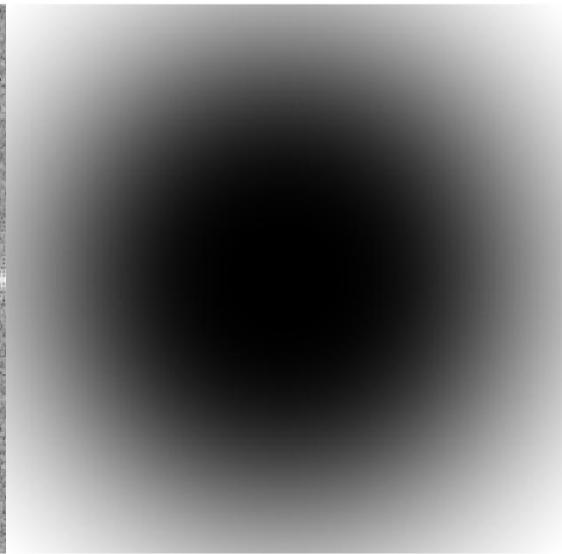
增强图

5. 高斯高通滤波器

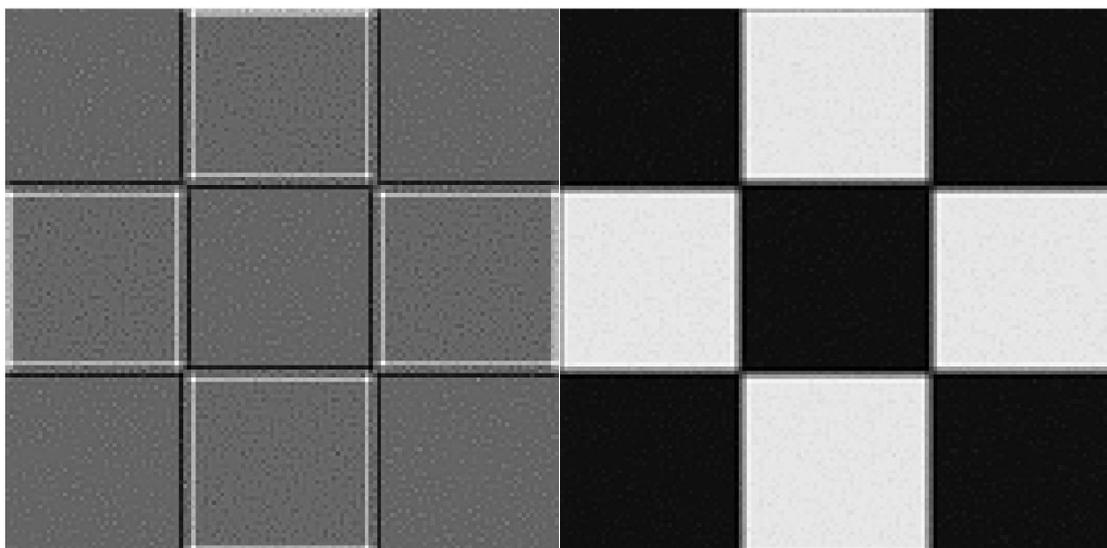
选择 $D_0=75$ 的高斯高通滤波器



滤波 FFT

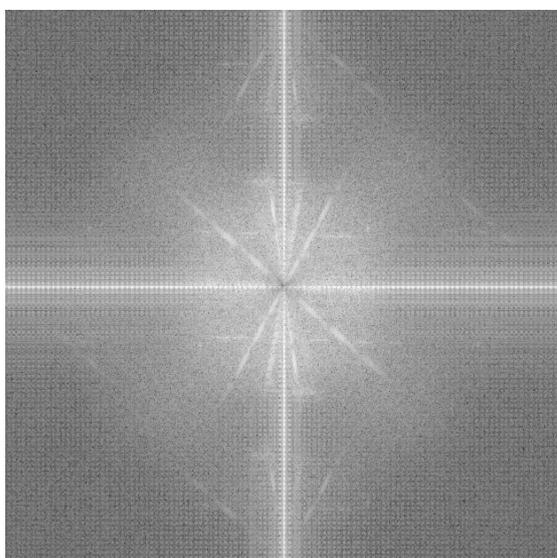


功率谱比

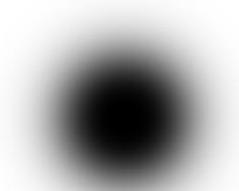


边缘图

增强图



滤波 FFT



功率谱比



边缘图



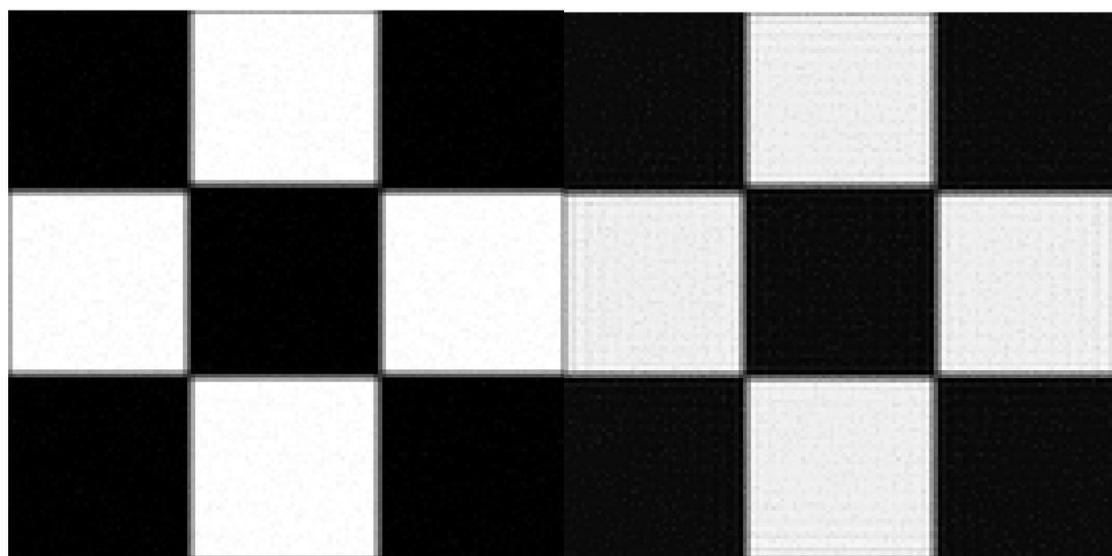
增强图

6. 三种滤波方式比较

理想低通在高通滤波器中由于边缘效应也会存在很明显的振铃现象，从两幅图当中都可以很明显的看出来。

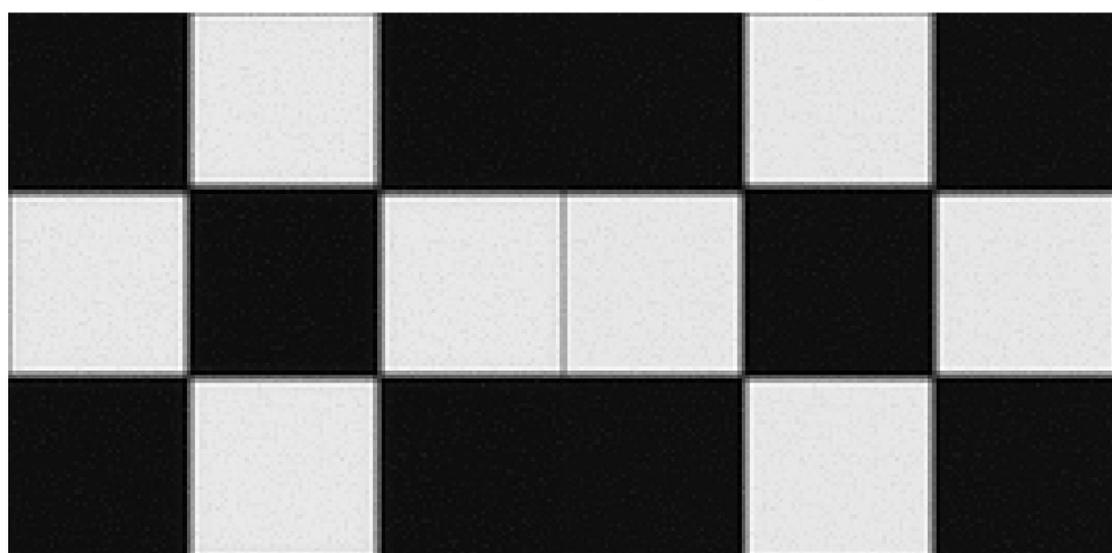
而巴特沃斯和高斯高通滤波器的增强效果都相差不大，在视力可分辨的范围内没有明显区别。

而且有一个十分有趣的现象：原图中纯色部分经过增强变得颜色暗淡偏灰，这是因为这些部分是光滑区域，在高通滤波器下被滤掉的部分很多，这就导致这些位置颜色整体变灰。



原图

理想低通



巴特沃斯

高斯低通



原图



理想低通



巴特沃斯



高斯低通

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；

1. 两种滤波方法简介

(1) 拉普拉斯

在频域拉普拉斯算子可由以下传递函数描述：

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

在经过中心点变换后可以描述为：

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

(2) Unsharp mask

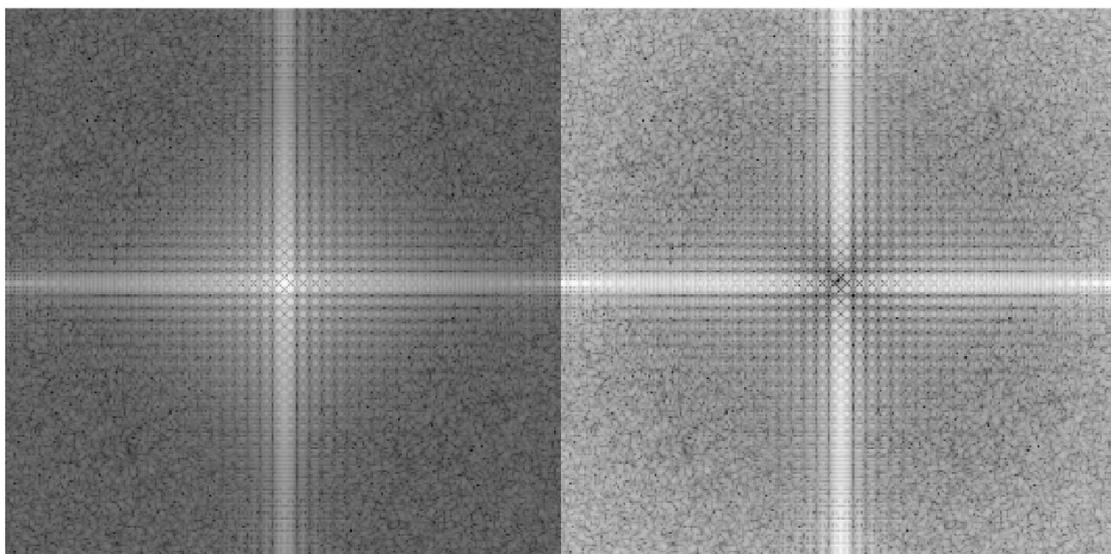
$$g_{max}(x,y) = f(x,y) - f_{LP}(x,y)$$

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

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

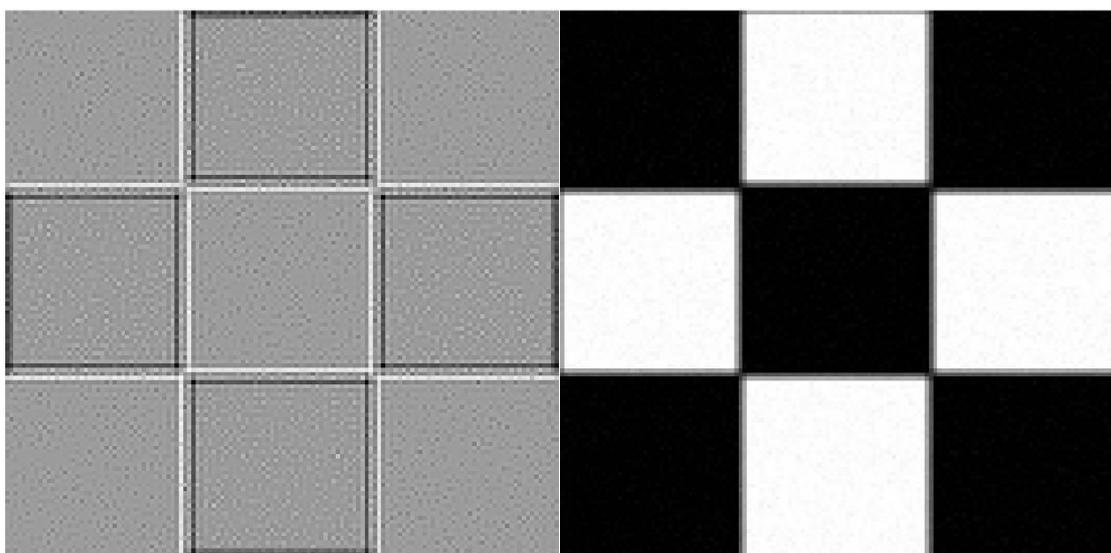
其中低通滤波器采用上一问当中的巴特沃斯二阶低通滤波器，而且经过分析，当 $k=1$ 时，这个增强效果与二阶巴特沃斯高通滤波器的效果是完全一样的。

2. 拉普拉斯



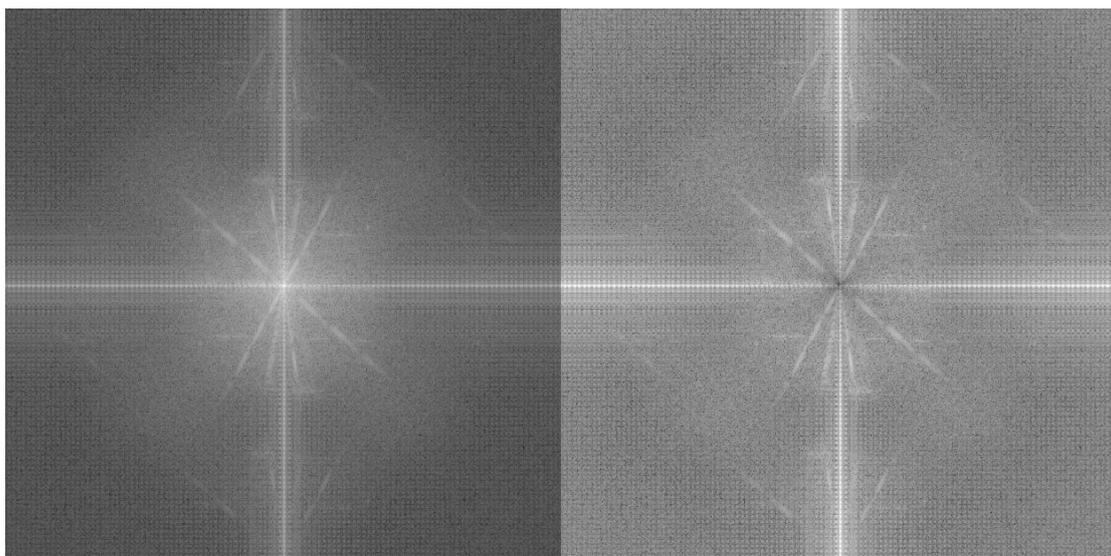
FFT

滤波 FFT



边缘图

增强图



FFT

滤波 FFT

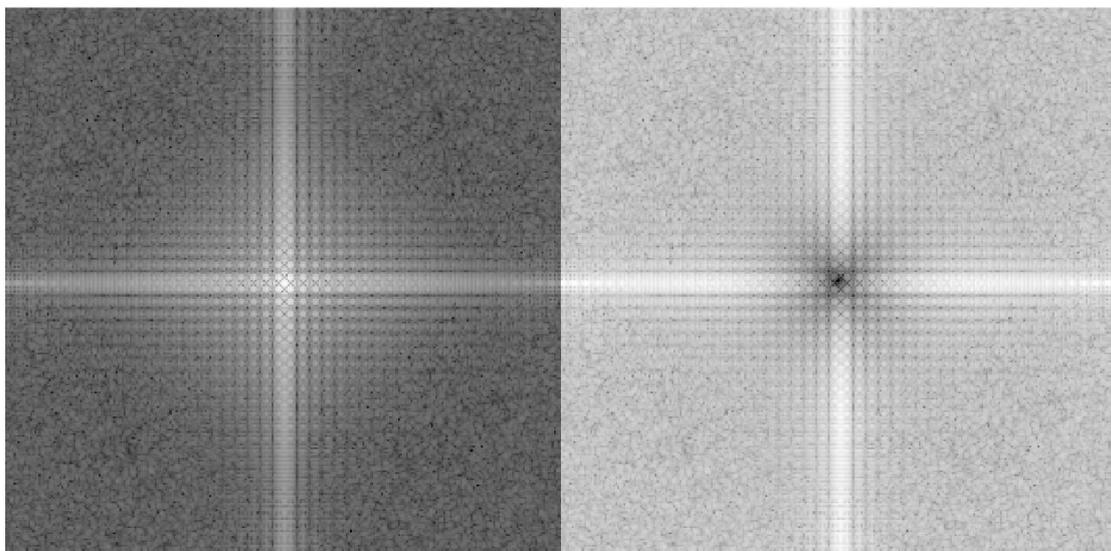


边缘图

增强图

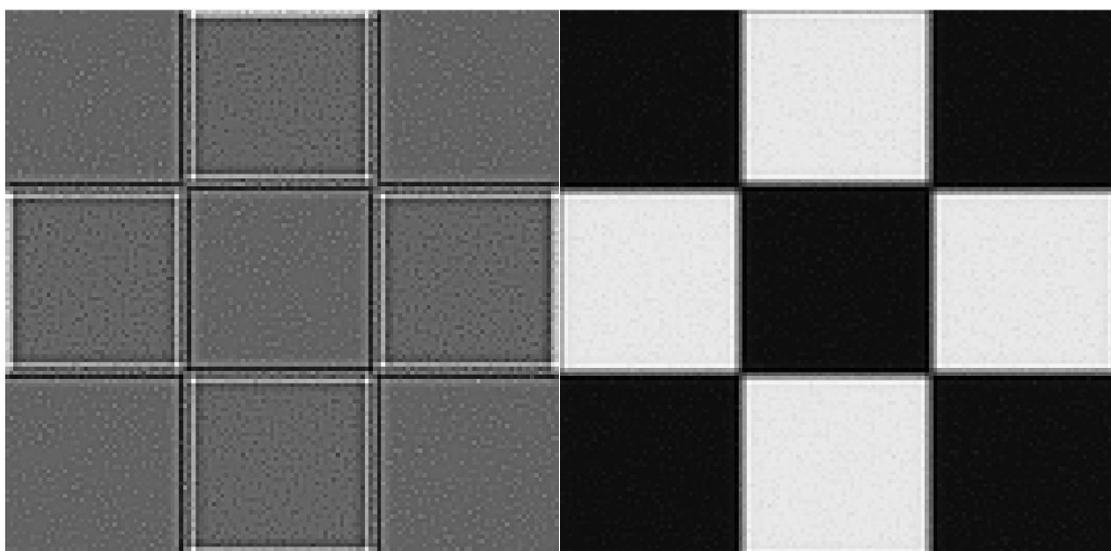
3. Unsharp mask

低通滤波器采用巴特沃斯二阶低通， $D_0=75$



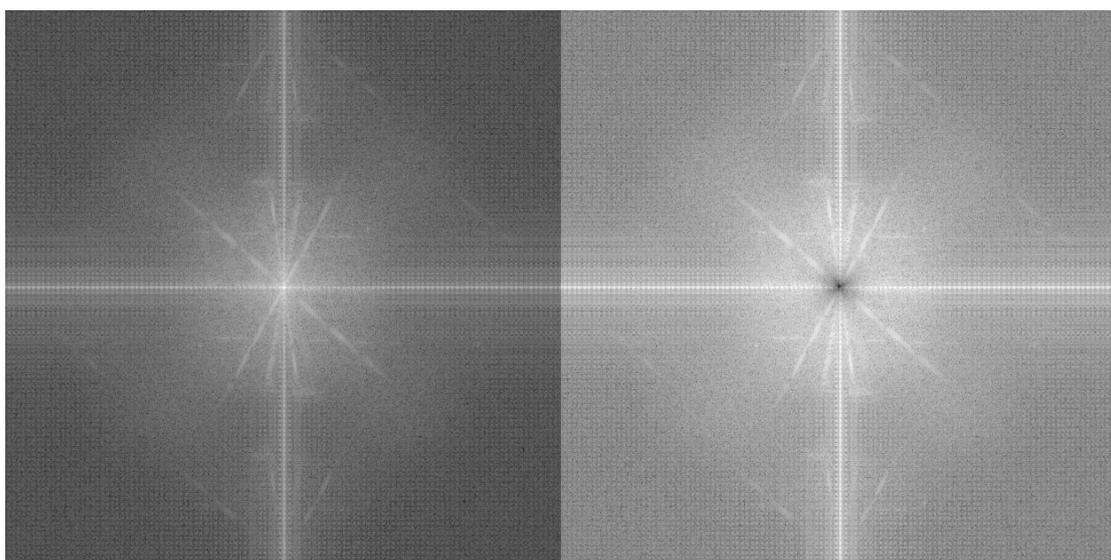
FFT

滤波 FFT



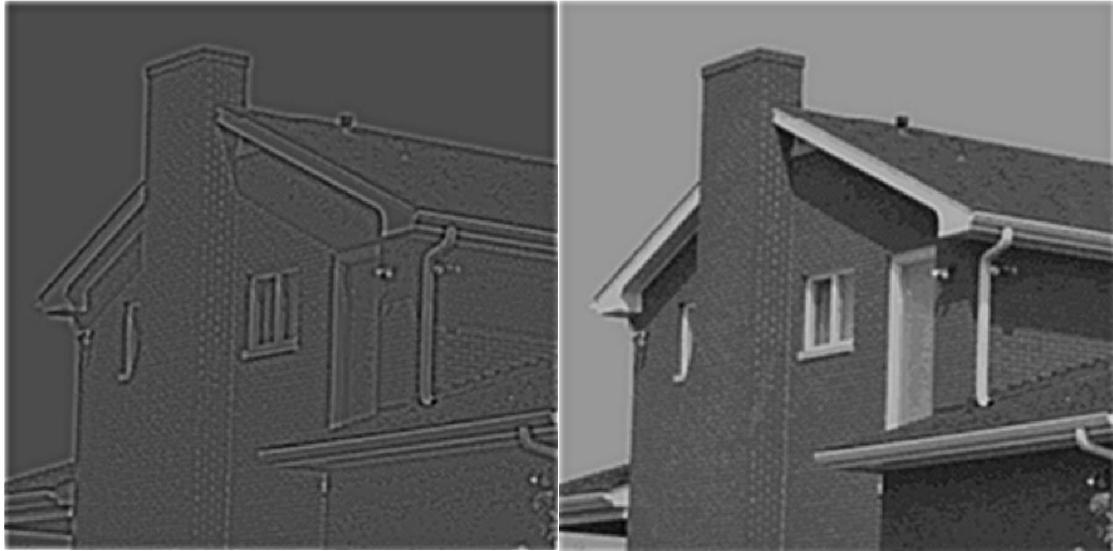
边缘图

增强图



FFT

滤波 FFT

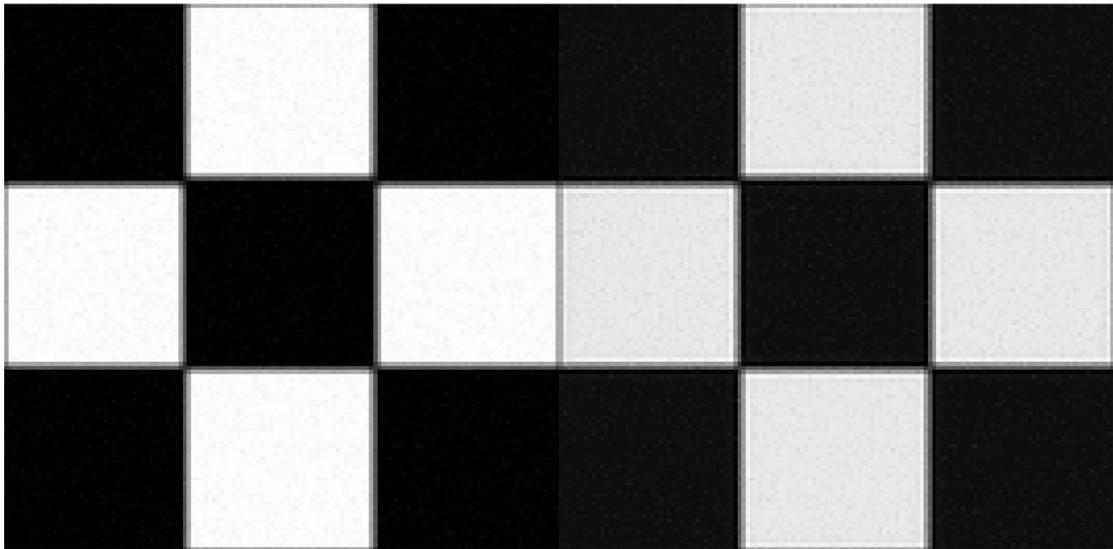


边缘图

增强图

4. 两种滤波方式比较

对比两者可以很明显地看出：unsharp mask 仍然存在平滑区域偏灰的情况，而拉普拉斯滤波可以很好地解决这一问题，这是因为拉普拉斯滤波器不会把零频点整个去掉，而是保留了一部分零频点，这就保证了图像整体的亮度不会变化太大。



拉普拉斯

unsharp mask



拉普拉斯

unsharp mask

四、 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；
试分析高通、低通滤波器在频域和对应的
空域滤波结果是否等效。频域滤波结果如
何等效在空域滤波器。**

空域与频域低通和高通的关系：

(1) 对于空域来说，滤波操作为卷积，而对于频域来说，滤波操作为乘积。

(2) 频域的滤波器为空域滤波器的傅里叶变换，空域的滤波器是频域的傅里叶反变换

(3) 频域增强技术与空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，许多空域增强技术可转化到频域实现，而许多频域增强技术可转化到空域实现。

(4) 频域高通对应空域的锐化，频域的低通对应空域的平滑

等效性：

对于空域模板各处相同的情况，理论分析上来说频域滤波的结果和空域滤波是完全等效的，但是在实际操作中由于计算机的计算误差，无法做到二者的完全等效，但是在很小的误差允许范围内，二者是相同的。但是对于在空域各处分别使用不同模板的情况下无法等效。

频域滤波结果如何等效在空域：

将频域滤波结果进行离散傅里叶反变换即可等效在空域得到空域处理结果。在实际应用中，可先分别对原图和模板进行 FFT 变换将其变换到频域，再对二者进行相乘运算，即可得到对应的频域滤波结果，再对这个结果进行 IFFT 即可得到对应的空域滤波结果。

一些思考：

虽然频域滤波看似简单，却需要进行 FFT 和 IFFT 的两次变换，而空域滤波虽然是卷积计算看似麻烦，但对于编制好的程序来说执行起来是很快。在实际应用中空域直接滤波应用更广泛，但是可以使用频域滤波为空域滤波器的设计提供参考，也可以用于分析不同滤波器的性能。同时，二者在一些性质上也有所差别，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。在实际应用中需要取长补短。

附录

1. 参考文献

[1] Rafael C. Gonzalez (拉斐尔 C. 冈萨雷斯), Richard E. Woods (理查德 E. 伍兹). 数字图像处理(第三版) (英文版). 北京: 电子工业出版社. 2017 年.

2. 源代码

(3) 第一问

画图

```
clear
% pic=imread('test1.pgm');
pic=imread('test2.tif');

%参数设置

D0=75;%滤波器半径

method=2;%滤波器类型 (0 理想; 1 巴特沃斯; 2 高斯)

Order=2;%巴特沃斯阶次

pic=double(pic);
[m,n]=size(pic);
picfft=zeros(2*m,2*n);

for i=1:m %转变原点为中点
    for j=1:n
        picfft(i,j)=pic(i,j)*(-1)^(i+j);
    end
end
end
```

```

picfft=fft2(picfft);
picfftabs=log(abs(picfft));
figure;imshow(picfftabs,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

for x=1:2*m %滤波器
    for y=1:2*n
        i=x-m;j=y-n;
        if method==0
            if sqrt(i^2+j^2) > D0
                picfft(x,y)=0;
            end
        elseif method==1
            picfft(x,y)=picfft(x,y)/(1+(sqrt(i^2+j^2)/D0)^(2*Order));
        elseif method==2
            picfft(x,y)=picfft(x,y)*exp(-(i^2+j^2)/2/D0^2);
        else
            error('方法错误! ');
        end
    end
end
picfftabs=log(abs(picfft));
figure;imshow(picfftabs,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])
NewPic=ifft2(picfft);
NewPic=NewPic(1:m,1:n);

for i=1:m %变回原图
    for j=1:n
        NewPic(i,j)=NewPic(i,j)*(-1)^(i+j);
    end
end
figure;imshow(NewPic,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

```

算功率谱比

```
clear
```

```
%参数设置
```

```
D0=75;%滤波器半径
```

```
method=2;%滤波器类型（0 理想； 1 巴特沃斯； 2 高斯）
```

```
Order=2;%巴特沃斯阶次
```

```
m=512; n=512;
```

```
H=zeros(2*m,2*n);
```

```
for x=1:2*m %滤波器
```

```
    for y=1:2*n
```

```
        i=x-m;j=y-n;
```

```
        if method==0
```

```
            if sqrt(i^2+j^2) < D0
```

```
                H(x,y)=1;
```

```
            end
```

```
        elseif method==1
```

```
            H(x,y)=1/(1+(sqrt(i^2+j^2)/D0)^(2*Order));
```

```
        elseif method==2
```

```
            H(x,y)=exp(-(i^2+j^2)/2/D0^2);
```

```
        else
```

```
            error('方法错误! ');
```

```
        end
```

```
    end
```

```
end
```

```
H=H.^2;
```

```
figure;imshow(H,[]);set(gca,'position',[0 0 1 1]);
```

```
set(gcf,'position',[500,280,500,500])
```

(4) 第二问

画图

```
clear
% pic=imread('test3_corrupt.pgm');
pic=imread('test4 copy.bmp');

%参数设置

D0=75;%滤波器半径

method=2;%滤波器类型 (0 理想; 1 巴特沃斯; 2 高斯)

Order=2;%巴特沃斯阶次

pic=double(pic);
[m,n]=size(pic);
picfft=zeros(2*m,2*n);

for i=1:m %转变原点为中点
    for j=1:n
        picfft(i,j)=pic(i,j)*(-1)^(i+j);
    end
end
picfft=fft2(picfft);
picfftabs=log(abs(picfft));
figure;imshow(picfftabs,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

for x=1:2*m %滤波器
    for y=1:2*n
        i=x-m;j=y-n;
        if method==0
            if sqrt(i^2+j^2) < D0
                picfft(x,y)=0;
            end
        end
    end
end
```

```

        end
    elseif method==1
        picfft(x,y)=picfft(x,y)/(1+(D0/sqrt(i^2+j^2))^(2*Order));
    elseif method==2
        picfft(x,y)=picfft(x,y)*(1-exp(-(i^2+j^2)/2/D0^2));
    else
        error('方法错误! ');
    end
end
end
picftabs=log(abs(picfft));
figure;imshow(picftabs,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])
NewPic=ifft2(picfft);
NewPic=NewPic(1:m,1:n);
for i=1:m %变回原图
    for j=1:n
        NewPic(i,j)=NewPic(i,j)*(-1)^(i+j);
    end
end
NewPic=real(NewPic); %边缘
figure;imshow(NewPic,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])
enhance=pic+NewPic; %增强
figure;imshow(enhance,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

```

算功率谱比

clear

%参数设置

```

D0=75;%滤波器半径

method=0;%滤波器类型（0 理想； 1 巴特沃斯； 2 高斯）

Order=2;%巴特沃斯阶次

m=512; n=512;
H=zeros(2*m,2*n);

for x=1:2*m %滤波器
    for y=1:2*n
        i=x-m;j=y-n;
        if method==0
            if sqrt(i^2+j^2) > D0
                H(x,y)=1;
            end
        elseif method==1
            H(x,y)=1/(1+(D0/sqrt(i^2+j^2))^(2*Order));
        elseif method==2
            H(x,y)=1-exp(-(i^2+j^2)/2/D0^2);
        else
            error('方法错误! ');
        end
    end
end
H=H.^2;
figure;imshow(H,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

```

(5) 第三问

```

clear
pic=imread('test3_corrupt.pgm');
% pic=imread('test4_copy.bmp');

```

%参数设置

D0=75;%滤波器半径

method=1;%滤波器类型 (1 拉普拉斯; 2 unsharp mask)

Order=2;%巴特沃斯阶次

pic=double(pic);

[m,n]=size(pic);

picfft=zeros(2*m,2*n);

for i=1:m %转变原点为中点

 for j=1:n

 picfft(i,j)=pic(i,j)*(-1)^(i+j);

 end

end

picfft=fft2(picfft);

% picftabs=log(abs(picfft));

% figure;imshow(picftabs,[]);set(gca,'position',[0 0 1 1]);

% set(gcf,'position',[500,280,500,500])

for x=1:2*m %滤波器

 for y=1:2*n

 i=x-m;j=y-n;

 if method==1

 picfft(x,y)=-4*pi^2*(i^2+j^2)*picfft(x,y);

 elseif method==2

picfft(x,y)=picfft(x,y)*(1-1/(1+(sqrt(i^2+j^2)/D0)^(2*Order)));

 else

 error('方法错误! ');

 end

 end

end

```

picfftabs=log(abs(picfft));
figure;imshow(picfftabs,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])
NewPic=ifft2(picfft);
NewPic=NewPic(1:m,1:n);

for i=1:m %变回原图
    for j=1:n
        NewPic(i,j)=NewPic(i,j)*(-1)^(i+j);
    end
end
NewPic=real(NewPic); %边缘

figure;imshow(NewPic,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])
if method==1
    enhance=pic-10^(-7)*NewPic;
else
    enhance=pic+NewPic; %增强
end
figure;imshow(enhance,[]);set(gca,'position',[0 0 1 1]);
set(gcf,'position',[500,280,500,500])

```