实验二 图像增强

实验内容：

一、空间域图像增强

1 实现对比度拉伸变换（二选一来做）

（1）根据书上的原理，实现对比度拉伸变换，并对图像进行处理，把原图和处理后的结果显示在一个窗口中。

$$g=\begin{cases}\alpha\cdot f & 0\le f<a\\ \beta\cdot(f-a)+g_a & a\le f<b\\ \gamma\cdot(f-b)+g_b & b\le f<L\end{cases}$$

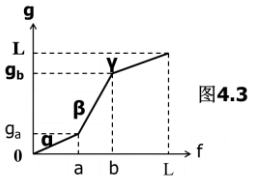


图4.3

双重循环遍历判断,两段压缩,中间拉伸,方法一比较麻烦

（2）根据下面的公式对图像进行处理，自行选择不同的参数 m 和 E，把原图和处理后的结果显示在一个窗口中。

$$s=T(r)=\frac{1}{1+(m/r)^E}$$

R 为输入图像,S 输入图像,需要选择合适的 M 与 E

对比度提高与拉伸,提高对比度

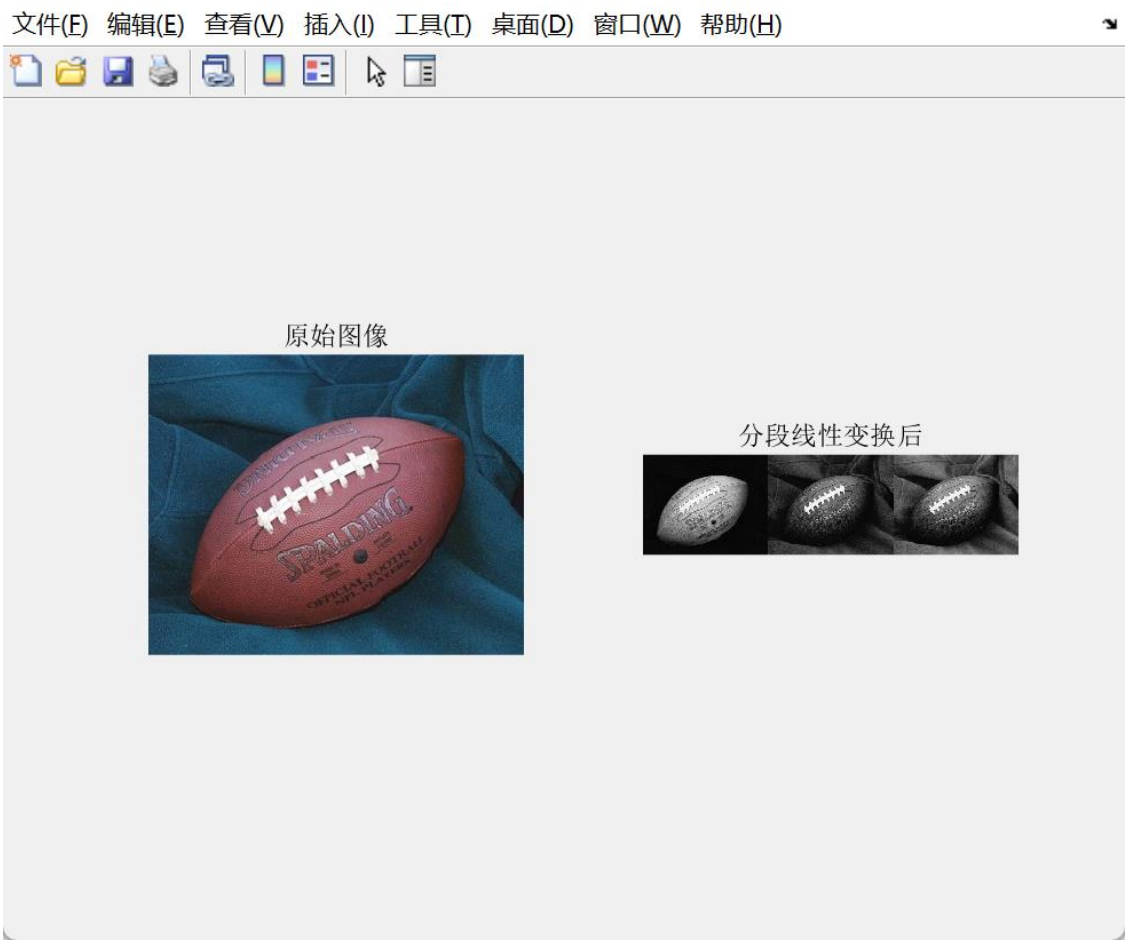
对比度太高会趋向二值图

请将实验代码贴在此处：

```
%% 1.1 对比度拉伸变换
f = imread('football.jpg');
[m,n] = size(f);
f = im2double(f);

% 定义变换参数
alpha = 0.3;  % 第一段斜率
belta = 1.5;  % 第二段斜率
gamma = 0.5;  % 第三段斜率
a = 0.2;      % 第一段分界点
b = 0.7;      % 第二段分界点

% 初始化输出图像
```

```
g = ones(m,n);

% 分段线性变换
for i = 1:m
    for j = 1:n
        if(f(i,j) < a && f(i,j) >= 0)
            g(i,j) = f(i,j) * alpha;
        end
        if(f(i,j) < b && f(i,j) >= a)
            g(i,j) = (f(i,j) - a) * belta + a * alpha;
        end
        if(f(i,j) < 1 && f(i,j) >= b)
            g(i,j) = (f(i,j) - b) * gamma + belta * (b-a) + a * alpha;
        end
    end
end

% 显示结果
figure
subplot(121), imshow(f); title('原始图像');
subplot(122), imshow(g); title('分段线性变换后');
```

请将运行结果贴在此处：



2 利用直方图进行图像增强：

直方图均衡与直方图匹配(直方图的形状进行定制与匹配)

Imhist 生成直方图函数

思想,所有的灰度级都有,对比度高

（1）利用函数 histeq 对图像’pout.tif’进行直方图均衡，并显示均衡后的结果。

（2）使用函数 histeq()对灰度图像’pout.tif’做直方图规定化处理，要匹配直方图的图像分别用’coins.png’和’circuit.tif’，并且要求将原图和规定化后的两幅图像显示在同一个图像窗口中，原始图像的直方图和直方图规定化后图像的直方图显示在另一个图像窗口中。

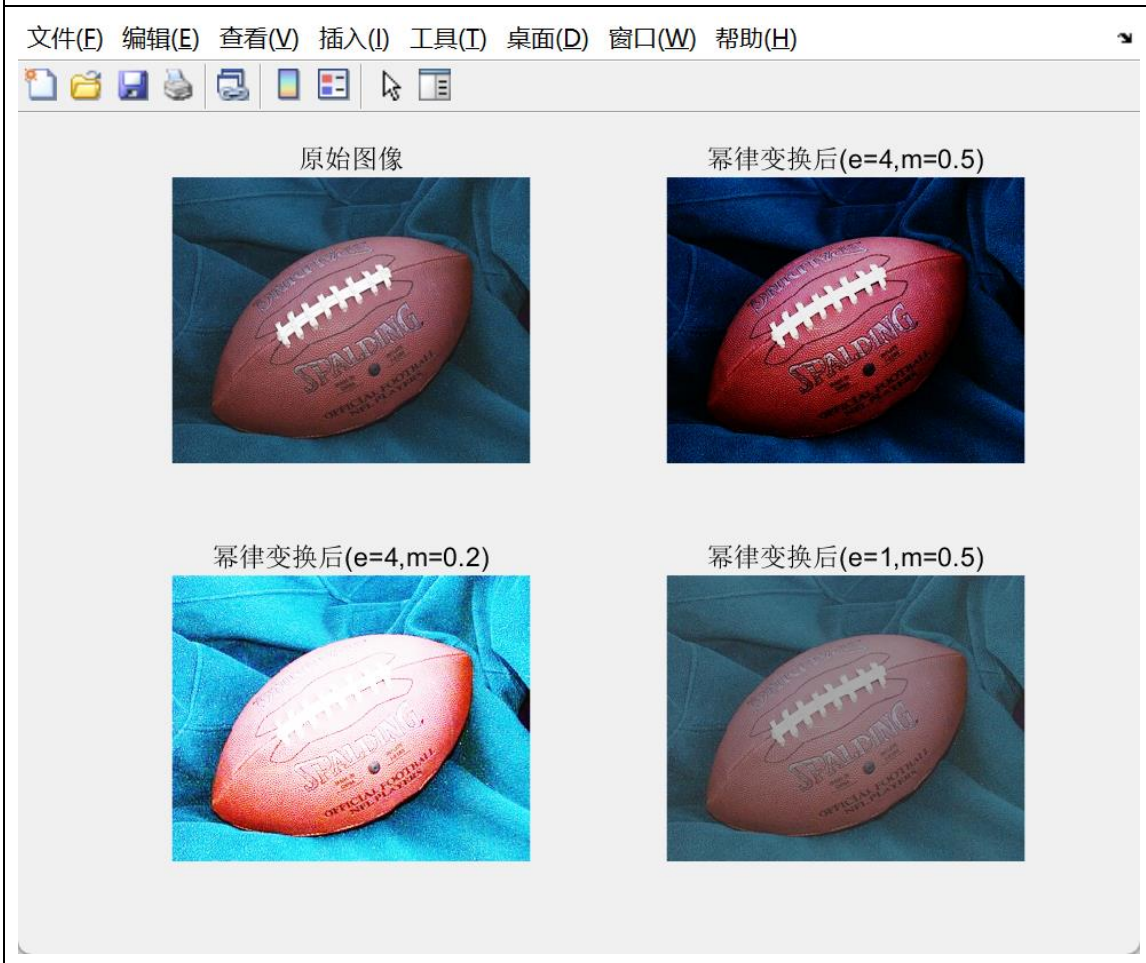
请将实验代码贴在此处：

```
%% 2 直方图均衡和匹配
%% 2.1 直方图均衡化
% 读取并转换图像
f = imread('football.jpg');
f = im2double(f);
r1 = histeq(f);

% 显示均衡化后的图像与直方图
figure
subplot(221), imshow(f); title('原始图像');
subplot(222), imhist(f); title('原始直方图');
subplot(223), imshow(r1); title('均衡化后图像');
subplot(224), imhist(r1); title('均衡化后直方图');

%% 2.3 直方图匹配改进
% 读取源图像和目标图像
f = imread('pout.tif');
f1 = imread('coins.png');
figure
subplot(121),imshow(f);title("原始图像");
subplot(122),imshow(f1);title("用于匹配的图像")

% % 转换为双精度并确保图像大小一致
% f = im2double(f);
% f1 = im2double(f1);
% if ~isequal(size(f), size(f1))
%     f1 = imresize(f1, size(f));
% end

% % 对图像进行预处理 - 对比度调整
% f = imadjust(f);  % 自动调整对比度
```

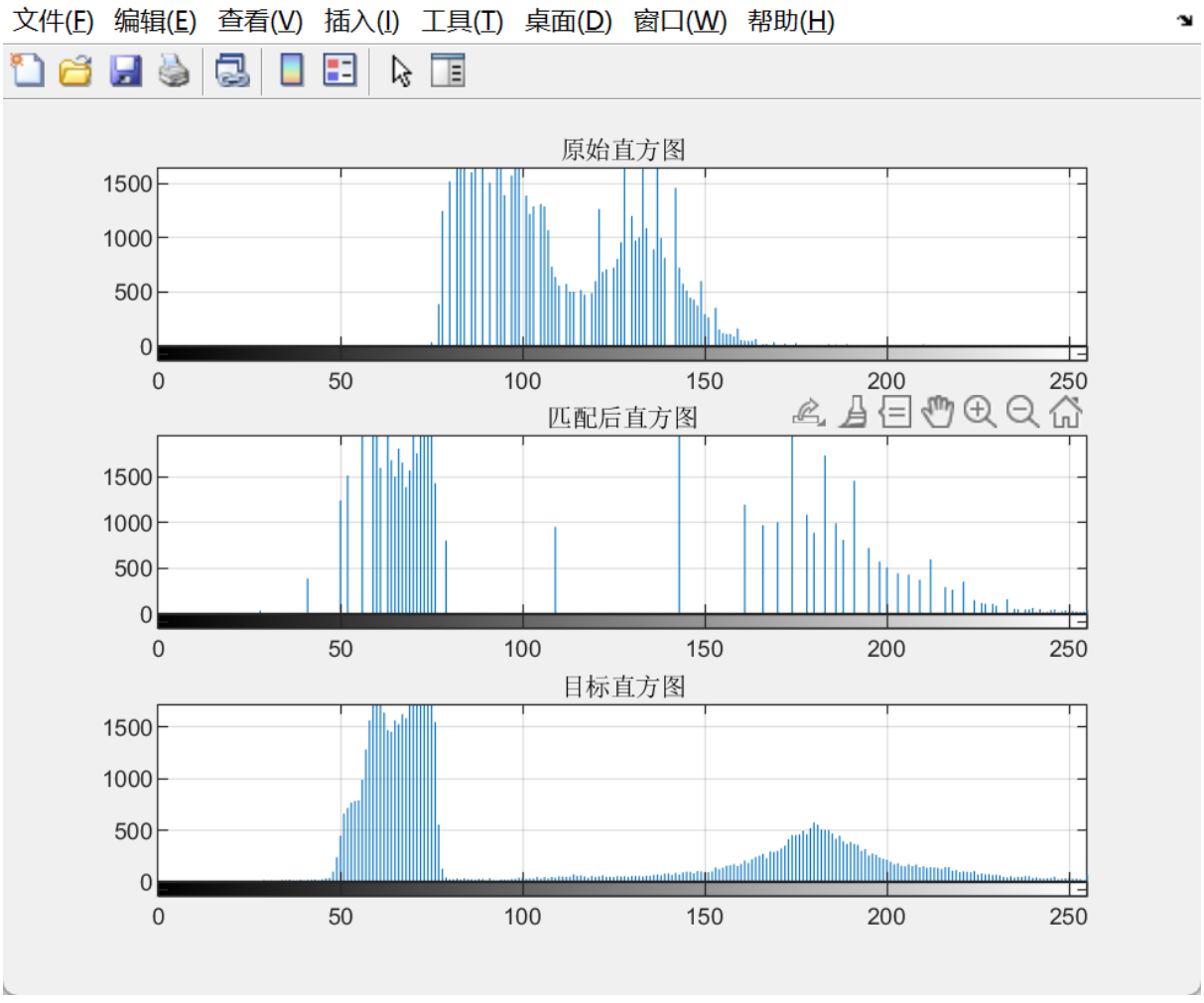
```
% f1 = imadjust(f1);

% 执行直方图匹配
g = histeq(f, imhist(f1));

% 计算评估指标
mse = immse(f, g);  % 均方误差
ssim_val = ssim(f, g);  % 结构相似性
correlation = corr2(f, g);  % 相关系数

% 显示匹配结果
figure('Name', '直方图匹配结果对比', 'NumberTitle', 'off');
subplot(121)
imshow(f)
title('原始图像')
subplot(122)
imshow(g)
title('直方图匹配后')

% 显示直方图对比
figure('Name', '直方图对比', 'NumberTitle', 'off');
subplot(311)
imhist(f)
title('原始直方图')
grid on
subplot(312)
imhist(g)
title('匹配后直方图')
grid on
subplot(313)
imhist(f1)
title('目标直方图')
grid on

% 显示评估指标
fprintf('\n 直方图匹配评估指标:\n');
fprintf('均方误差 (MSE): %f\n', mse);
fprintf('结构相似性 (SSIM): %f\n', ssim_val);
fprintf('相关系数: %f\n', correlation);

% 显示累积直方图对比
figure('Name', '累积直方图对比', 'NumberTitle', 'off');
[counts_f, x] = imhist(f);
[counts_g, ~] = imhist(g);
[counts_f1, ~] = imhist(f1);
```

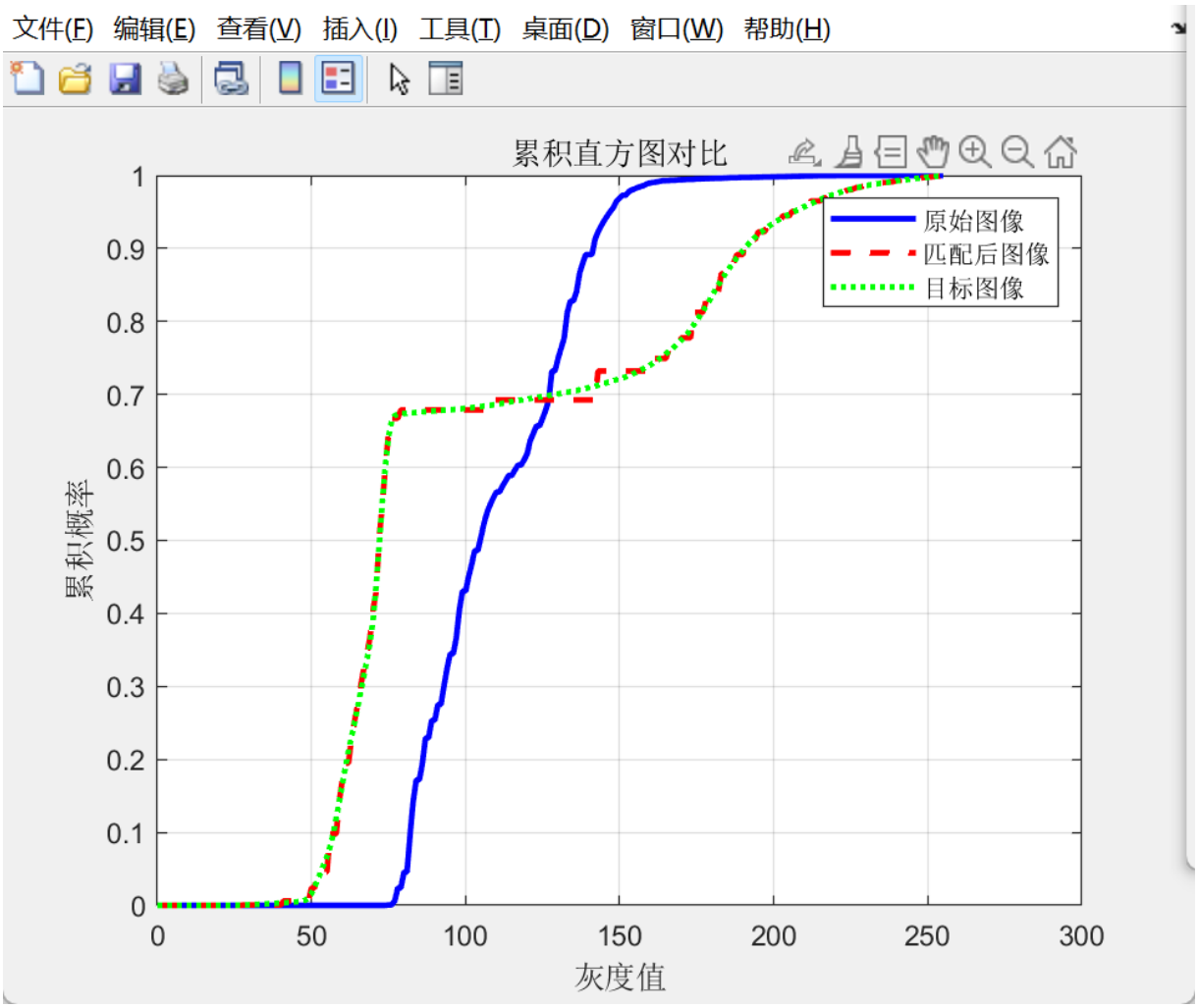
```
cdf_f = cumsum(counts_f) / sum(counts_f);
cdf_g = cumsum(counts_g) / sum(counts_g);
cdf_f1 = cumsum(counts_f1) / sum(counts_f1);

plot(x, cdf_f, 'b-', 'LineWidth', 2);
hold on;
plot(x, cdf_g, 'r--', 'LineWidth', 2);
plot(x, cdf_f1, 'g:', 'LineWidth', 2);
grid on;
legend('原始图像', '匹配后图像', '目标图像');
title('累积直方图对比');
xlabel('灰度值');
ylabel('累积概率');
```

请将运行结果贴在此处：




文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始图像
用于匹配的图像
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始图像


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始直方图
匹配后直方图
目标直方图
0
500
1000
1500
0
50
100
150
200
250



直方图匹配评估指标:

评估指标	值
均方误差 (MSE)	1345.803279
结构相似性 (SSIM)	0.711337
相关系数	0.913577

1. 结构相似性中等偏高（0.71），说明图像结构大致保留；
2. 相关性很强（0.91），说明亮度、灰度变化趋势接近；
3. 但 MSE 偏大（1345），说明局部像素值差异仍较大，可能有噪声、模糊或位移误差。

3 用空间滤波完成对图像的去噪操作：

图像的平滑与图像的去噪

（1）先使用函数 imnoise 对图像添加高斯噪声和椒盐噪声，其语法格式如下：

g=imnosie(f,'gaussian',0,0.01);其中 f 为输入图像，g 为添加了噪声的图像

g=imnoise(f,'salt & pepper',0.3),其中 0.3 为噪声的比例

（2）使用函数 fspecial 生成 3×3 的均值滤波器；语法如下：

h=fspecial('average',3);其中 3 表示模板大小为 3×3；

（3）用均值模板对（1）中添加了噪声的图像进行卷积滤波，并且要求将原图和滤波后的两幅图像显示在同一个图像窗口中。语法如下：

g=imfilter(f,h);其中 g 表示滤波结果

（4）用中值滤波器对（1）中添加了噪声的图像进行中值滤波，并且要求将原图和滤波后的两幅图像显示在同一个图像窗口中。语法如下：

中值滤波不需要做卷积不用滤波器,消除椒盐噪声的效果好

g=medfit2(f);其中 f 为添加了噪声的图像，g 为中值滤波的结果

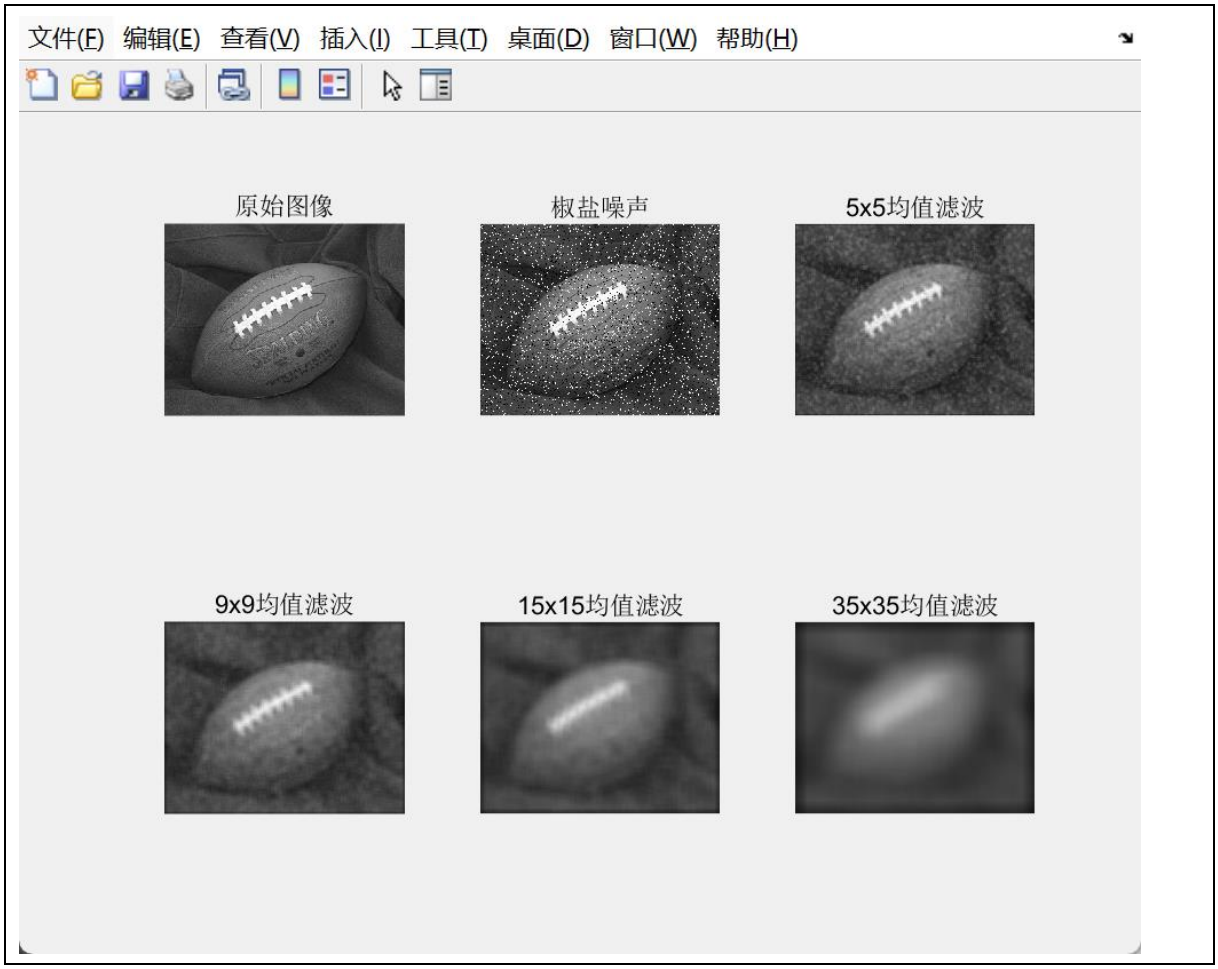
（5）选做：使用 5×5、9×9、15×15、35×35 的均值滤波器对添加了噪声的图像进行滤波，观察实验结果。

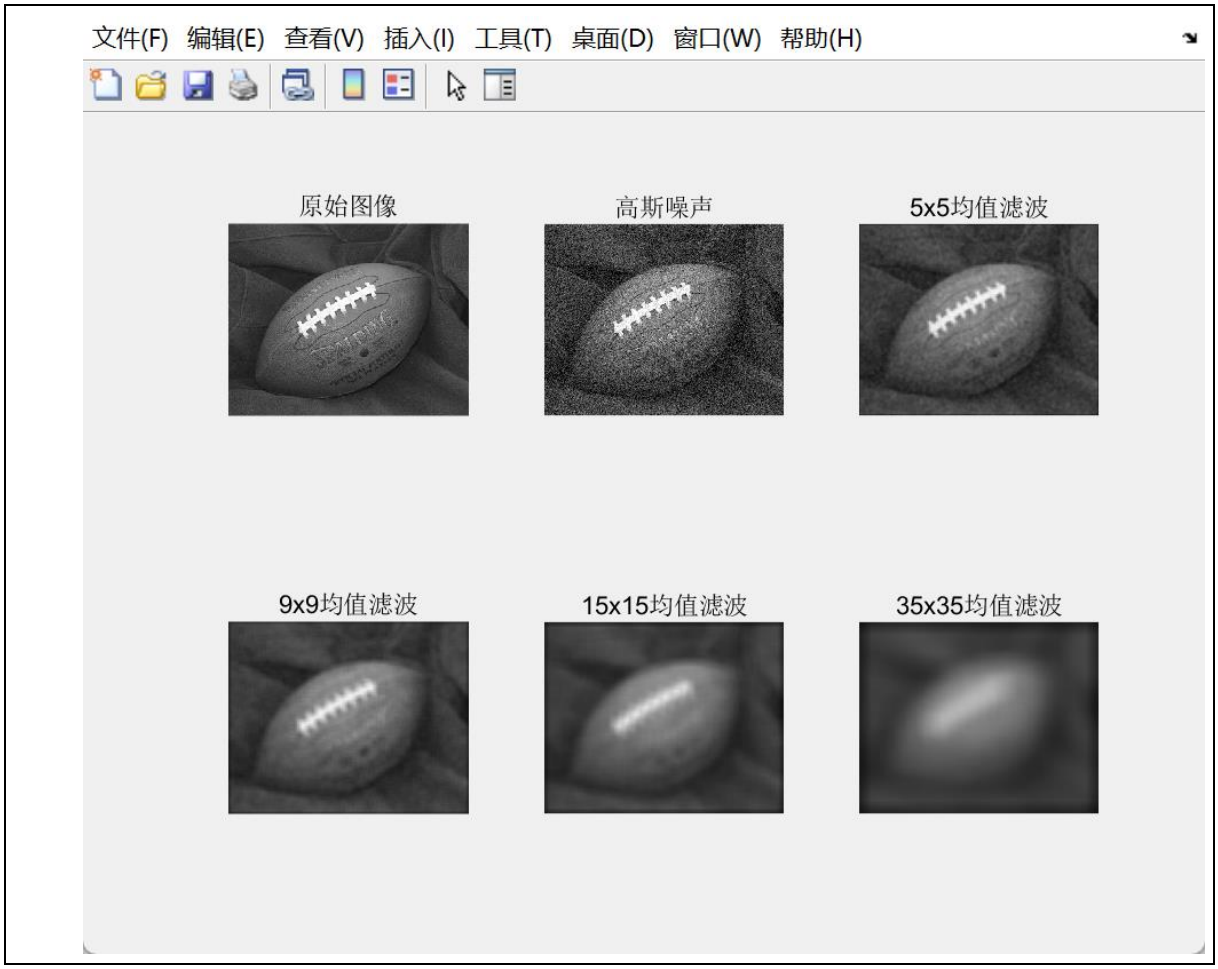
请将实验代码贴在此处：

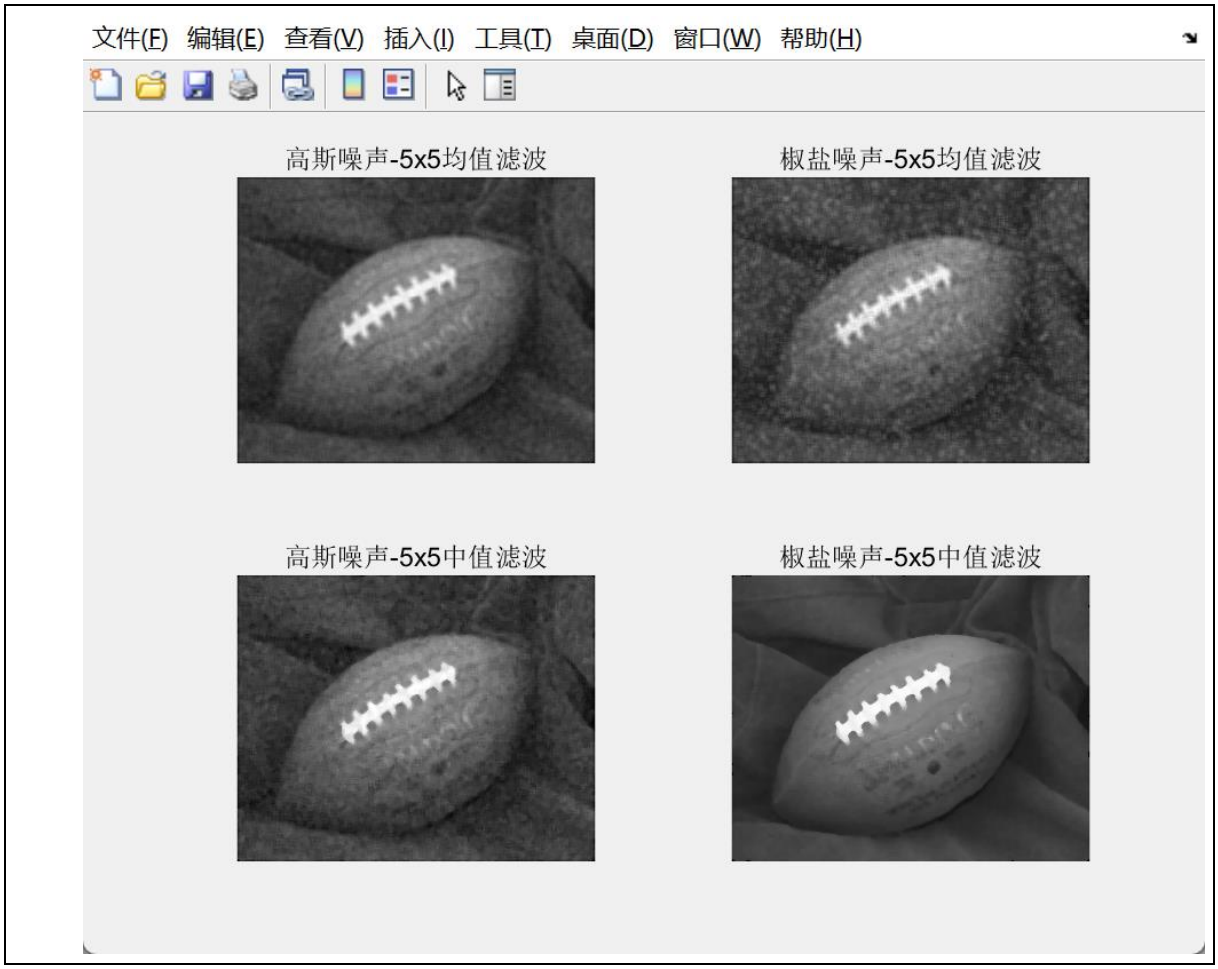
```
%% 3 均值滤波
clc; close all; clear;

% 读取图像并添加噪声
f = imread('football.jpg');
if size(f,3) == 3
    f_gray = rgb2gray(f);
else
    f_gray = f;
end
f_noise_gaussian = imnoise(f_gray, 'gaussian', 0, 0.01);    % 添加高斯噪声
f_noise_saltpepper = imnoise(f_gray, 'salt & pepper', 0.1); % 添加椒盐噪声

% 创建不同尺寸的均值滤波器
kernel_sizes = [5, 9, 15, 35];
for i = 1:length(kernel_sizes)
    h{i} = fspecial('average', kernel_sizes(i));
end

% 对高斯噪声图像进行均值滤波
for i = 1:length(kernel_sizes)
    gaussian_mean{i} = imfilter(f_noise_gaussian, h{i});
end

% 对椒盐噪声图像进行均值滤波
for i = 1:length(kernel_sizes)
    saltpepper_mean{i} = imfilter(f_noise_saltpepper, h{i});
end

% 对高斯噪声和椒盐噪声图像进行中值滤波（支持彩色图像）
gaussian_median = medfilt2(f_noise_gaussian, [5 5]);
```

```
saltpepper_median = medfilt2(f_noise_saltpepper, [5 5]);

% 显示高斯噪声均值滤波效果
figure('Name','高斯噪声均值滤波');
subplot(2,3,1), imshow(f_gray), title('原始图像');
subplot(2,3,2), imshow(f_noise_gaussian), title('高斯噪声');
for i = 1:length(kernel_sizes)
    subplot(2,3,i+2), imshow(gaussian_mean{i}),
title([num2str(kernel_sizes(i)),'x',num2str(kernel_sizes(i)),'均值滤波']);
end

% 显示椒盐噪声均值滤波效果
figure('Name','椒盐噪声均值滤波');
subplot(2,3,1), imshow(f_gray), title('原始图像');
subplot(2,3,2), imshow(f_noise_saltpepper), title('椒盐噪声');
for i = 1:length(kernel_sizes)
    subplot(2,3,i+2), imshow(saltpepper_mean{i}),
title([num2str(kernel_sizes(i)),'x',num2str(kernel_sizes(i)),'均值滤波']);
end

% 对比均值滤波和中值滤波的结果
figure('Name','均值与中值滤波对比');
subplot(2,2,1), imshow(gaussian_mean{1}), title('高斯噪声-5x5 均值滤波');
subplot(2,2,2), imshow(saltpepper_mean{1}), title('椒盐噪声-5x5 均值滤波');
subplot(2,2,3), imshow(gaussian_median), title('高斯噪声-5x5 中值滤波');
subplot(2,2,4), imshow(saltpepper_median), title('椒盐噪声-5x5 中值滤波');
```

请将运行结果贴在此处：


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始图像
椒盐噪声
5x5均值滤波
9x9均值滤波
15x15均值滤波
35x35均值滤波


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始图像
高斯噪声
5x5均值滤波
9x9均值滤波
15x15均值滤波
35x35均值滤波



4 利用拉普拉斯滤波对图像做锐化：

对图像做锐化,方法:做一阶导和二阶导,先滤波再与原图叠加实现锐化

注意滤波器的中心滤波的正负确定锐化叠加时是做减还是做加

或者直接使用合成的拉普拉斯模板,不用做加减(一步到位)

（1）生成拉普拉斯滤波模板 w1 = [0 1 0;1 -4 1;0 1 0]和 w2 = [-1 -1 -1;-1 8 1;-1 -1 -1];

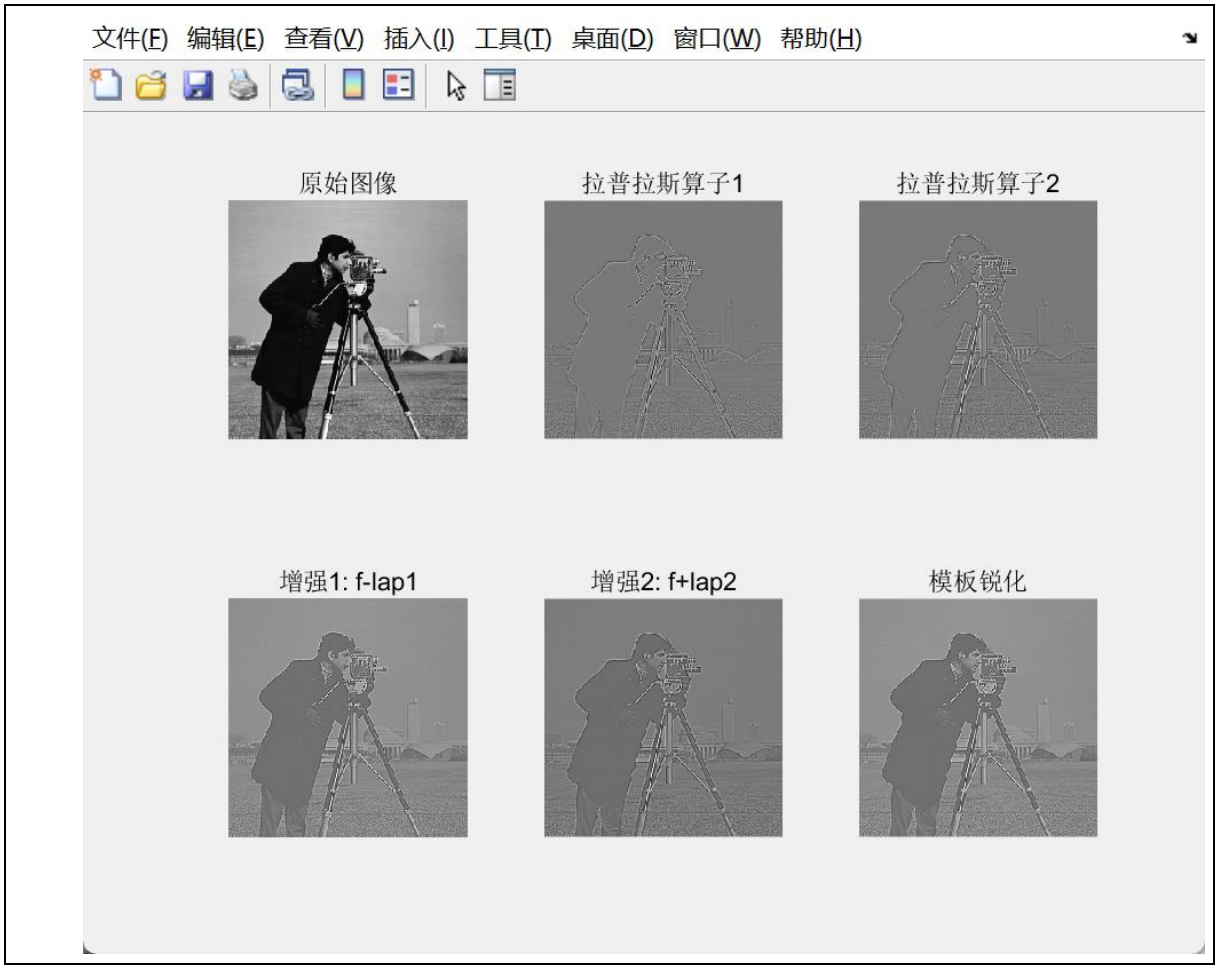
（2）使用（1）中的两个拉普拉斯模板对图像进行滤波，并且要求将滤波后的两幅图像显示在一个图像窗口中，原图和使用拉普拉斯算子锐化后的两幅图像显示在一个图像窗口中。

请将实验代码贴在此处：

```
%% 4 拉普拉斯锐化
% 读取并转换图像
f = imread('cameraman.tif');
f = im2double(f);

% 定义标准拉普拉斯算子
h1 = fspecial('laplacian', 0); % Matlab 标准
h2 = [-1 -1 -1; -1 8 -1; -1 -1 -1]; % 常用中心对称拉普拉斯
```

```
% 应用拉普拉斯算子（指定边界处理方式'replicate'）
lap1 = imfilter(f, h1, 'replicate');
lap2 = imfilter(f, h2, 'replicate');

% 拉普拉斯增强
sharp1 = f - lap1; % 减去拉普拉斯结果
sharp2 = f + lap2; % 加上拉普拉斯结果

% Unsharp Masking（高斯模糊减法锐化）
h3 = fspecial('gaussian', [3 3], 1);
blur = imfilter(f, h3, 'replicate');
sharp3 = imadjust(f + (f - blur)); % 调整对比度

% 定义并应用锐化模板
h4 = [0 -1 0; -1 5 -1; 0 -1 0];
sharp4 = imfilter(f, h4, 'replicate');

% 归一化显示结果，防止超范围
lap1 = mat2gray(lap1);
lap2 = mat2gray(lap2);
sharp1 = mat2gray(sharp1);
sharp2 = mat2gray(sharp2);
sharp3 = mat2gray(sharp3);
sharp4 = mat2gray(sharp4);

% 显示所有结果
figure('Name','拉普拉斯锐化与增强对比');
subplot(2,3,1), imshow(f), title('原始图像');
subplot(2,3,2), imshow(lap1), title('拉普拉斯算子 1');
subplot(2,3,3), imshow(lap2), title('拉普拉斯算子 2');
subplot(2,3,4), imshow(sharp1), title('增强 1: f-lap1');
subplot(2,3,5), imshow(sharp2), title('增强 2: f+lap2');
subplot(2,3,6), imshow(sharp4), title('模板锐化');

% 显示 Unsharp Masking 结果
figure('Name','Unsharp Masking 锐化');
subplot(1,2,1), imshow(f), title('原始图像');
subplot(1,2,2), imshow(sharp3), title('Unsharp Masking');
```

请将运行结果贴在此处：


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原始图像
拉普拉斯算子1
拉普拉斯算子2
增强1: f-lap1
增强2: f+lap2
模板锐化

二、频域图像增强

1 生成理想低通滤波器、巴特沃斯低通滤波器、高斯低通滤波器，选择合适的参数；对灰度图像’exp5_test.tif’做频域滤波处理，将原图和滤波后的图像显示在一个图像窗口中。三种滤波器的传递函数如下：

快速傅里叶变换

生成频率的滤波器使用 dftuv 函数生成频域的坐标系,和图像的大小相同

图像的频谱图的特点:坐标系的原点在四角或者中心,距离原点越远频率值越大(注意比较四角和中心原点之间的联系和区别)

四角的局势由原点到另一个原点先↑后↓

设置截止频率在生成理想低通滤波器

理想低通：

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

将我们的的周期分散在四周,原本是像圆柱的形状(边缘出现 0 和 1 的情况),截止频率越大,底面积越大,后分散到四个角,背靠背

巴特沃斯：

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

n 越大越像理想低通,会出现振铃效应

高斯：

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

Mesh(fftshift(HB))

三个滤波器的大小和图像的大小都是一样的

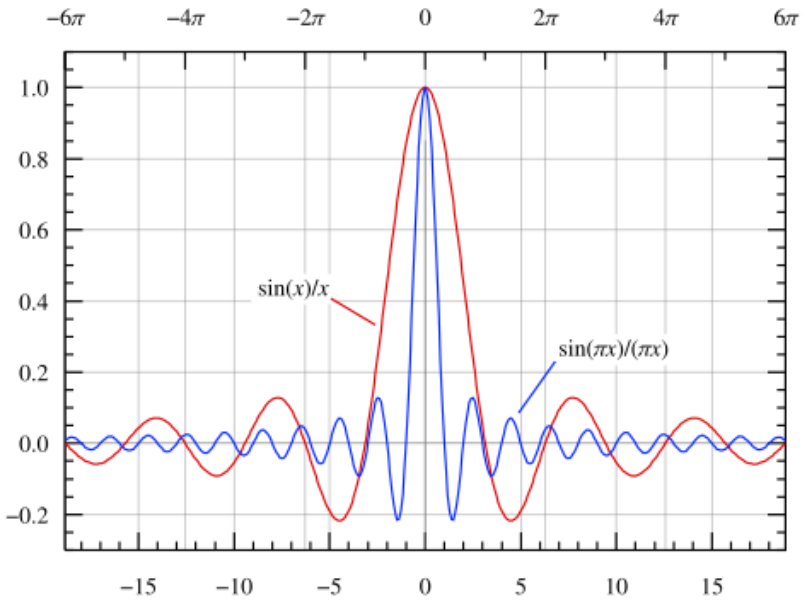
实现了卷积到点积的计算量的减少

低通滤波器滤掉高频信号

要得到更加可视化的图像需要采用傅里叶反变换,但是采用反变换可能会增加虚部的误差

理想低通会产生什么现象,为什么会产生这样的现象

理想低通滤波器会让图像边缘出现一圈圈的波纹，这是因为它在处理图像时，像用刀一样把某些频率“直接切掉”了。这样一切，反而会让图像边缘的明暗变化变得不自然，就像在水面丢石头后产生的波纹一样。



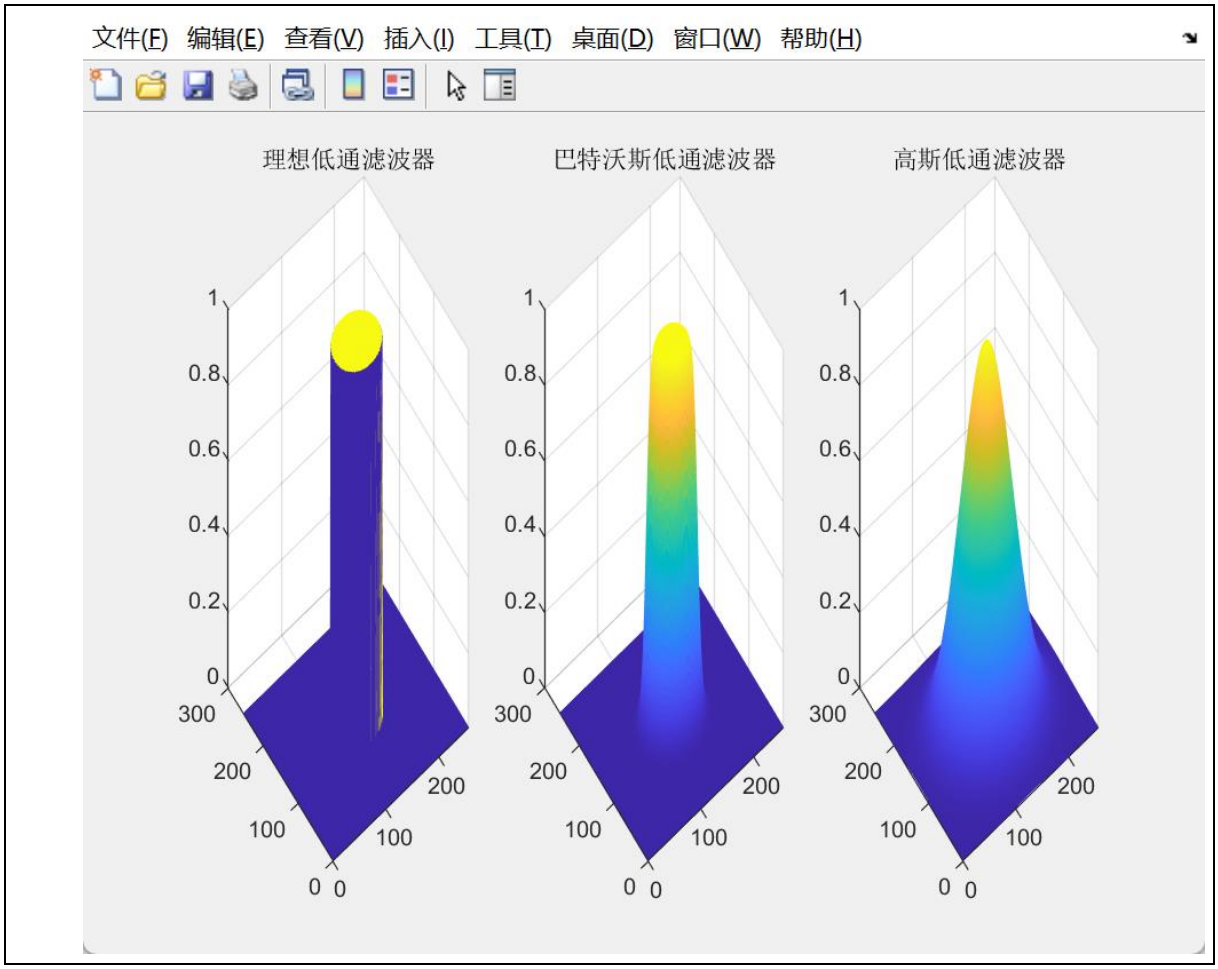
请将实验代码贴在此处：

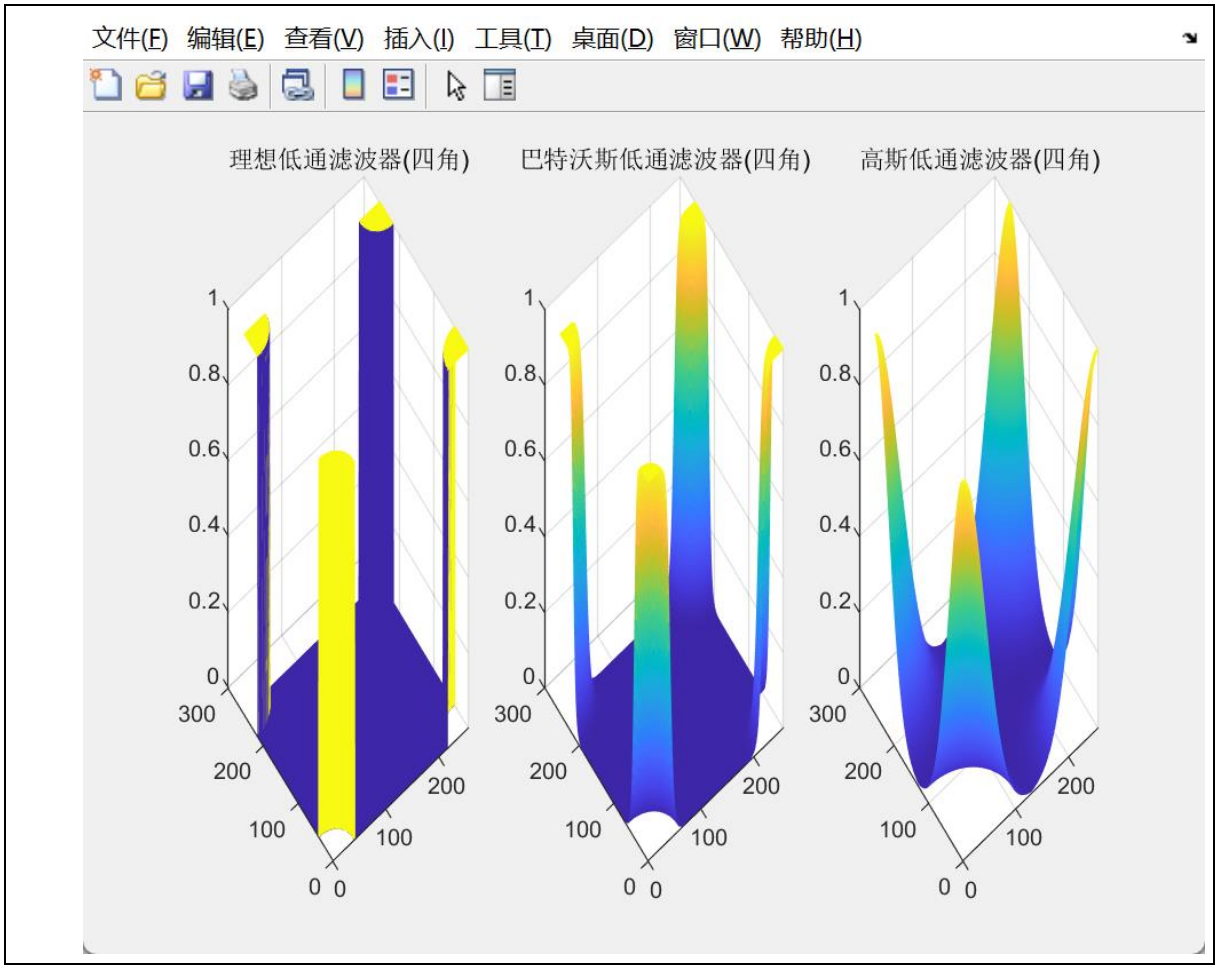
```
%% 频域滤波

% clear;close;clc
f=imread(' exp5_test.tif');
f=im2double(f);
%图像进行傅里叶变换
```

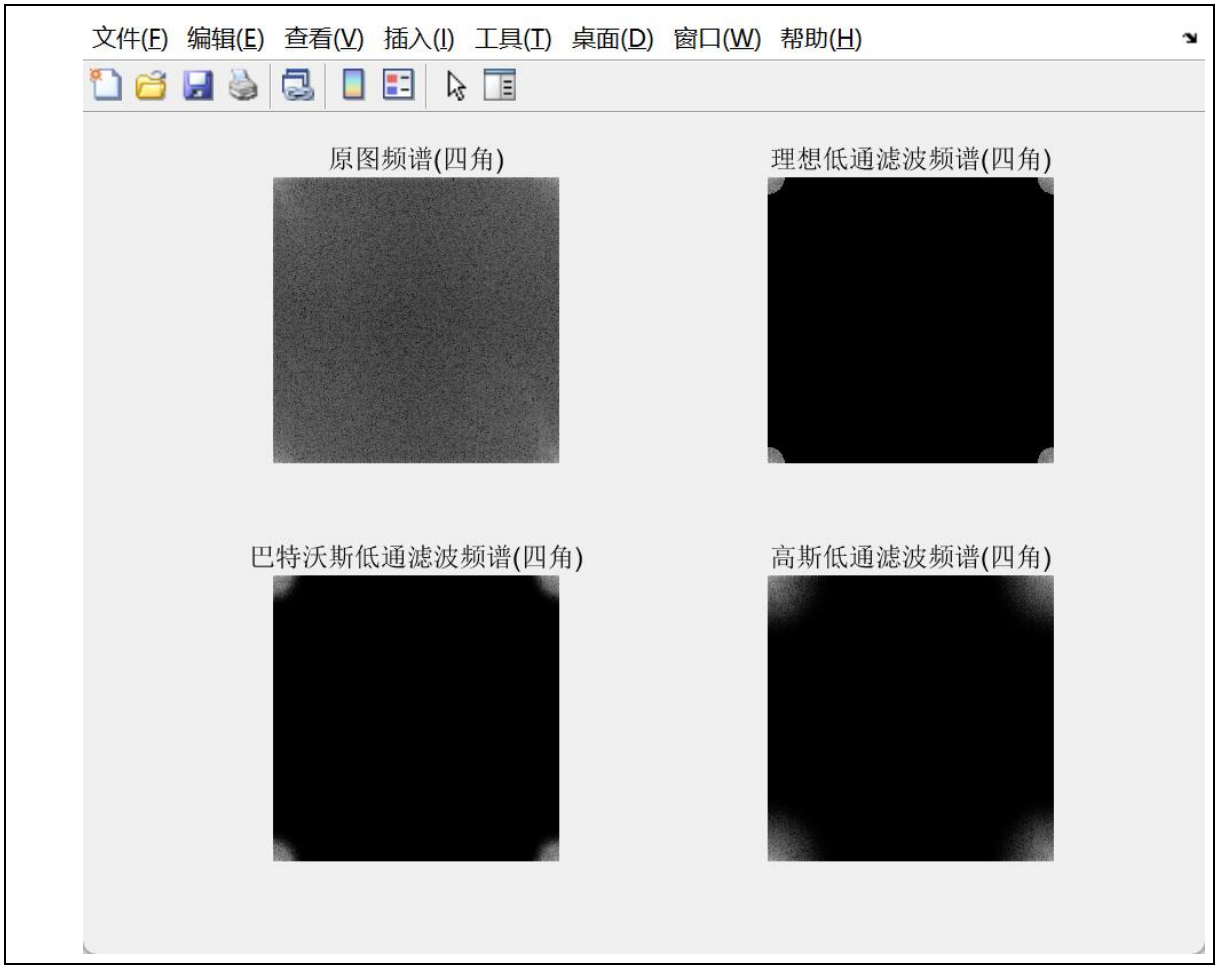
```
F=fft2(f);

%利用 dftuv 函数生成频域坐标
[U,V]=dftuv(size(f,1),size(f,2));
D=hypot(U,V);

%设置截止频率
D0=40;
%生成理想低通滤波器
H1=single(D<=D0);
%巴特沃斯
N=6;
HB=1./(1+(D/D0).^(2*N));

%高斯
HG=exp(-(D.^2)/(2*D0*D0));

% 显示三种滤波器的频率响应（中心视图）
figure(2)
subplot(131),mesh(fftshift(H1)),title('理想低通滤波器(中心)');
subplot(132),mesh(fftshift(HB)),title('巴特沃斯低通滤波器(中心)');
subplot(133),mesh(fftshift(HG)),title('高斯低通滤波器(中心)');

% 显示三种滤波器的频率响应（四角视图）
figure(5)
subplot(131),mesh(H1),title('理想低通滤波器(四角)');
subplot(132),mesh(HB),title('巴特沃斯低通滤波器(四角)');
subplot(133),mesh(HG),title('高斯低通滤波器(四角)');

% 应用三种滤波器并显示结果
G1=F.*H1;
g1=real(ifft2(G1));

G2=F.*HB;
g2=real(ifft2(G2));

G3=F.*HG;
g3=real(ifft2(G3));

% 显示滤波后的频谱和结果
figure(3)
subplot(221), imshow(f), title('原图');
subplot(222), imshow(g1), title('理想低通滤波结果');
subplot(223), imshow(g2), title('巴特沃斯低通滤波结果');
subplot(224), imshow(g3), title('高斯低通滤波结果');
```

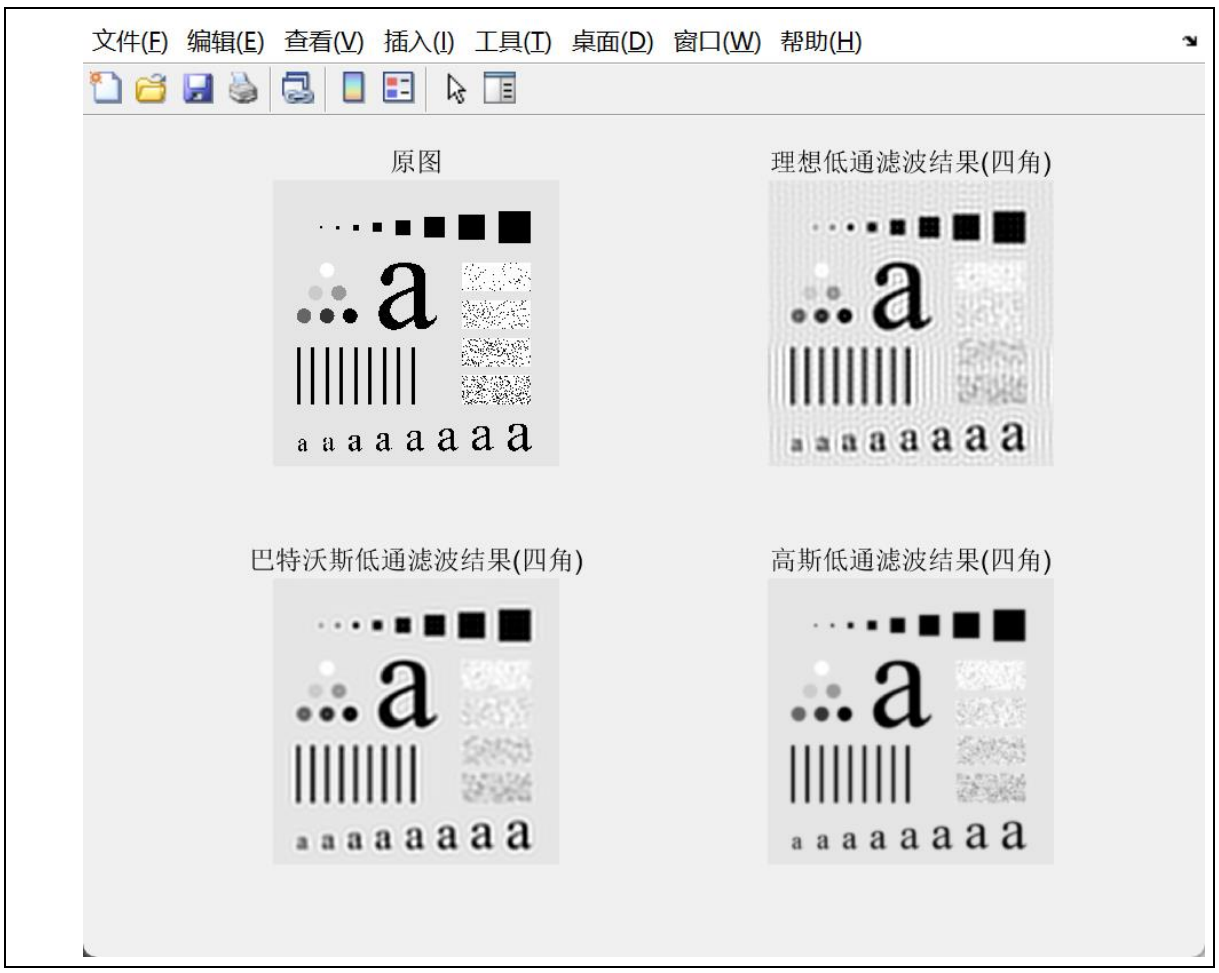
```
% 显示滤波后的频谱
figure(4)
subplot(221), imshow(log(1+abs(fftshift(F))), []), title('原图频谱');
subplot(222), imshow(log(1+abs(fftshift(G1))), []), title('理想低通滤波频谱');
subplot(223), imshow(log(1+abs(fftshift(G2))), []), title('巴特沃斯低通滤波频谱');
subplot(224), imshow(log(1+abs(fftshift(G3))), []), title('高斯低通滤波频谱');

% 显示四角视图的滤波结果
figure(6)
subplot(221), imshow(f), title('原图');
subplot(222), imshow(real(ifft2(F.*H1))), title('理想低通滤波结果(四角)');
subplot(223), imshow(real(ifft2(F.*HB))), title('巴特沃斯低通滤波结果(四角)');
subplot(224), imshow(real(ifft2(F.*HG))), title('高斯低通滤波结果(四角)');

% 显示四角视图的滤波频谱
figure(7)
subplot(221), imshow(log(1+abs(F)), []), title('原图频谱(四角)');
subplot(222), imshow(log(1+abs(G1)), []), title('理想低通滤波频谱(四角)');
subplot(223), imshow(log(1+abs(G2)), []), title('巴特沃斯低通滤波频谱(四角)');
subplot(224), imshow(log(1+abs(G3)), []), title('高斯低通滤波频谱(四角)');
```

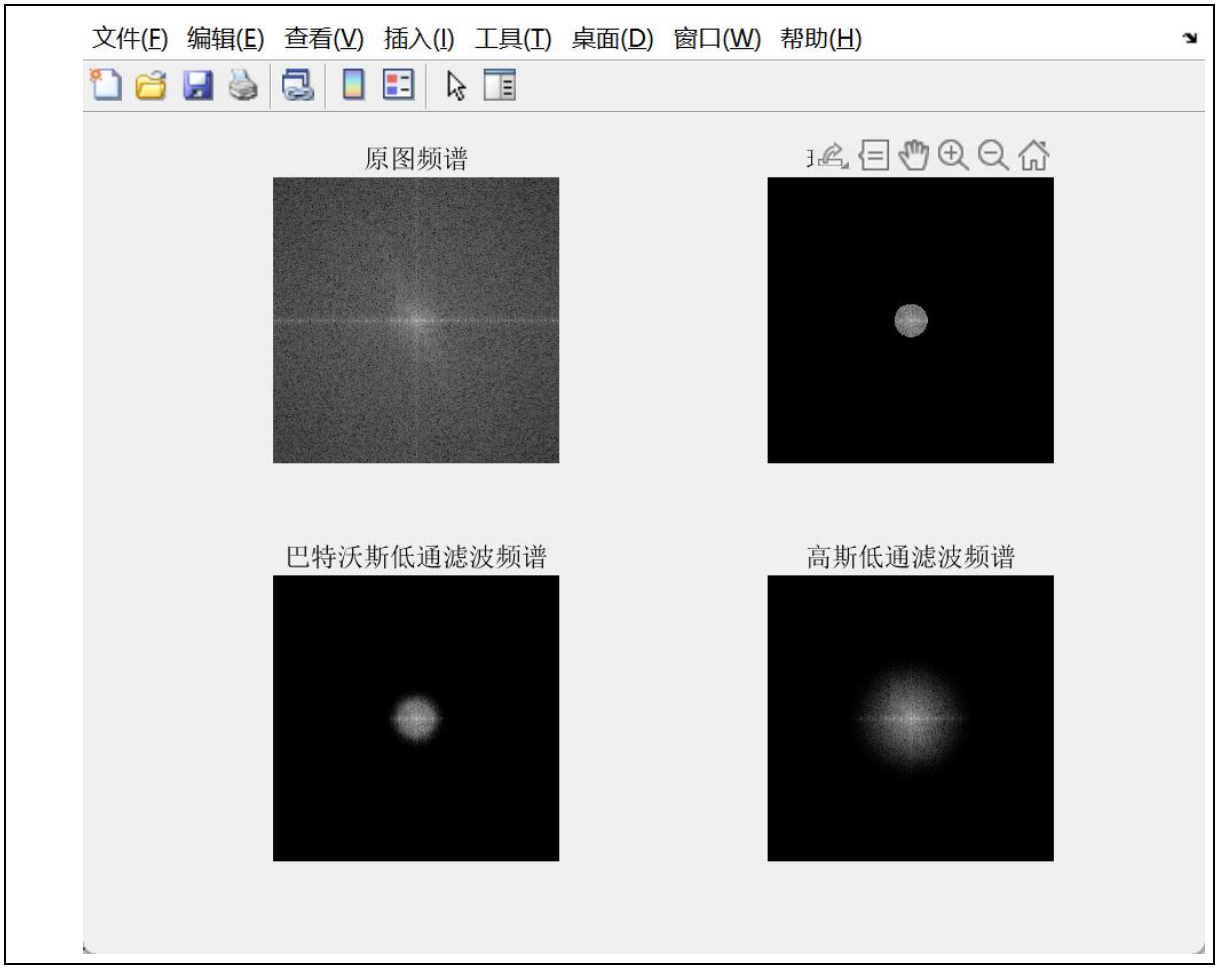
请将运行结果贴在此处：


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
理想低通滤波器
巴特沃斯低通滤波器
高斯低通滤波器
1
0.8
0.6
0.4
0.2
0
300
200
100
0
0
100
200


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
理想低通滤波器(四角)
巴特沃斯低通滤波器(四角)
高斯低通滤波器(四角)
1
0.8
0.6
0.4
0.2
0
300
200
100
0
0
100
200


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图频谱(四角)
理想低通滤波频谱(四角)
巴特沃斯低通滤波频谱(四角)
高斯低通滤波频谱(四角)


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图
理想低通滤波结果(四角)
巴特沃斯低通滤波结果(四角)
高斯低通滤波结果(四角)


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图频谱
巴特沃斯低通滤波频谱
高斯低通滤波频谱

2 生成理想高通滤波器、巴特沃斯高通滤波器、高斯高通滤波器，选择合适的参数；对灰度图像’exp5_test.tif’做频域滤波处理，将原图和滤波后的图像显示在一个图像窗口中。

用 1-低通滤波就是对应的高通滤波

方法 1:类似低通滤波的方法

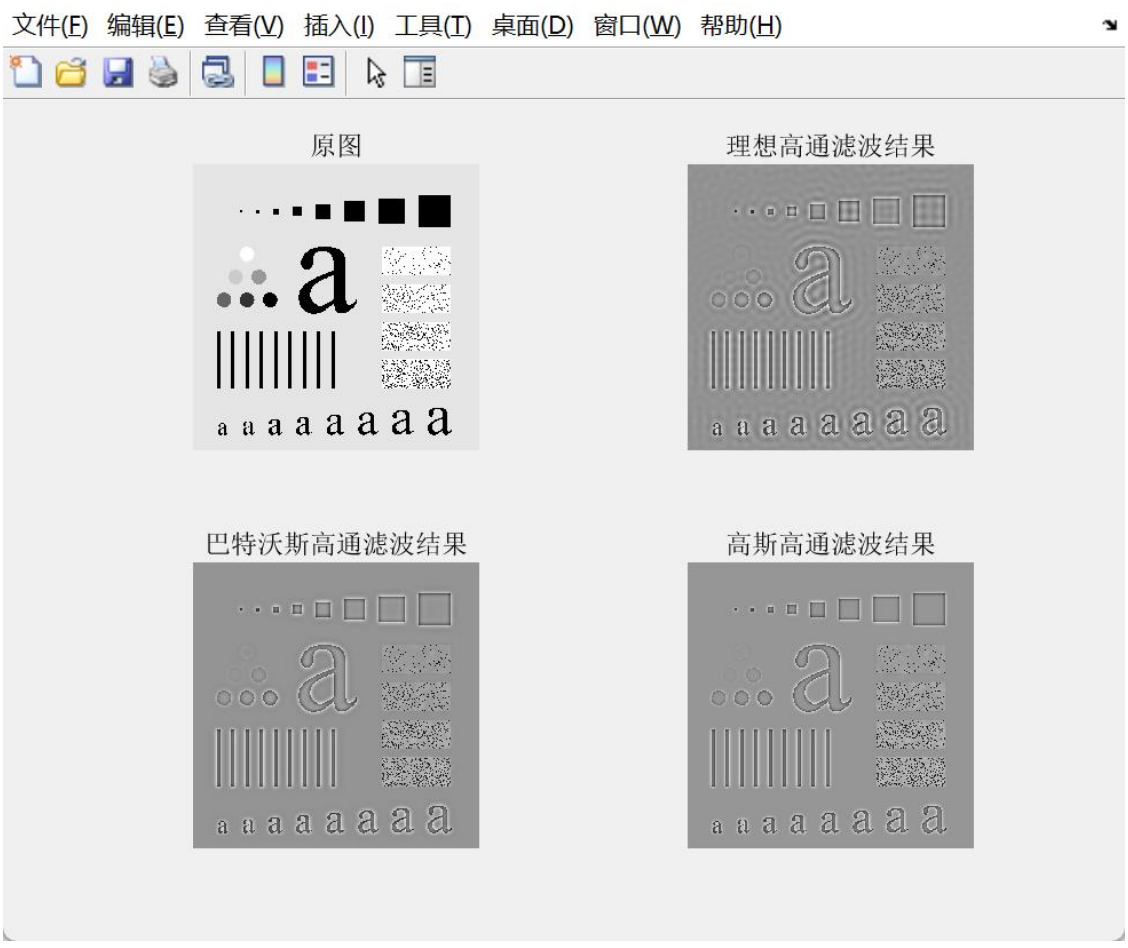
方法 2,采用老师写的函数的方法

请将实验代码贴在此处：

```
%% 高通滤波器
f = imread('exp5_test.tif');

% 生成理想高通滤波器
D0 = 30;  % 截止频率
[M, N] = size(f);
[U, V] = dftuv(M, N);
D = hypot(U, V);
H1 = double(D > D0);

% 生成巴特沃斯高通滤波器
n = 2;  % 阶数
HB = 1./(1 + (D0./D).^(2*n));
```

```
% 生成高斯高通滤波器
HG = 1 - exp(-(D.^2)./(2*D0^2));

% 应用滤波器
F = fft2(f);
G1 = F.*H1;
G2 = F.*HB;
G3 = F.*HG;

g1 = real(ifft2(G1));
g2 = real(ifft2(G2));
g3 = real(ifft2(G3));

% 显示结果
figure
subplot(221), imshow(f), title('原图');
subplot(222), imshow(g1, []), title('理想高通滤波结果');
subplot(223), imshow(g2, []), title('巴特沃斯高通滤波结果');
subplot(224), imshow(g3, []), title('高斯高通滤波结果');
```

请将运行结果贴在此处：



3 用带阻滤波器消除周期噪声。

带阻滤波器,周期噪声,和我们的图像有关联性,重复出现的规律,类似纹理,用我们之前那的知识可以用均值和中值滤波去噪

带阻滤波器->滤掉特定的一个圈，宽度 w,初始截止频率 D_0 位置

截止频率 $D_0$50,宽度 w30 是较优的

采用循环遍历的方法找较优的 w 和 D_0

1）生成一幅具有周期噪声的图像 f，给出其空域图像及频域图像；

2）设计高斯带阻滤波器 H 并画出其频域响应，表达式如下：

$$H(u,v)=1\text{-}e^{-\left[\frac{D(u,v)^2-D_0^2}{WD(u,v)}\right]^2}$$

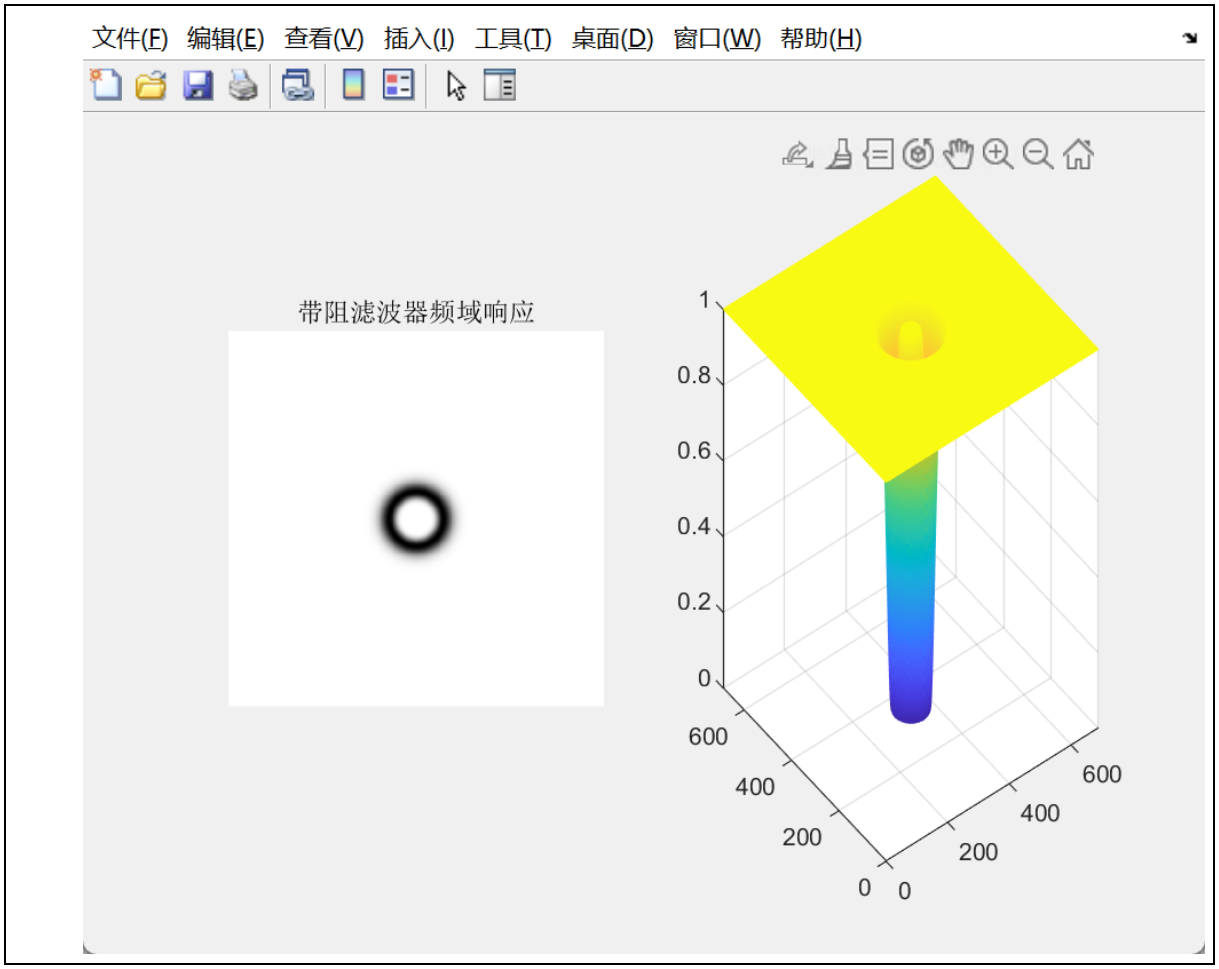
3）选择合适的参数，进行频域滤波，比较滤波后的结果。

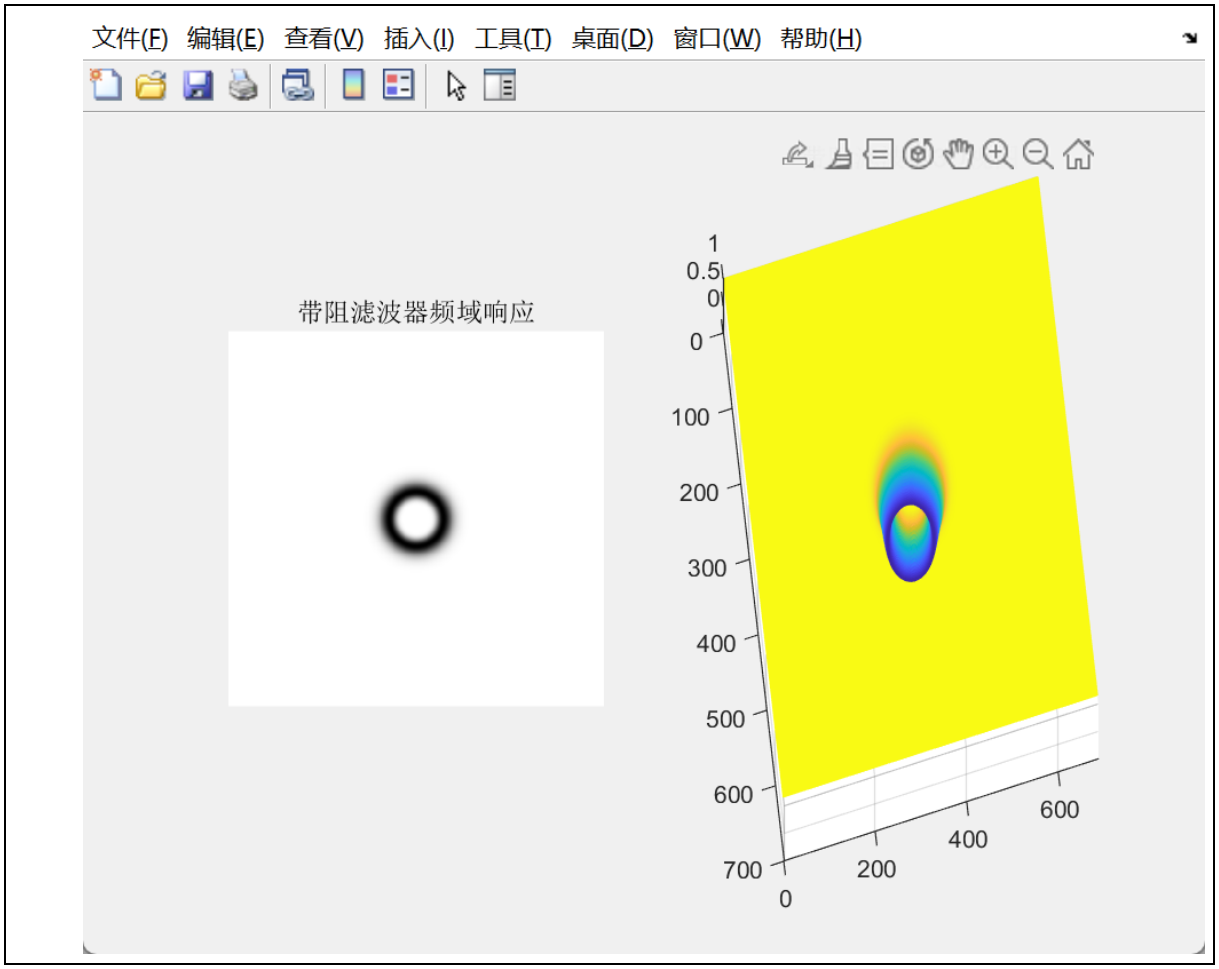
请将实验代码贴在此处：

```
%% 带阻滤波器
clc; clear; close all;

% 1. 生成具有周期噪声的图像
f = imread('exp5_test.tif');
[m, n] = size(f);
fn = f;
for i = 1:m
    for j = 1:n
        fn(i,j) = fn(i,j) + 20*sin(20*i) + 20*sin(20*j);
    end
end

% 显示原图和添加噪声后的图像
figure(1)
subplot(121), imshow(f), title('原图');
subplot(122), imshow(fn), title('添加周期噪声后的图像');

% 2. 显示频域图像
F = fft2(fn);
FN = fftshift(log(1 + abs(F)));
figure(2)
imshow(FN, []), title('频域图像');

% 3. 设计高斯带阻滤波器
[U, V] = dftuv(size(FN,1), size(FN,2));
D = hypot(U, V);

% 设置最优参数
D0 = 50;  % 截止频率
```

```
W = 30;   % 带宽

% 生成带阻滤波器
H = 1 - exp(-((D.^2 - D0^2)./(D.*W + eps)).^2);

% 显示滤波器频域响应
figure(3)
subplot(121), imshow(fftshift(H), []), title('带阻滤波器频域响应');
subplot(122), mesh(fftshift(H)), title('带阻滤波器 3D 视图');

% 4. 应用滤波器
H = ifftshift(H);
g = real(ifft2(H.*F));

% 显示滤波结果
figure(4)
subplot(131), imshow(f), title('原图');
subplot(132), imshow(fn), title('含噪声图像');
subplot(133), imshow(g, []), title('滤波结果');

% 5. 对比均值滤波结果
avg = fspecial('average', 3);
res = imfilter(fn, avg);
figure(5)
subplot(121), imshow(fn), title('含噪声图像');
subplot(122), imshow(res), title('均值滤波结果');
```

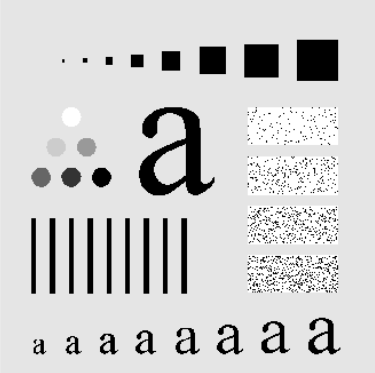
请将运行结果贴在此处：


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
带阻滤波器频域响应
1
0.8
0.6
0.4
0.2
0
600
400
200
0
0
200
400
600

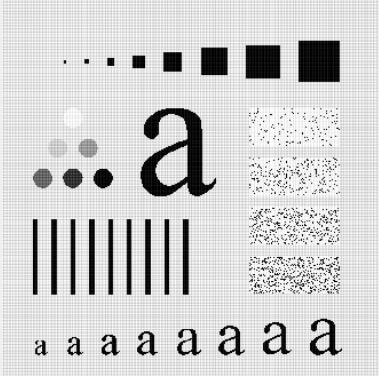

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
带阻滤波器频域响应
1
0.5
0
0
100
200
300
400
500
600
700
0
200
400
600

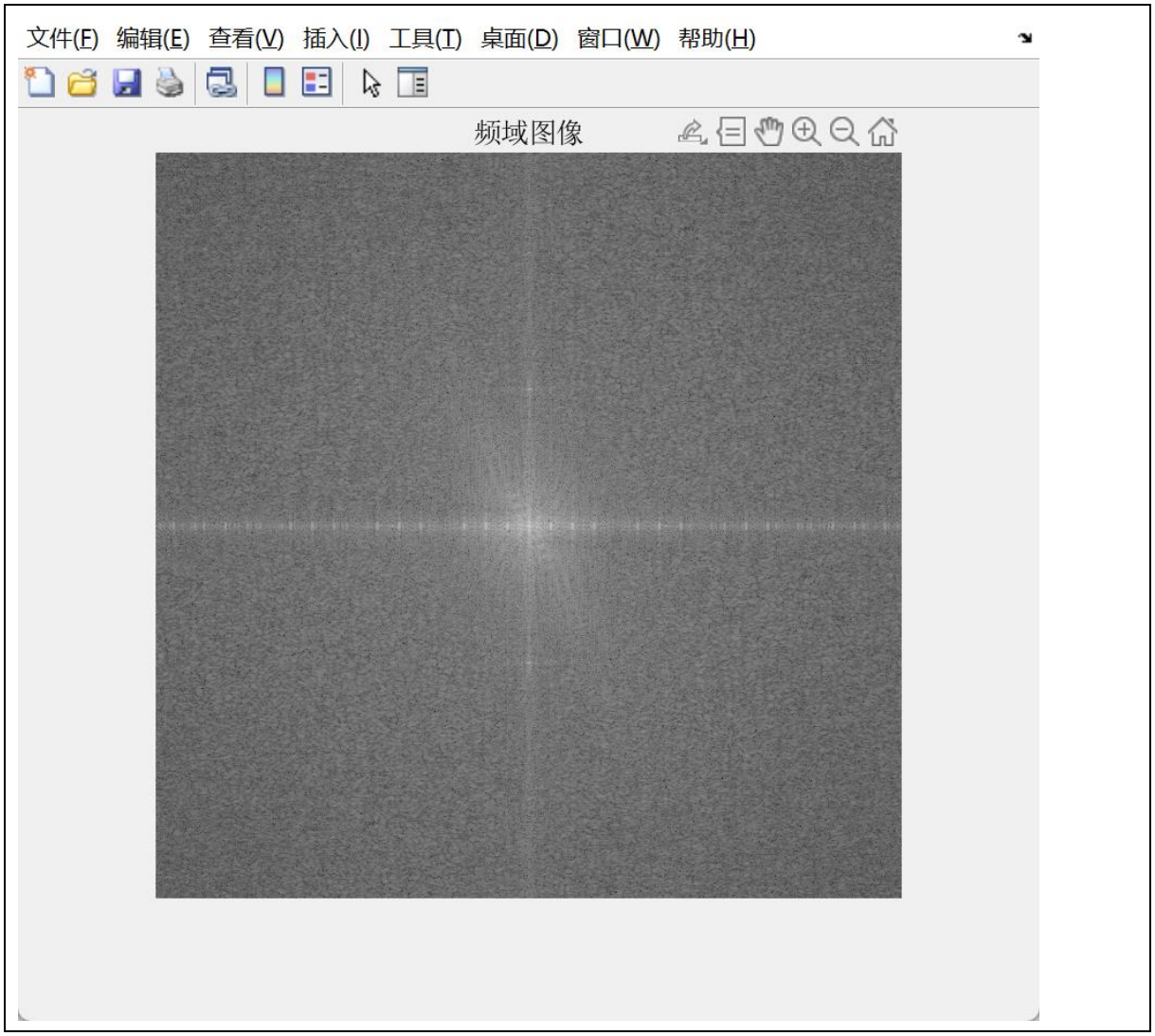
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)

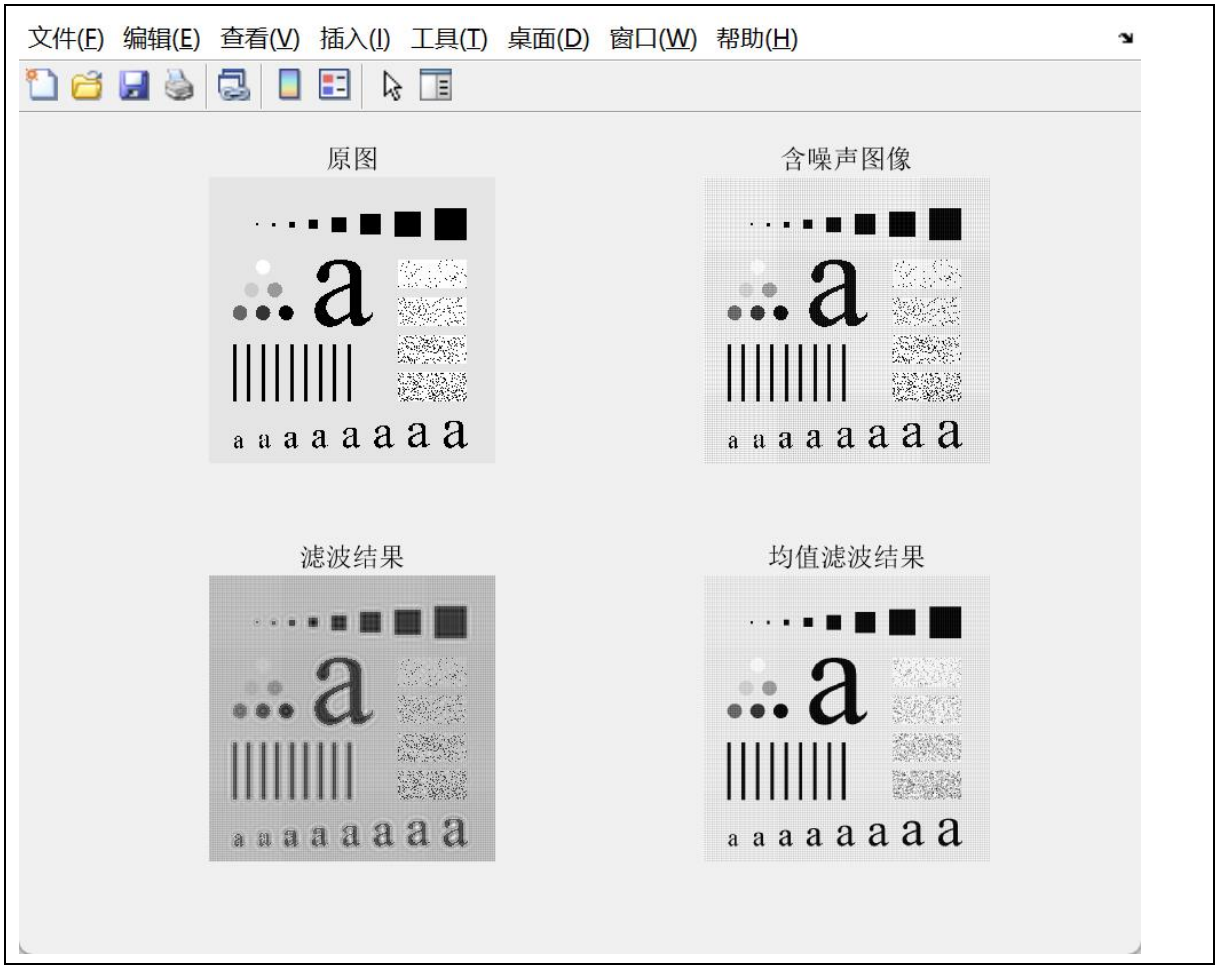
原图



添加周期噪声后的图像




文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
频域图像



注意：本实验报告要求直接将本 word 文档上传至课程平台