

高频噪声图像的去噪仿真

一、 研究背景

噪声广泛存在于数字图像的获取，编码，传输和处理阶段。图像噪声是图像信息或者像素亮度的随机变化。它是由一些因素造成的图像信号的退化。图像通常会包含很多噪声，这样的图像一般表现为：图像越亮的区域，噪声越多。可以有下面公式定义：

$$A(x, y) = N(x, y) + F(x, y)$$

这表示的是加性噪声的模型。其中 $A(x, y)$ 是含噪声图像， $N(x, y)$ 是纯噪声图像， $F(x, y)$ 是原始纯净图像。顾名思义，含噪声图像是纯噪声图像加上原始纯净图像。

噪声的来源有很多，包括：

- 1、电子元器件传输发热。
- 2、成像系统的调制，缺陷。
- 3、环境的其他影响和干扰。

在图像预处理中，图像去噪是一个十分重要的步骤，对于后面图像的后处理会有很大的影响。需要选择适当的滤波技术来增强图像。

本实践将制作一个高频噪声并加入一张图像，对加噪后的图像进行快速傅里叶变换，画出频谱分布图，设计合理的低通滤波器（如巴特沃思）滤除高频分量，运用逆傅里叶变换还原图像，对比原始图像、加噪图像和还原图像的误差。

二、 研究理论

1、基本概念

图像：主要有三种数字图像，分别是二值黑白图像、灰度图像、彩色图像。

二值黑白图像，一般地，白色代表最亮的像素，编码为 255，黑色代表最暗的像素，编码为 0。

灰度图像，灰度从全黑到全白变化，所以，一般用 0-255 分别编码不同的灰度级。在 matlab 中，读取图像文件（jpg）会保存为 uint8 格式的文件，对应 256（ 2^8 ）级灰度编码，编写代码时注意将 double 转为 uint8 格式。

彩色图像，一般地，是三通道的图像，每一个通道像素的编码类似于灰度图像的编码，最后对三个通道施加不同的权重，融合成彩色图像。

本实验主要研究灰度图像。彩色图像可以由三通道的灰度图像融合而成。

2、高斯噪声（gauss）

高斯噪声是指它的概率密度函数服从高斯分布（即正态分布）的一类噪声，是数字图像的一个常见噪声。高斯噪声服从高斯分布，有均值和方差两个参数。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

在数字图像中的高斯噪声的主要来源出现在采集期间。由于不良照明或高温引起的传感器噪声。

3、巴特沃斯滤波器

滤波是将信号中特定波段频率滤除，得到想要的频率信号的操作，是抑制和防止干扰的一项重要措施。在本实验中，用到一个高通滤波器和一个低通滤波器。其中高通滤波器用于产生高频噪声，低通滤波器用于对加噪图像进行滤噪。实验时都采用巴特沃斯(Butterworth)滤波器实现。

巴特沃斯滤波器 (Butterworth filter)：巴特沃斯滤波器 (Butterworth filter) 是电子滤波器的一种，它也被称作最大平坦滤波器。巴特沃斯滤波器的特点是通频带内的频率响应曲线最大限度平坦，没有纹波，而在阻频带则逐渐下降为零。

类同于切比雪夫滤波器，它有高通，低通，带通，带阻等多种滤波器。它在通频带内外都有平稳的幅频特性，但有较长的过渡带，在过渡带上很容易造成失真，在调用 MATLAB 里的巴特沃斯滤波器做仿真时，信号总会在第一个周期略微有些失真。但往后的幅频特性就非常的好。

巴特沃斯低通滤波器可用如下振幅的平方对频率的公式表示：

$$|H(w)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

其中，n 是滤波器的阶数， ω_c = 截止频率

巴特沃斯低通频率响应：

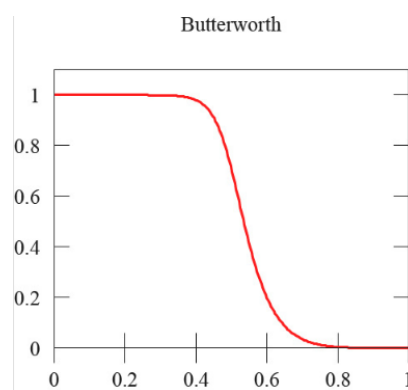
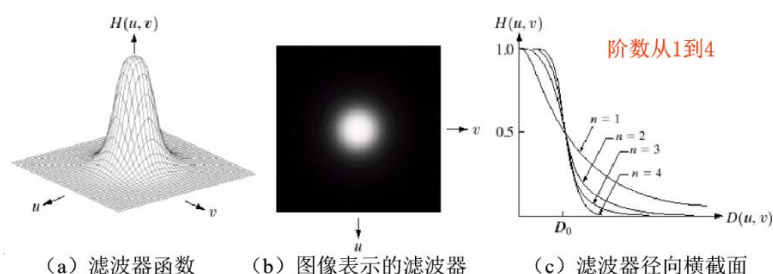


图 1 巴特沃斯低通频响图

理论中的巴特沃斯滤波器是针对一维而言，对于二维，可以做以下改变：

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

其中 D_0 为截止频率与原点的距离， $D(u,v)$ 是点 (u,v) 与原点的距离。滤波特性图如下：



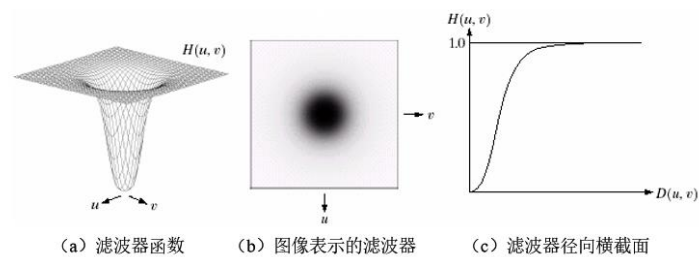
图二 低通巴特沃斯特性

低通滤波在图像中可以用于模糊、平滑等。例如用于字符识别，通过模糊图像，桥接断裂字符的裂缝；用于美颜，减少人脸的皮肤细纹和小斑点；处理卫星和航空图像，尽可能模糊细节，从而保留大的可识别特征。

对于高通滤波器，公式变为：

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

特性图如下：



图三 高通巴特沃斯滤波器

三、 程序实现

1、 图像部分

这部分代码最需要注意的是数据的格式问题。在 matlab 中图像处理采用的是 uint8 格式（对应 256 级灰阶），而在处理过程中可能会出现 double 型，需要注意转换。

图像导入：I = imread('test.jpg')。将 test.jpg 图像中的数据导入 I 中。这时导入的是三维数组数据，对应 RGB 彩色图像。

图像显示：imshow(I)。

图像写入：imwrite(I, 'test1.jpg')。

数据转换：F=rgb2gray(I)。将三维 RGB 数据变为二维灰阶数据。

时频转换：F_fft = fft2(F)。对二维数据进行傅里叶变换。

频域显示：mesh(abs(fftshift(F_fft)))。mesh 用于显示三维图像，fftshift 用于坐标系转换、

2、 高斯噪声

实验中编写一个函数 gauss 用于生成高斯噪声：

```
function output=gauss(m,n)
%I=uint8(256*rand(m,n));%法1、直接用rand随机。
I=zeros(m,n);%法2、先生成一个0矩阵，再用imnoise给图片加高斯噪声
% 添加高斯白噪声给图像I, 均值为m, 方差为v. 默认m = 0, v = 0.01
J=imnoise(I, 'gaussian', 0, 0.5);
output=J;
end
```

图四 生成高斯代码

实验尝试了两种方式产生高斯噪声：

一种是用 rand 函数直接生成随机数，另一种先生成零矩阵再用 imnoise 函数对该矩阵加上高斯噪声。

3、 巴特沃斯滤波器

低通部分：

```

function output=butter_low(I)
Y=fft2(im2double(I));%fft, 注意进行uint8转double型
Y=fftshift(Y);%将坐标系移动到中心
[M,N]=size(Y);%获得图像的高度和宽度
h=zeros(M,N);%滤波器函数
%图像中心点
M0=M/2;
N0=N/2;
d0=200;%40、100
%巴特沃斯滤波器的阶数
n_0=2;
for x=1:M
    for y=1:N
        distance=sqrt((x-M0)^2+(y-N0)^2);
        h(x,y)=1/(1+(distance/d0)^(2*n_0));%滤波器函数
    end
end
%滤波后结果
res=h.*Y;%频域相乘
output=real(ifft2(ifftshift(res)));%输出
end

```

图五 低通巴特沃斯代码

高通部分:

```

function output=butter_high(I)
Y=fft2(im2double(I));%傅里叶变换
Y=fftshift(Y);%坐标系转换
[M,N]=size(Y);
h=zeros(M,N);%滤波器函数
%图像中心点
M0=M/2;
N0=N/2;
d0=300;
%巴特沃斯滤波器的阶数
n_0=2;
for x=1:M
    for y=1:N
        distance=sqrt((x-M0)^2+(y-N0)^2);
        h(x,y)=1/(1+(d0/distance)^(2*n_0));%滤波器公式
    end
end
%滤波后结果
res=h.*Y;%频域相乘
output=real(ifft2(ifftshift(res)));%转换回时域、输出
end

```

图六 高通巴特沃斯代码

高通与低通的差别只在截至频率 d0 和滤波器公式部分。

4、主函数:

```

%读取图像
I = imread('lena.bmp');%读取图像
[m, n, y] = size(I);%图像参数
%图像预处理
subplot(6,2,1);
figure
imshow(I);
title(' 原图');
F=rgb2gray(I);%rgb转灰度图
F=I;
F=im2double(F);%灰度uint8转double

subplot(6,2,3);
figure
imshow(F);
title(' 灰度图');

F_fft = fft2(F);%求频域
subplot(6,2,4);
figure
mesh(abs(fftshift(F_fft)));%频域图,将低频移到中间
title(' 频域图');
%高频高斯噪声
N=gauss(m,n);
subplot(6,2,5);
figure
imshow(N);
title(' 高斯时域');

N_fft=fft2(N);%噪声频域
subplot(6,2,6);
figure
mesh(abs(fftshift(N_fft)));
title(' 高斯频域');

N_high=butter_high(N);%高斯噪声过高频滤波器
subplot(6,2,7);
figure
imshow(N_high);
title(' 高频高斯时域');

N_high_fft=fft2(N_high);
subplot(6,2,8);
figure
mesh(abs(fftshift(N_high_fft)));
title(' 高频高斯频域');
%噪声与图像叠加
Y=N_high+F;%加性噪声
subplot(6,2,9);
figure
imshow(Y);
title(' 加噪图');

Y_fft=fft2(Y);%加噪图的频谱
subplot(6,2,10);
figure
mesh(abs(fftshift(Y_fft)));
title(' 加噪频域');
%图像低通处理
Y_low=butter_low(Y);%过低通滤波器
subplot(6,2,11);
figure
imshow(Y_low);
title(' 滤波后时域图');

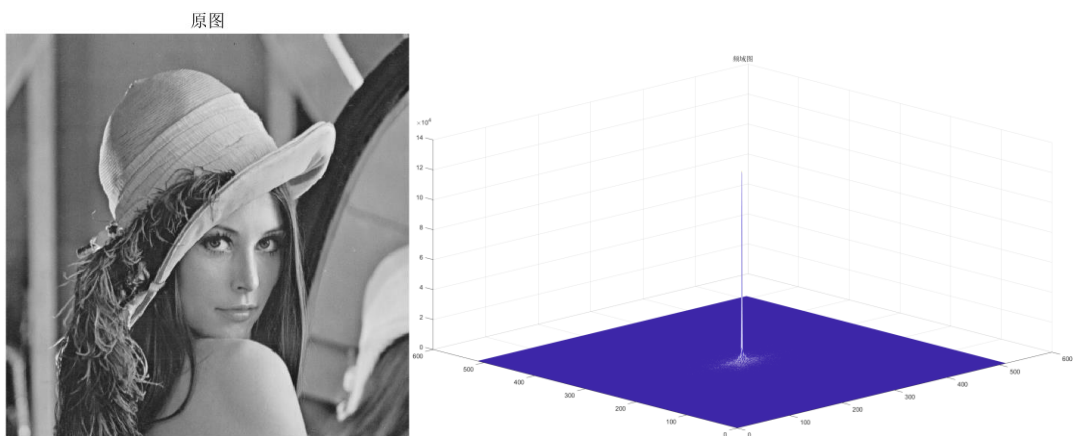
Y_low_fft=fft2(Y_low);%通过低通滤波器后的频谱
subplot(6,2,12);
figure
mesh(abs(fftshift(Y_low_fft)));
title(' 滤波后频域');

```

图七 主函数代码

四、 仿真图形：

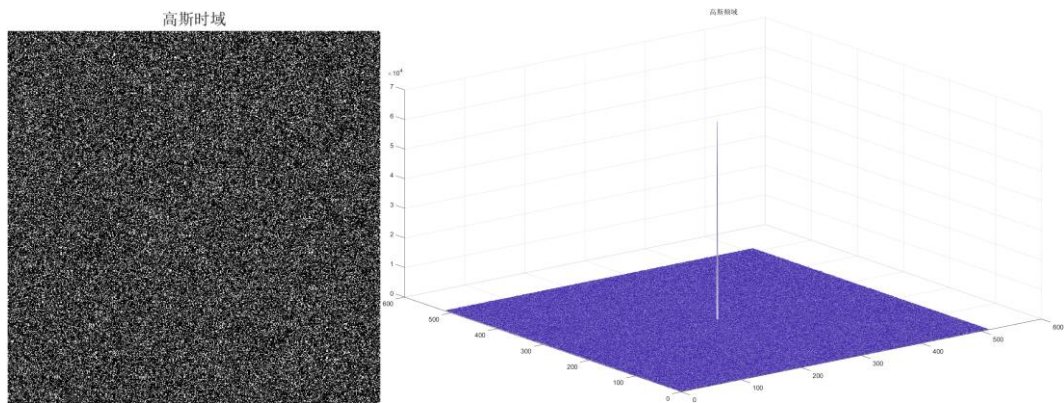
首先输入原图，实验采用图像处理中经典的 luna 图像进行实验。这是其图片以及频谱图片：



图八 原图的时域与频域

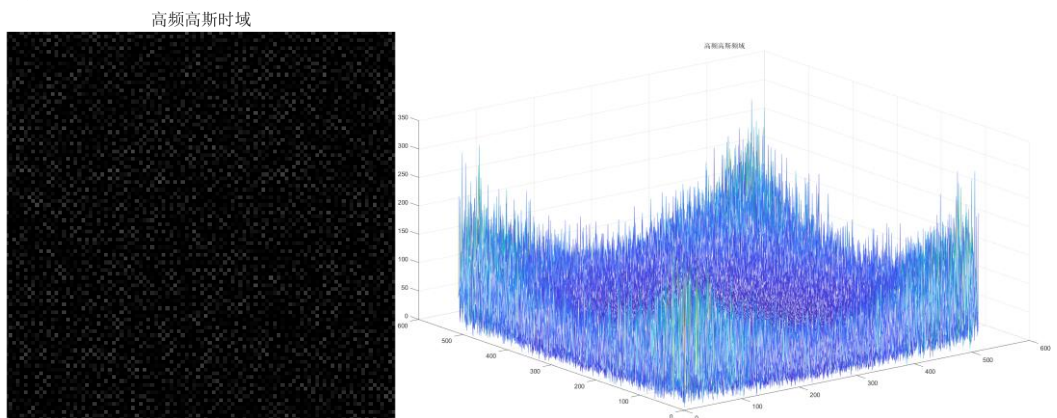
由频谱图可以看出，图片主要集中在低频。所以本实验对图片加上高频噪声再过低通滤波器后仍然能保留大部分的信息。

接下来看高斯噪声部分：



图九 高斯噪声的时域与频域图

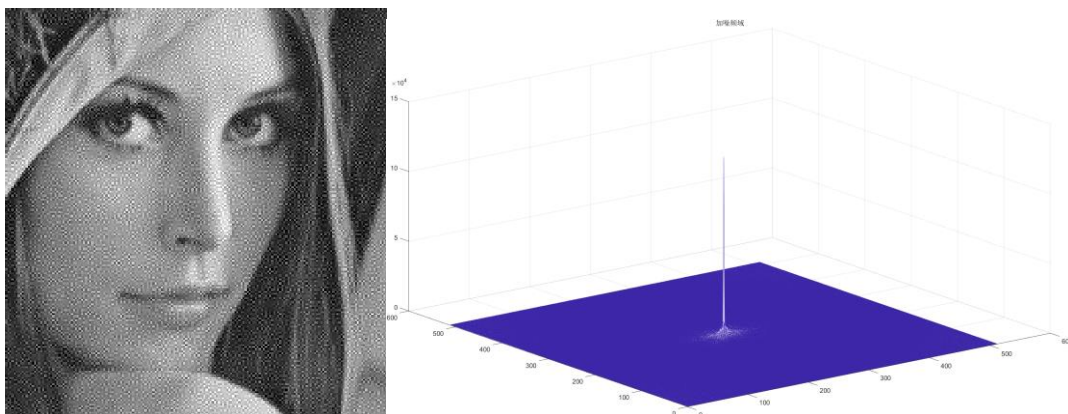
对比高斯的频谱和图像的频谱可以发现，高斯噪声的频谱更为集中，高频分布更为均匀。而图像的频谱反而不如高斯的集中。



图十 高频高斯时域和频域

对比时域图可以得出：高斯噪声是一种普遍的、均匀分布的噪声；而高频高斯噪声在图像中表现为细小的、局部化的随机亮度变化，可能会导致图像出现颗粒状的噪点。对应到频域可以看出，在高斯噪声中低频非常突出而高频部分并不明显；高通处理后低频部分大大衰减而高频相对突出。

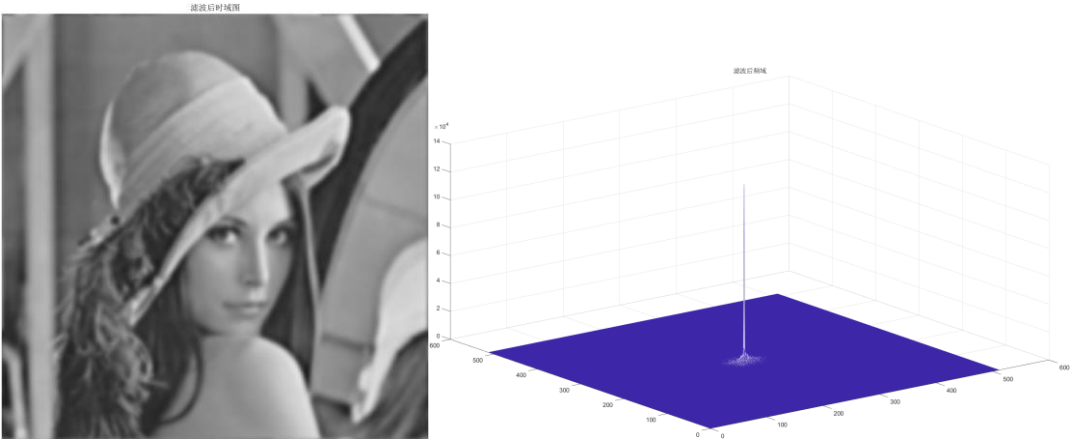
高斯噪声是加性噪声，将噪声与原图数据相加得到加噪图：



图十一 加噪后的时域与频域

可以看到加噪后的图片出现一些噪点，使图像中的细节部分受到影响，导致细节损失、边缘模糊等现象，图像整体看起来变得更加粗糙。人眼在观察图像时感觉到更多的视觉混乱，降低图像清晰度；频域上与原图相比，得到叠加。由线性性质，时域相加，频域对应的也相加。加噪后的频谱实际上是两个频谱的叠加，这点从低频的峰值可以直观地观察到。

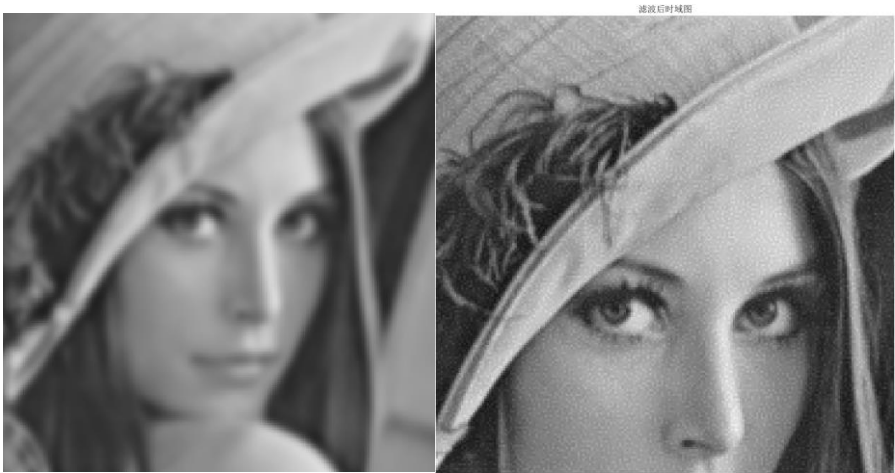
接下来对图像进行低通滤噪。



图十二 低通滤波后的时域与频域图

观察滤噪后的波形，发现图片中由高频噪声带来的噪点消失。但是，也能很明显地感觉到图片变模糊了，这是低通滤波器带来的弊端。低通滤波器会抑制图像中的高频成分，因此可以减弱或移除由高频噪声引起的图像细节损失。由于低通滤波器会移除图像中的高频成分，因此可能导致一定程度上的图像细节丢失或模糊化。这意味着一些细小的特征和纹理可能会受到影响。通过去除高频噪声，低通滤波器有助于提高图像的整体质量，在一定程度上恢复图像原有的清晰度。

如果在已知高频噪声频谱范围的情况下，可以适当增大低通滤波器的截止频率以达到一定的保留原图像细节的效果。



图十三 低通截止频率 40 与 100Hz 时域图

不难看出，100Hz 低通时的图像虽然还是存在一些噪声，但细节相较于 40Hz 时大大增加了。此时可以保存更多的细节同时进行了一定的滤波。

五、 分析与结论

实验中使用了经典的 luna 图像作为原图进行处理。通过频谱图可以观察到原图主要集中在低频，说明该图像的能量主要分布在低频区域。因此，在加入高频噪声并经过低通滤波器后，仍然能够保留图像的大部分信息。

随后，我们观察了高斯噪声的时域和频域图像。对比高斯噪声的频谱和图像的频谱可以发现，高斯噪声的频谱更为集中，高频分布更为均匀。与高斯噪声相比，原图的频谱分布不

如高斯噪声集中。在时域图像方面，高斯噪声表现为细小且局部化的随机亮度变化，可能会导致图像出现颗粒状的噪点。

通过对比高斯和高频高斯噪声的时域图像，我们可以总结出：高斯噪声是一种普遍且均匀分布的噪声，而高频高斯噪声在图像中表现为细小的、局部化的随机亮度变化，可能会导致图像出现颗粒状的噪点。在频域上，高斯噪声的低频成分相对突出，而高通处理后的频域中，低频部分大大衰减，高频则相对突出。

接下来，将高斯噪声与原图数据相加，得到了加噪图像。可以观察到加噪后的图像出现了一些噪点，导致图像的细节部分受到影响，使图像整体看起来变得更加粗糙。加噪后的频谱实际上是两个频谱的叠加，这一点可以从低频的峰值上直观地观察出来。由于加噪过程是线性的，所以在时域上相加的同时，在频域上也相加。

然后，对图像进行了低通滤波。观察滤噪后的时域图像，可以看到由高频噪声引起的噪点消失了。然而，也能明显感觉到图像变得模糊了，这是低通滤波器的一个弊端。低通滤波器会抑制图像中的高频成分，从而减弱或移除由高频噪声引起的图像细节损失。然而，由于低通滤波器会移除图像中的高频成分，因此可能导致一定程度上的图像细节丢失或模糊化。这意味着一些细小的特征和纹理可能会受到影响。通过去除高频噪声，低通滤波器有助于提高图像的整体质量，但在一定程度上可能恢复图像原有的清晰度。

最后，在已知高频噪声频谱范围的情况下，我们可以适当增大低通滤波器的截止频率，以达到保留原图像细节的效果。根据实验结果，对比了低通截止频率为 40Hz 和 100Hz 时的时域图像，可以发现在 100Hz 的低通滤波下，图像的细节相较于 40Hz 的滤波更加清晰。因此，适当调整低通滤波器的截止频率可以在保留更多细节的同时进行滤波处理。综上所述，根据实验结果，可以利用图像处理中的低通滤波技术有效地去除高频噪声，减少图像细节损失和模糊化，并通过适当调整低通滤波器的截止频率来保留更多的图像细节。这些结论对于图像处理和去噪技术的应用具有一定的指导意义。

六、收获与反思

整体看来，这是一个基础的图像处理方面的实验。原理并不复杂，但实际操作起来还是遇到了不少问题。其中最大的问题是数据的格式方面，导入图片时 matlab 默认的是 uint8 型，而处理时多用的是 double 型。这是我在尝试了几个代码后发现高斯的时域图经常是全黑，因为生成的 double 型在 0-1 范围，而 uint8 是 0-255。在没有转换的情况下，0-1 对应的就是灰度中最黑的部分。还有一个不足在于频谱图输出方面，频谱是在做了坐标变换后输出的，即坐标移到中心，但输出图像时坐标系并没有变换，导致输出的图像实际零点在图中心但标注的在边缘上。

做这个实验时也是复习数字信号处理准备期末考的时候，白天复习晚上回宿舍研究代码，颇有种理论与实际相结合的感觉，也有不少的收获。上手了 matlab 处理图像一些语句和函数，对于知识有了更深的了解，以及对知识的实际运用也有了更深的理解，学以致用。

七、原代码：

```
%读取图像
I = imread('lena.bmp');%读取图像
[m, n, y] = size(I);%图像参数
%图像预处理
```



```

%subplot(6,2,1);
figure
imshow(I);
title('原图');
%F=rgb2gray(I);%rgb 转灰度图
F=I;
F=im2double(F);%灰度 uint8 转 double

%subplot(6,2,3);
figure
imshow(F);
title('灰度图');

F_fft = fft2(F);%求频域
%subplot(6,2,4);
figure
mesh(abs(fftshift(F_fft)));%频域图,将低频移到中间
title('频域图');
%高频高斯噪声
N=gauss(m,n);
%subplot(6,2,5);
figure
imshow(N);
title('高斯时域')

N_fft=fft2(N);%噪声频域
%subplot(6,2,6);
figure
mesh(abs(fftshift(N_fft)));
title('高斯频域')

N_high=butter_high(N);%高斯噪声过高频滤波器
%subplot(6,2,7);
figure
imshow(N_high);
title('高频高斯时域')

N_high_fft=fft2(N_high);
%subplot(6,2,8);
figure
mesh(abs(fftshift(N_high_fft)));
title('高频高斯频域')
%噪声与图像叠加
Y=N_high+F;%加性噪声

```

```
%subplot(6,2,9);  
figure  
imshow(Y);  
title('加噪图');
```

```
Y_fft=fft2(Y);%加噪图的频谱
```

```
%subplot(6,2,10);
```

```
figure
```

```
mesh(abs(fftshift(Y_fft)));
```

```
title('加噪频域')
```

```
%图像低通处理
```

```
Y_low=butter_low(Y);%过低通滤波器
```

```
%subplot(6,2,11);
```

```
figure
```

```
imshow(Y_low);
```

```
title('滤波后时域图')
```

```
Y_low_fft=fft2(Y_low);%通过低通滤波器后的频谱
```

```
%subplot(6,2,12);
```

```
figure
```

```
mesh(abs(fftshift(Y_low_fft)));
```

```
title('滤波后频域')
```

```
function output=gauss(m,n)
```

```
%l=uint8(256*rand(m,n));%法 1、直接用 rand 随机。
```

```
l=zeros(m,n);%法 2、先生成一个 0 矩阵，再用 imnoise 给图片加高斯噪声
```

```
% 添加高斯白噪声给图像 l,均值为 m,方差为 v.默认 m = 0,v = 0.01
```

```
J=imnoise(l,'gaussian',0,0.5);
```

```
output=J;
```

```
end
```

```
function output=butter_low(l)
```

```
Y=fft2(im2double(l));%fft，注意进行 uint8 转 double 型
```

```
Y=fftshift(Y);%将坐标系移动到中心
```

```
[M,N]=size(Y);%获得图像的高度和宽度
```

```
h=zeros(M,N);%滤波器函数
```

```
%图像中心点
```

```
M0=M/2;
```

```
N0=N/2;
```

```
d0=200;%40、100
```

```
%巴特沃斯滤波器的阶数
```

```
n_0=2;
```

```
for x=1:M
```

```

        for y=1:N
            distance=sqrt((x-M0)^2+(y-N0)^2);
            h(x,y)=1/(1+(distance/d0)^(2*n_0));%滤波器函数
        end
    end
    %滤波后结果
    res=h.*Y;%频域相乘
    output=real(ifft2(ifftshift(res))));%输出
end

```

```

function output=butter_high(l)
Y=fft2(im2double(l));%傅里叶变换
Y=fftshift(Y);%坐标系转换
[M,N]=size(Y);
h=zeros(M,N);%滤波器函数
%图像中心点
M0=M/2;
N0=N/2;
d0=300;
%巴特沃斯滤波器的阶数
n_0=2;
for x=1:M
    for y=1:N
        distance=sqrt((x-M0)^2+(y-N0)^2);
        h(x,y)=1/(1+(d0/distance)^(2*n_0));%滤波器公式
    end
end
%滤波后结果
res=h.*Y;%频域相乘
output=real(ifft2(ifftshift(res))));%转换回时域、输出
end

```

八、 参考文献

数字信号处理（第四版）高西全 丁玉美编著
https://blog.csdn.net/qq_45767476/article/details/115561686
https://blog.csdn.net/qq_44111805/article/details/126346525