

数字图像处理

第五次作业



摘要

本报告通过了解各种频域滤波器，分别设计低通以及高通滤波器，需要调整以选择一个最佳的半径，计算相应的功率谱比。所使用的编译环境是Matlab2018。

杨致远

电信钱51

2150405061

2018/5/3

## 任务1—频域低通滤波器

**实验介绍：**

1、Butterworth低通滤波器的原理是：

其中：，是截止频率

2、Gaussian低通滤波器的原理是：

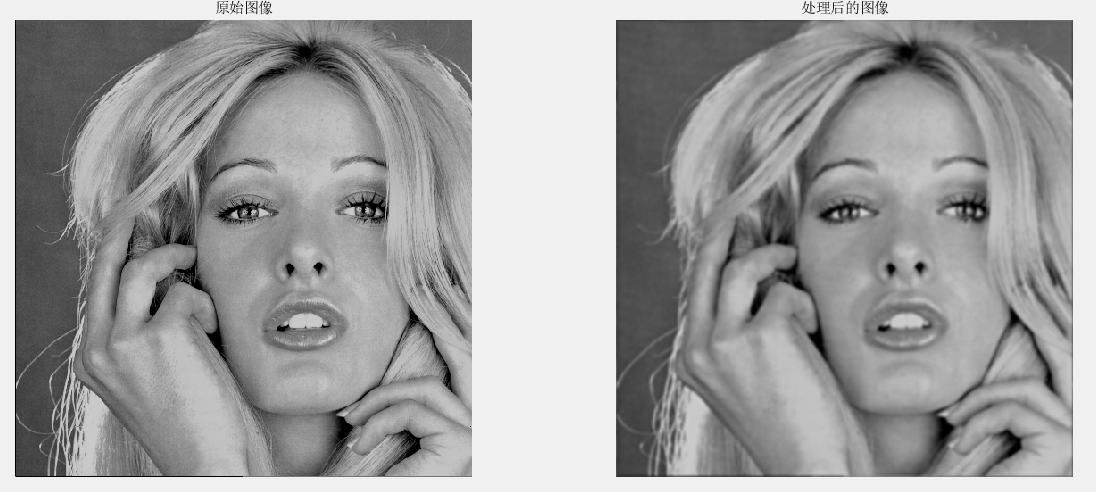
**结果展示：**



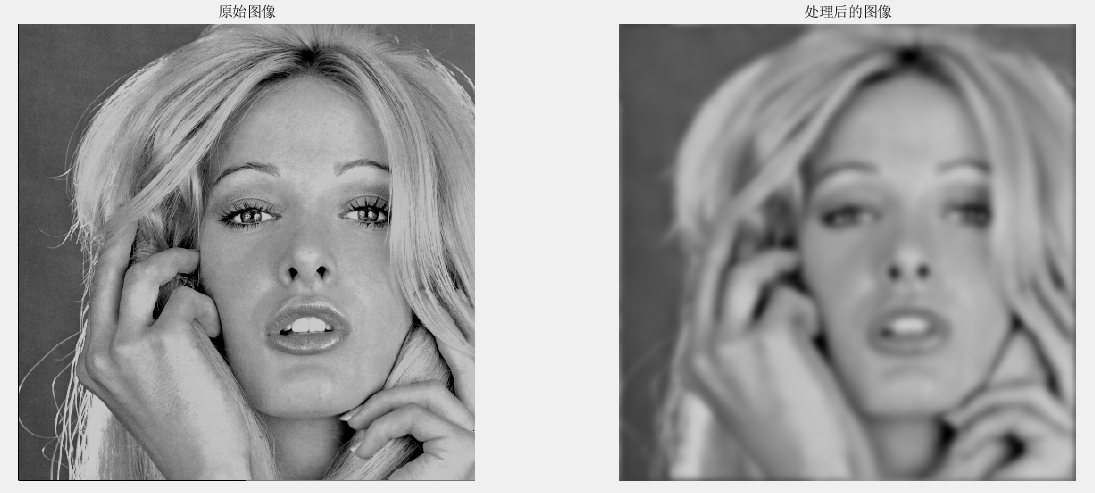
巴特沃斯滤波器 D0=75 功率谱比为： 0.99575



巴特沃斯滤波器 D0=25 功率谱比为： 0.97408



巴特沃斯滤波器 D0=75 功率谱比为： 0.99155



巴特沃斯滤波器 D0=25 功率谱比为： 0.98036



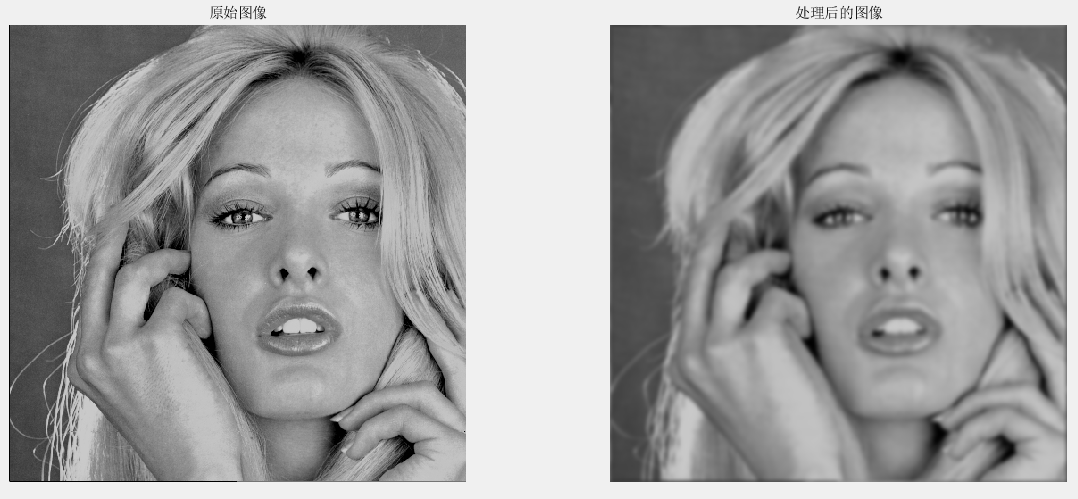
高斯滤波器 D0=75 功率谱比为： 0.99579



高斯滤波器 D0=25 功率谱比为： 0.97780



高斯滤波器 D0=75 功率谱比为： 0.99281



高斯滤波器 D0=25 功率谱比为： 0.98172

**结果讨论：**

单纯的巴特沃斯和高斯比较：由上图可以看出，滤波器截止频率越小，图像变得越老越模糊，滤波器的功率谱也越来越小。这是因为当截止频率设的越来越小，它所虑过的高频信息越来越多，留下的低频信息越来越少。

巴特沃斯和高斯进行比较： 在相同的截止频率下，高斯滤波器的平滑效果更好。所以平滑效果的排列顺序是：IHPF<BHPF<GHPF。虽然高斯滤波器效果更好，但因为它是指数型的，所以在计算时计算量更大，速度更慢。

## 任务2—频域高通滤波

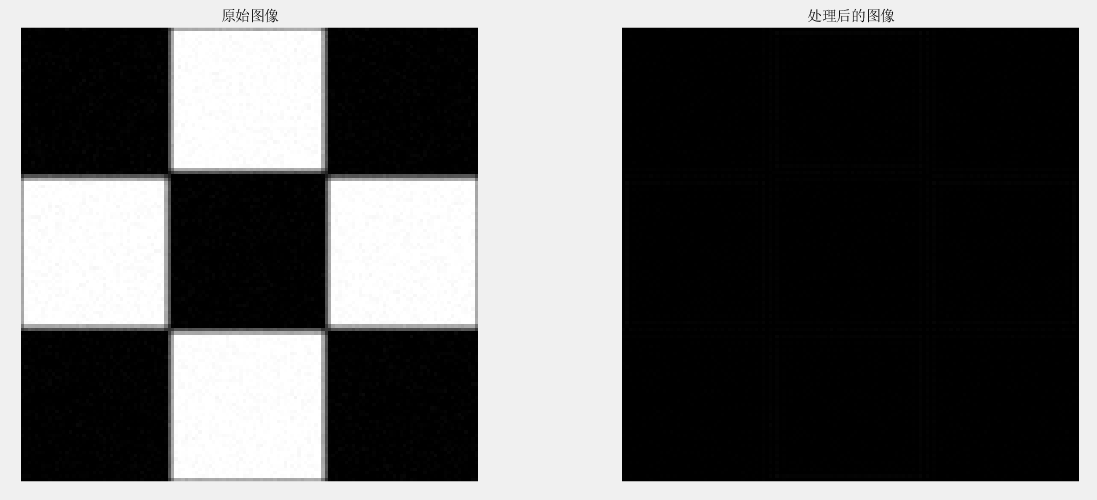
**实验介绍：**

1、Butterworth滤波器的原理是：

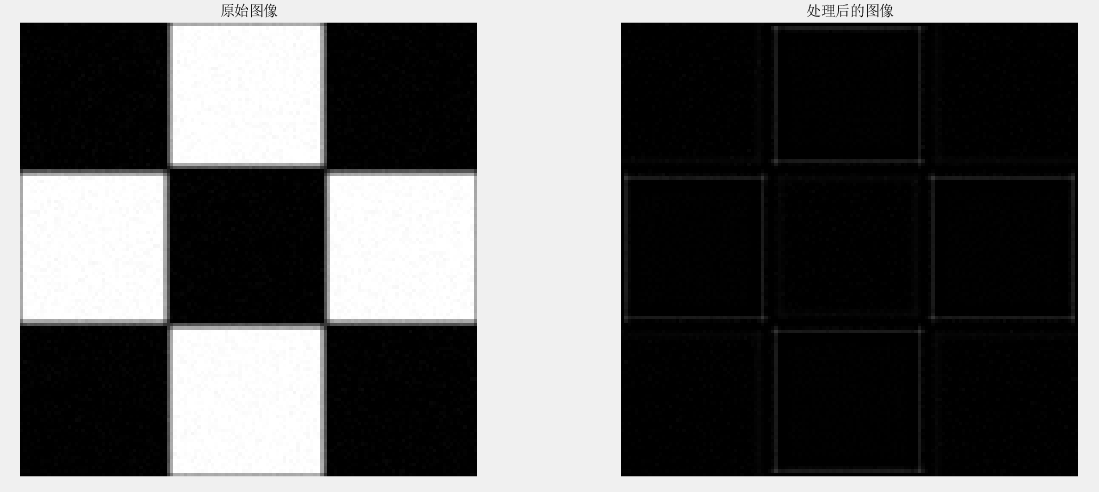
其中：，是截止频率

2、Gaussian低通滤波器的原理是：

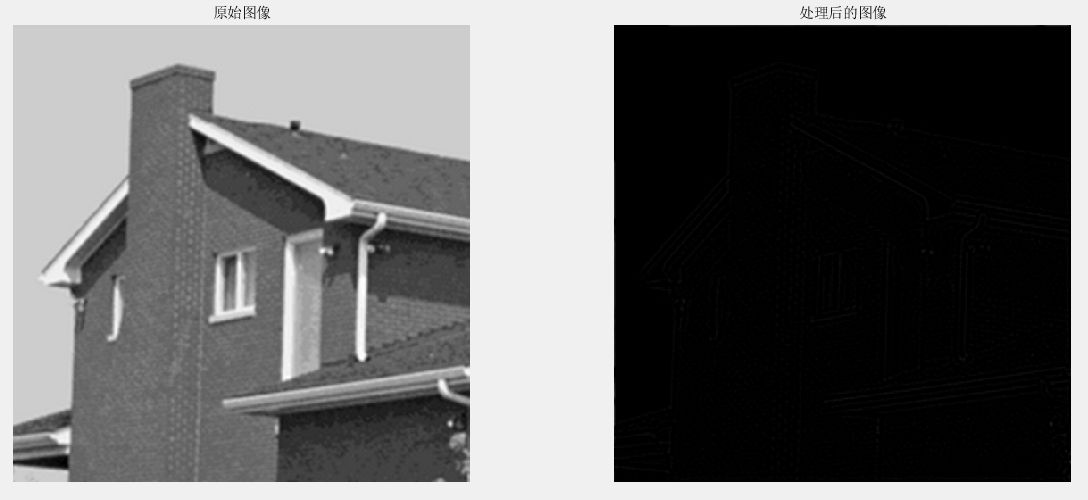
**结果展示：**



巴特沃斯滤波器 D0=75 功率谱比为： 0.00005



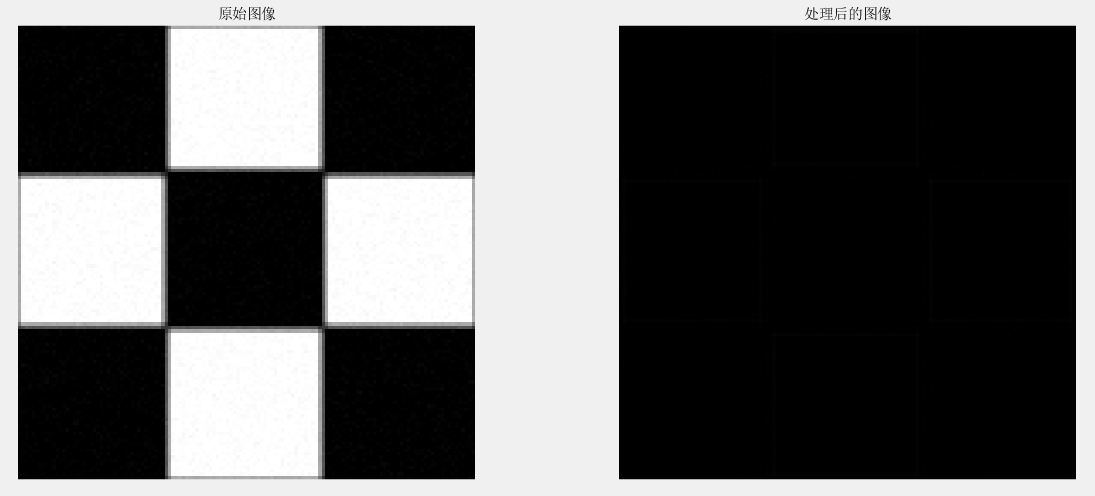
巴特沃斯滤波器 D0=25 功率谱比为： 0.00221



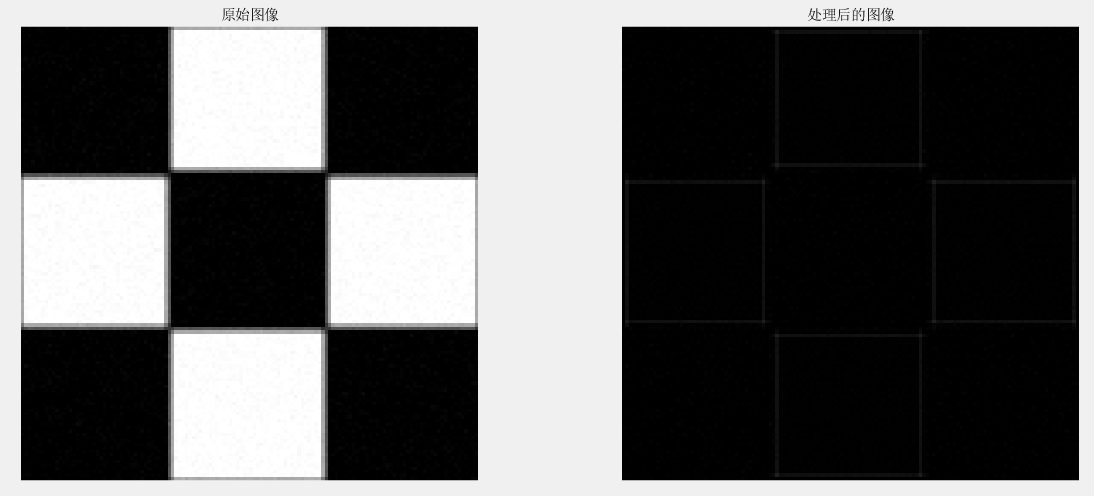
巴特沃斯滤波器 D0=75 功率谱比为： 0.00074



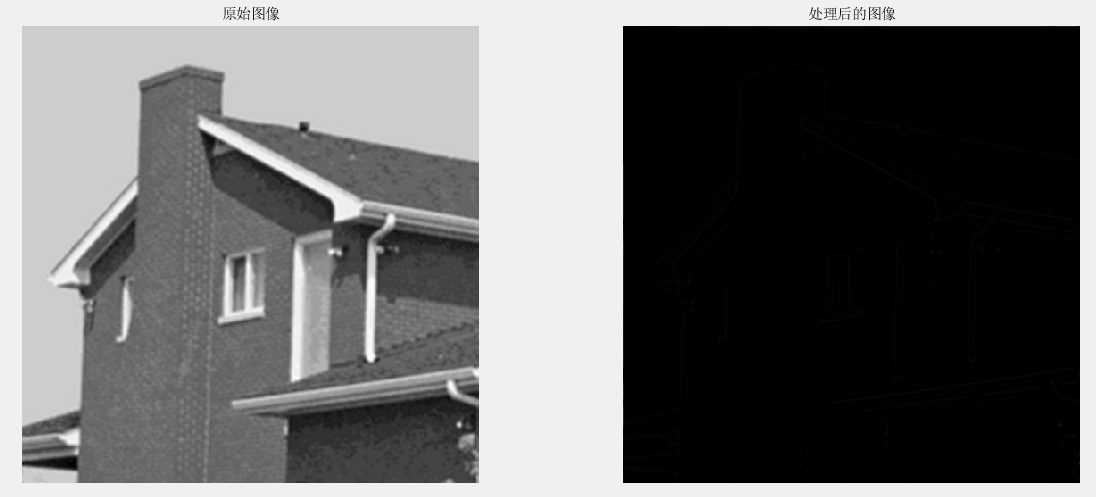
巴特沃斯滤波器 D0=25 功率谱比为：0.00707



高斯滤波器 D0=75 功率谱比为：0.00002



高斯滤波器 D0=25 功率谱比为：0.00077



高斯滤波器 D0=75 功率谱比为：0.00031



高斯滤波器 D0=25 功率谱比为：0.00303

**结果讨论：**

单纯的巴特沃斯和高斯高通滤波器比较：发现当截止频率D0越小时，边界越明显，功率谱越大，这与低通是相反的，但道理是一致的，因为当高通截止频率越小时，可以通过的高通信息就越多。当截止频率为75时，几乎就看不到图像了。

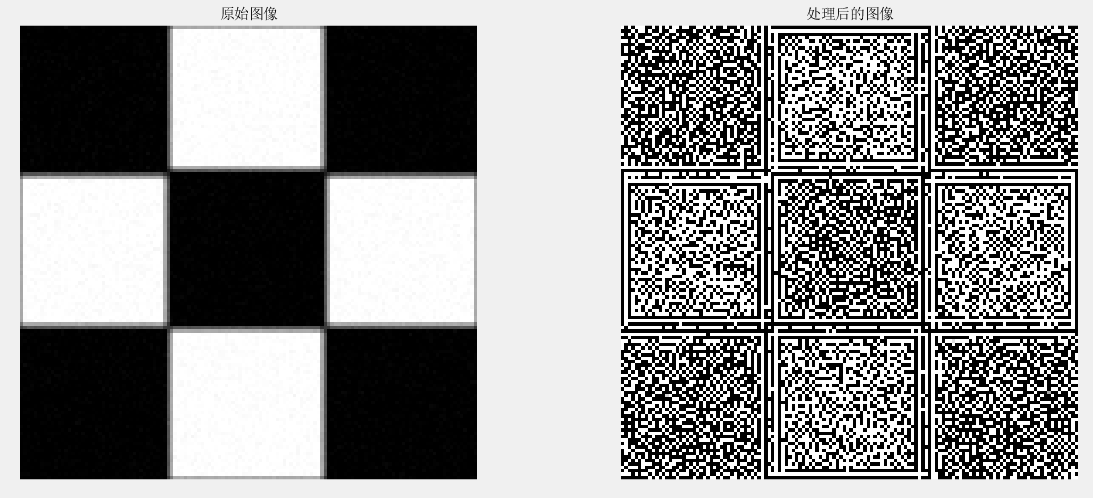
比较巴特沃斯滤波器和高斯滤波器发现高斯滤波器的滤过效果更好。与上面的低通滤波器一样，高斯滤波器虽然效果好，但计算量太大了。

## 任务3—其他高通滤波器

**实验介绍：**

1. 拉普拉斯算子的原理是：
2. Unmask原理是：

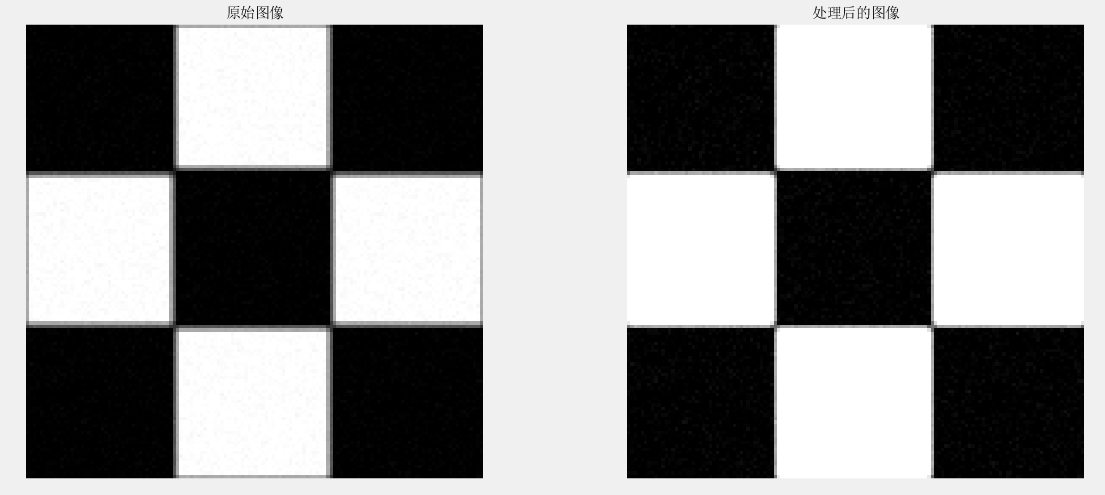
**结果展示：**



拉普拉斯滤波器 功率谱比为：16795072.17003



拉普拉斯滤波器 功率谱比为：846231806.05550



Unmask滤波器 功率谱比为：3.99856



Unmask滤波器 功率谱比为：3.99673

**结果讨论：**

这两个高通滤波器与之前的不一样，图像的信息得到了很好的保留，且边缘被增强了。拉普拉斯高通滤波器将原始图像加回了滤波后的图像中，所以没有除去傅里叶变换中零频率成分，使得最后的结果图像变亮了。但也加入了一定的噪声，出现了多重轮廓。

## 任务4—讨论交流

空域滤波器做一个傅里叶变换就变成了频域滤波器，同样的，频域滤波器做一个傅里叶反变换就成了空域滤波器。而他们之间还有一个变换关系：。频域和空域是可以紧密变换的。

空域滤波有平滑滤波和锐化滤波两种，平滑滤波是要除去不规则的噪声，在频域中，噪声具有较高的频率，所以在频域中是用低通滤波器实现平滑效果。所以空域平滑滤波=频域低通滤波。锐化滤波是为了增强轮廓，边缘和轮廓在频域中具有较高的频率，所以这时就要用一个频域高通滤波器。所以空域锐化滤波=频域高通滤波。

两者的优劣对比：因为图像的直观视觉效果与频域的关系更加紧密，所以在频域变换更加值观，但随之带来的就是复杂的实现计算。而空域滤波在具体实现上和硬件设计上更简单。还有一点是空域是针对部分像素进行的变换操作，而频域是对全局进行的。

## 代码：

%巴特沃斯低通滤波器

clc;clear;

picture = imread('test2.tif');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

n = 2; %巴特沃斯滤波器的阶数

D0 = 75; %截止频率半径

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = 1 / (1 + (D(u, v) / D0)^(2 \* n));

G(u, v) = H(u, v) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

%高斯低通滤波器

clc;clear;

picture = imread('test2.tif');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

n = 2; %巴特沃斯滤波器的阶数

D0 = 25; %截止频率半径

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = exp(- D(u, v)^2 / (2 \* D0)^2 );

G(u, v) = H(u, v) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

%%巴特沃斯高通滤波器

clc;clear;

picture = imread('test4 copy.bmp');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

n = 2; %巴特沃斯滤波器的阶数

D0 = 25; %截止频率半径

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = 1 / (1 + (D0 / D(u, v))^(2 \* n));

G(u, v) = H(u, v) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

%%高斯高通滤波器

clc;clear;

picture = imread('test4 copy.bmp');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

n = 2; %巴特沃斯滤波器的阶数

D0 = 25; %截止频率半径

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = 1 - exp(- D(u, v)^2 / (2 \* D0)^2 );

G(u, v) = H(u, v) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

%%拉普拉斯高通滤波器

clc;clear;

picture = imread('test4 copy.bmp');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

c = 1;

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = 1 + c \* 4 \* pi^2 \* D(u, v)^2;

G(u, v) = H(u, v) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

%%非锐化高通滤波器

clc;clear;

picture = imread('test4 copy.bmp');

subplot(1,2,1);

imshow(picture);

title('原始图像');

%频域滤波，先要进行傅里叶变换

picture = double(picture);

FourierTransform = fft2(picture);

FourierTransform = fftshift(FourierTransform);

[x, y] = size(FourierTransform);

D = zeros(x, y);

H = zeros(x, y);

G = zeros(x, y);

%设定参数

k1 = 1;

k2 = 1;

D0 = 100; %截止频率半径

for u = 1:x

for v = 1:y

D(u, v) = sqrt((u - (x/2))^2 + (v - (y/2))^2);

H(u, v) = 1 - exp(- D(u, v)^2 / (2 \* D0)^2 );

G(u, v) = (k1 + k2 \* (1 - H(u, v))) \* FourierTransform(u, v);

end

end

picture1 = ifftshift(G);

picture1 = ifft2(picture1);

picture1 = uint8(real(picture1));

subplot(1,2,2);

imshow(picture1);

title('处理后的图像');

%计算功率谱

s = 0;

s1 = 0;

for u = 1:x

for v = 1:y

L1 = (abs(G(u, v)))^2;

s1 = s1 + L1;

L = (abs(FourierTransform(u, v)))^2;

s = s + L;

end

end

fprintf('功率谱比为： %8.5f\n',s1/s)

## 参考文献

[1] 冈萨雷斯.数字图像处理（第三版）北京：电子工业出版社，2011

[2] 维基百科