贵州大学实验报告

学院：大数据与信息工程学院　　专业：通信工程　　班级：通信 152 班

姓名	朱桐	学号	1500890087	实验组	
实验时间	2018/06/13	指导教师	刘洪	成绩	
实验项目名称	实验三 数字图像的频域增强处理				

实验目的

1. 通过本次实验的学习使学生熟悉和掌握数字图像处理中频域增强的基本原理及应用。

实验要求

1. 认真阅读实验手册，严格遵守实验室安全管理规定，规范使用设备；
2. 保存实验过程中的输出图像及结果；
3. 认真撰写实验报告并及时上交。

实验原理

1、 频域增强原理：

设函数 f (x, y) 与线性位不变算子 h(x, y) 的卷积结果是 g(x, y) ，即

$$g(x, y) = h(x, y) * f(x, y)$$

那么根据卷积定理在频域有

$$G(u,v) = H(u,v)F(u,v)$$

其中， G(u,v)、H(u,v)、F(u,v)分别是 g(x, y)、h(x, y)、f (x, y)的傅立叶变换。

频域增强的主要步骤：

（1） 计算所需增强图像的傅立叶变换；

（2） 将其与一个（根据需要设计的）转移函数相乘；

（3） 再将结果进行傅立叶反变换以得到增强的图。

2、 低通滤波器

图像的能量大部分集中在幅度谱的低频和中频部分，而图像的边缘和噪声对应于高频部分，所以低通滤波器可以降低噪声的影响。

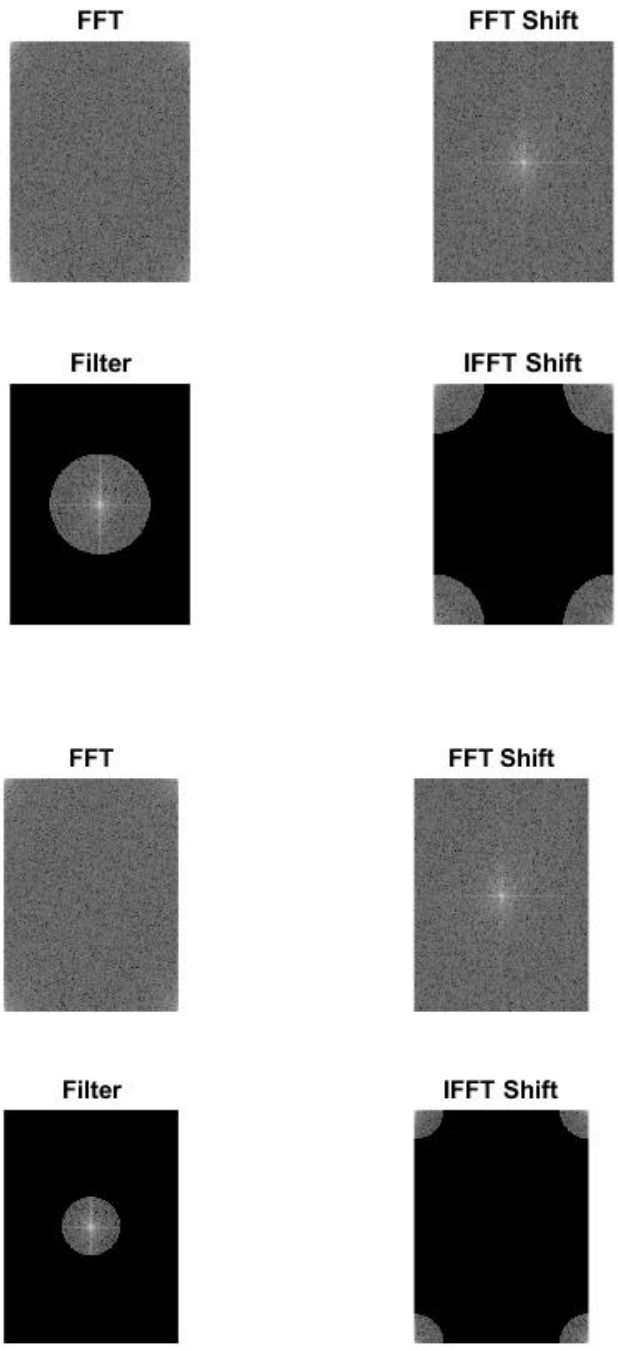
（1）理想低通滤波器数字图像处理实验指导书

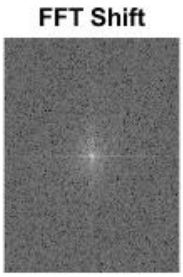
	$$H(u,v)=\begin{cases}1\,D(u,v)\le D_0\\0\,D(u,v)>D_0\end{cases}$$ 其中 D0 是一非负整数，D(u,v) 是从点(u,v) 到频率平面原点的距离。 $$D(u,v)=\sqrt{u^2+v^2}$$ （2）巴特渥斯低通滤波器 $$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$ 其中： n 是阶数。 3、 高通滤波器 图像中的边缘对应高频分量，所以要锐化图像可以采用高通滤波。 （1）理想高通滤波器 $$H(u,v)=\begin{cases}0\,D(u,v)\le D_0\\1\,D(u,v)>D_0\end{cases}$$ （2）巴特渥斯高通滤波器 $$H(u,v)=\frac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$$ 其中： n 是阶数。
实验仪器	1. 硬件环境 计算机 一台 2. 软件环境 MATLAB 2016b Windows10 64bit Professional
实验步骤	1. 编写程序。读取图像，并对其加入“盐和胡椒”噪声。对噪声图像做理想低通滤波和巴特渥斯低通滤波处理。比较选择不同参数时的实验结果； 2. 编写程序。读取图像。对图像做理想高通滤波和巴特渥斯高通滤波处理。 比较选择不同参数时的实验结果。

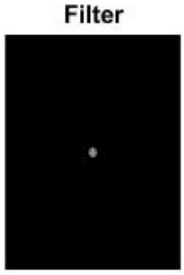
实验内容	1. 利用低通滤波器对图像增强； 2. 利用高通滤波器对图像增强。

实验数据	close all
	clear, clc
	im = imread('../data/jingjing.jpg');
	% im = imread('peppers.png');
	imGray = rgb2gray(im);
	imNoise = imnoise(imGray,'salt & pepper',0.02);
	
	Ideal Low Pass Filter
	
	d1 = 100;
	imIdeal1 = idealLowPassFilter(imNoise, d1);
	d2 = 60;
	imIdeal2 = idealLowPassFilter(imNoise, d2);
	d3 = 10;
	imIdeal3 = idealLowPassFilter(imNoise, d3);
	
	figure
	subplot(2,2,1)
	imshow(imNoise)
	title('Noisy Image')
	subplot(2,2,2)
	imshow(abs(imIdeal1), [])
	title(sprintf('Image After Ideal Low-Pass Filter %f', d1))
	subplot(2,2,3)

```
imshow(abs(imIdeal2), [])
title(sprintf('Image After Ideal Low-Pass Filter %f', d2))
subplot(2,2,4)
imshow(abs(imIdeal3), [])
title(sprintf('Image After Ideal Low-Pass Filter %f', d3))
```



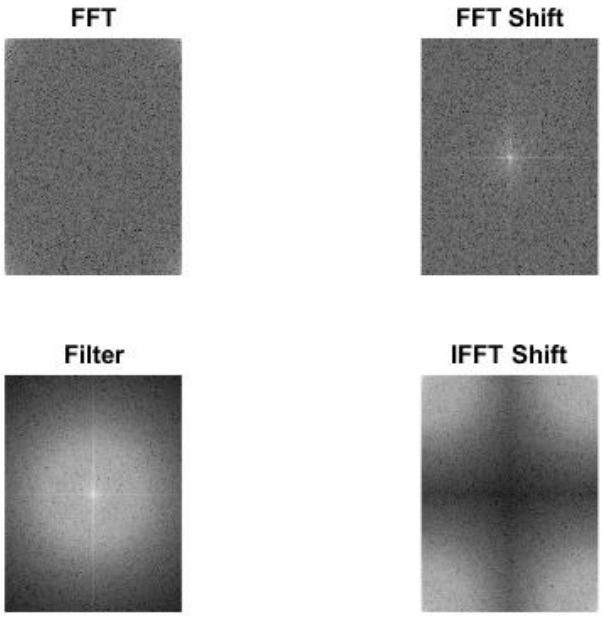




Butterworth Low-Pass Filter

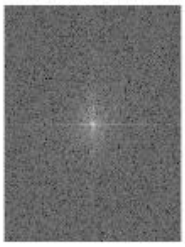
```
d1 = 100;
imButter1 = butterWorthLowPass(imNoise, d1, 5);
d2 = 60;
imButter2 = butterWorthLowPass(imNoise, d2, 5);
d3 = 10;
imButter3 = butterWorthLowPass(imNoise, d3, 5);

figure
```

```
subplot(2,2,1)
imshow(imNoise)
title('Noisy Image')
subplot(2,2,2)
imshow(abs(imButter1), [])
title(sprintf('Image After Butterworth Low-Pass Filter %f', d1))
subplot(2,2,3)
imshow(abs(imButter2), [])
title(sprintf('Image After Butterworth Low-Pass Filter %f', d2))
subplot(2,2,4)
imshow(abs(imButter3), [])
title(sprintf('Image After Butterworth Low-Pass Filter %f', d3))
```



FFT

FFT Shift

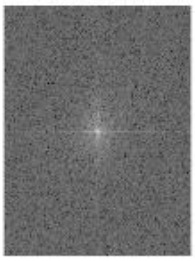


Filter

IFFT Shift

FFT

FFT Shift



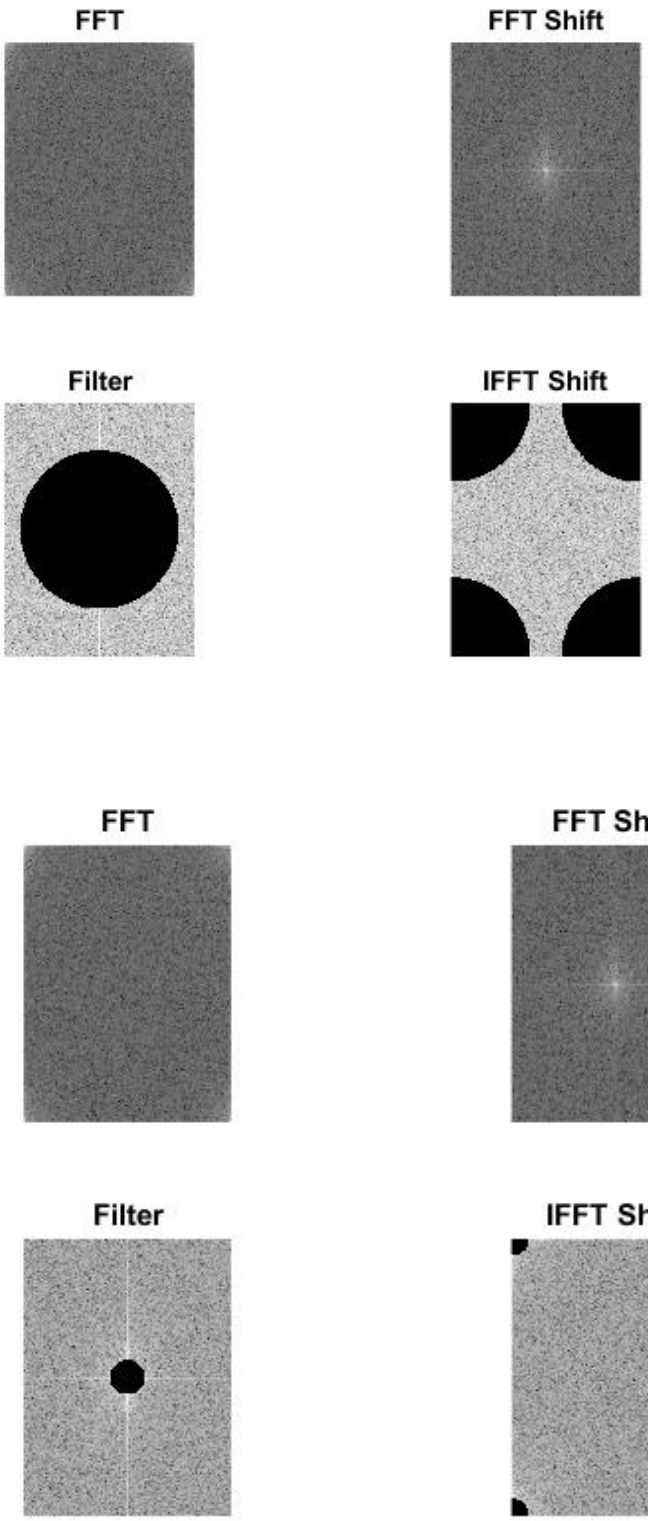
Filter

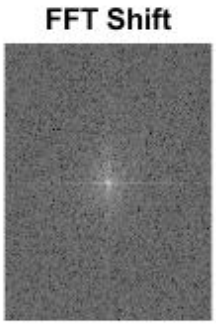
IFFT Shift

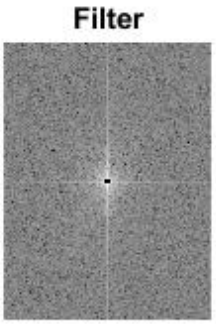
Ideal High Pass Filter

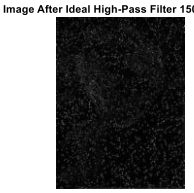
```
d1 = 150;
imIdeal1 = idealHighPassFilter(imNoise, d1);
d2 = 30;
imIdeal2 = idealHighPassFilter(imNoise, d2);
d3 = 5;
imIdeal3 = idealHighPassFilter(imNoise, d3);

figure
subplot(2,2,1)
imshow(imNoise)
title('Noisy Image')
subplot(2,2,2)
imshow(abs(imIdeal1), [])
title(sprintf('Image After Ideal High-Pass Filter %f', d1))
subplot(2,2,3)
imshow(abs(imIdeal2), [])
title(sprintf('Image After Ideal High-Pass Filter %f', d2))
subplot(2,2,4)
```

```
imshow(abs(imIdeal3), [])
title(sprintf('Image After Ideal Highs-Pass Filter %f', d3))
```





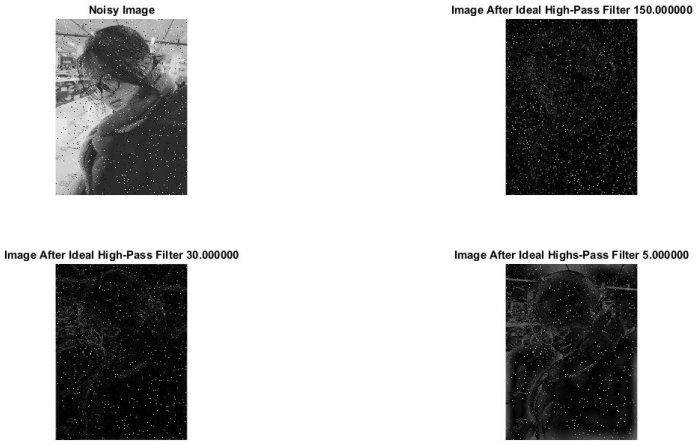




Butterworth High-Pass Filter

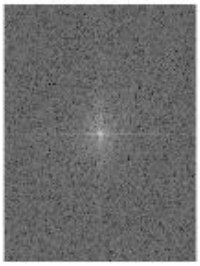
```
d1 = 100;
imButter1 = butterWorthHighPass(imNoise, d1, 5);
d2 = 60;
imButter2 = butterWorthHighPass(imNoise, d2, 5);
```

```
d3 = 10;
imButter3 = butterWorthHighPass(imNoise, d3, 5);


figure
subplot(2,2,1)
imshow(imNoise)
title('Noisy Image')
subplot(2,2,2)
imshow(abs(imButter1), [])
title(sprintf('Image After Butterworth High-Pass Filter %f', d1))
subplot(2,2,3)
imshow(abs(imButter2), [])
title(sprintf('Image After Butterworth High-Pass Filter %f', d2))
subplot(2,2,4)
imshow(abs(imButter3), [])
title(sprintf('Image After Butterworth High-Pass Filter %f', d3))
```

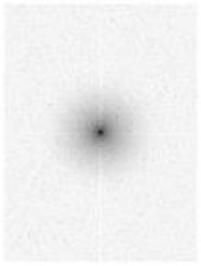


FFT

FFT Shift



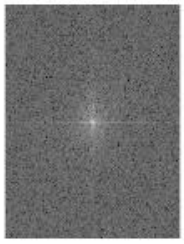
Filter



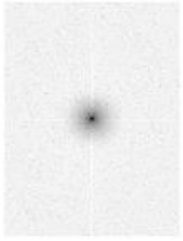
IFFT Shift

FFT

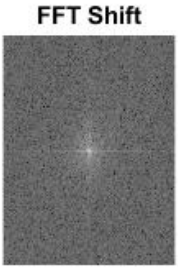
FFT Shift

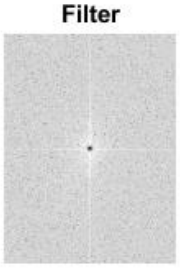


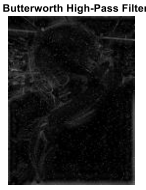
Filter



IFFT Shift








```
function re = idealLowPassFilter(imGray, d)
imFFT = fft2(imGray);
imFFTShift = fftshift(imFFT);
[rows, cols] = size(imFFTShift);
D = zeros(rows, cols);
for row = 1 : rows
    for col = 1 : cols
       D(row, col) = sqrt(double((row - int32(rows/2))^2 + (col -
int32(cols/2))^2));
    end
end
figure
subplot(2,2,1)
imshow(log10(abs(imFFT)), [])
title('FFT')
```

```
subplot(2,2,2)
imshow(log10(abs(imFFTShift)), [])
title('FFT Shift')
imFFTShift(D > d) = 0;
antiShift = ifftshift(imFFTShift);
iImFFT = ifft2(antiShift);
re = iImFFT;
subplot(2,2,3)
imshow(log10(abs(imFFTShift)), [])
title('Filter')
subplot(2,2,4)
imshow(log10(abs(antiShift)), [])
title('IFFT Shift')
end

function re = butterWorthLowPass(imGray, d, n)
imFFT = fft2(imGray);
imFFTShift = fftshift(imFFT);
[rows, cols] = size(imFFTShift);
D = zeros(rows, cols);
H = zeros(rows, cols);
for row = 1 : rows
   for col = 1 : cols
      D(row, col) = sqrt(double((row - int32(rows/2))^2 + (col -
int32(cols/2))^2));
   end
end
figure
subplot(2,2,1)
imshow(log10(abs(imFFT)), [])
title('FFT')
subplot(2,2,2)
imshow(log10(abs(imFFTShift)), [])
title('FFT Shift')
H = 1./( 1 + (D./d).^(2.*n) );
imFFTShift = imFFTShift.*H;
antiShift = ifftshift(imFFTShift);
iImFFT = ifft2(antiShift);
re = iImFFT;
subplot(2,2,3)
imshow(log10(abs(imFFTShift)), [])
title('Filter')
subplot(2,2,4)
imshow(log10(abs(antiShift)), [])
```

```
title('IFFT Shift')
end

function re = idealHighPassFilter(imGray, d)
imFFT = fft2(imGray);
imFFTShift = fftshift(imFFT);
[rows, cols] = size(imFFTShift);
D = zeros(rows, cols);
for row = 1 : rows
    for col = 1 : cols
        D(row, col) = sqrt(double((row - int32(rows/2))^2 + (col -
int32(cols/2))^2));
    end
end
figure
subplot(2,2,1)
imshow(log10(abs(imFFT)), [])
title('FFT')
subplot(2,2,2)
imshow(log10(abs(imFFTShift)), [])
title('FFT Shift')
imFFTShift(D < d) = 0;
antiShift = ifftshift(imFFTShift);
iImFFT = ifft2(antiShift);
re = iImFFT;
subplot(2,2,3)
imshow(log10(abs(imFFTShift)), [])
title('Filter')
subplot(2,2,4)
imshow(log10(abs(antiShift)), [])
title('IFFT Shift')
end

function re = butterWorthHighPass(imGray, d, n)
imFFT = fft2(imGray);
imFFTShift = fftshift(imFFT);
[rows, cols] = size(imFFTShift);
D = zeros(rows, cols);
H = zeros(rows, cols);
for row = 1 : rows
    for col = 1 : cols
        D(row, col) = sqrt(double((row - int32(rows/2))^2 + (col -
int32(cols/2))^2));
    end
```

```
end
figure
subplot(2,2,1)
imshow(log10(abs(imFFT)), [])
title('FFT')
subplot(2,2,2)
imshow(log10(abs(imFFTShift)), [])
title('FFT Shift')
H = 1./( 1 + (d./D).^(2.*n) );
imFFTShift = imFFTShift.*H;
antiShift = ifftshift(imFFTShift);
iImFFT = ifft2(antiShift);
re = iImFFT;
subplot(2,2,3)
imshow(log10(abs(imFFTShift)), [])
title('Filter')
subplot(2,2,4)
imshow(log10(abs(antiShift)), [])
title('IFFT Shift')
end
```

实验总结

通过本次实验，基本熟悉了图像处理中频域增强的基础操作及应用。本次试验中，可以清楚地看出理想与巴特沃斯低通滤波器与高通滤波器的频域处理过程。可以看出，理想滤波器得到的结果会有振铃效应。这是由于理想滤波器的截断效应造成的。若时域截断，则频域出现新的高频成分；若频域产生截断，则时域会产生明显的振铃效应。

此外，还应注意 imshow 函数的正确使用方法。imshow 函数的第二个参数最好填空矩阵[]，否则很有可能会无法显示图像。

指导教师意见

签名：　　年　　月　　日