

实验二 空间域、频率域图像增强及复原

课程名称：图像处理基础

实验学时：4 学时

综合性、设计性实验：☒是 ☒否

面向专业和班级：18 级软件工程（1 班、2 班、3 班、4 班、中澳班、卓越 1 班、卓越 2 班）

学生人数：102

任务编制人：课程负责人

实验内容

- 一、 图像增强---空域滤波
- 二、 图像增强---傅里叶变换（1）
- 三、 图像增强---傅里叶变换（2）
- 四、 图像退化和复原（1）
- 五、 图像退化和复原（2）

一、图像增强-空域滤波

(一) 实验目的

1、进一步了解 MatLab 软件/语言，学会使用 MatLab 对图像作滤波处理，使学生有机会掌握滤波算法，体会滤波效果。

2、了解几种不同滤波方式的使用和使用的场合，培养处理实际图像的能力，并为课堂教学提供配套的实践机会。

(二) 实验要求

(1)学生应当完成对于给定图像+噪声，使用平均滤波器、中值滤波器对不同强度的高斯噪声和椒盐噪声，进行滤波处理；能够正确地评价处理的结果；能够从理论上作出合理的解释。

(2)利用 MATLAB 软件实现空域滤波的程序：

```
I=imread('electric.jpg');
imshow(I);
J = imnoise(I,'gauss',0.02);           %添加高斯噪声
figure,imshow(J);
J = imnoise(I,'salt & pepper',0.02);   %添加椒盐噪声
figure,imshow(J);
ave1=fspecial('average',3)            %产生 3×3 的均值模版
ave2=fspecial('average',5)            %产生 5×5 的均值模版
K = filter2(ave1,J)/255;
%均值滤波 3×3，这一步会报错，原因是图像是 RGB 格式的，需要将图像转换成灰度图像才可以，考虑一下怎么将 rgb 转换成灰度图像；也用 imfilter 进行一次滤波，这里不需要将 RGB 转成灰度图像，
imfilter(J,ave1)
% imshow()显示图像时对 double 型是认为在 0~1 范围内,即大于 1 时都是显示为白色,而 imshow 显示 uint8 型时是 0~255 范围.而经过运算的范围在 0~255 之间的 double 型数据就被不正常得显示为白色图像了.
figure,imshow(K);
L = filter2(ave2,J)/255;                %均值滤波 5×5
figure,imshow(L);
M = medfilt2(J,[3 3]);                  %中值滤波 3×3 模板
figure,imshow(M);
N = medfilt2(J,[4 4]);                  %中值滤波 4×4 模板
figure,imshow(N);
```

(三) 实验设备与软件

(1) PC 计算机系统

(2) MatLab 软件/语言包括图像处理工具箱(Image Processing Toolbox)

(3) 实验所需要的图片

(四) 实验内容与步骤

- a) 调入并显示原始图像 `electric.jpg`。
- b) 利用 `imnoise` 命令在图像 `electric.jpg` 上加入高斯(gaussian) 噪声
- c) 利用预定义函数 `fspecial` 命令产生平均(average)滤波器,并手动产生一个下述滤波器

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

d) 分别采用 3x3 和 5x5 的模板, 分别用平均滤波器以及中值滤波器, 对加入噪声的图像进行处理并观察不同噪声水平下, 上述滤波器处理的结果;

e) 选择不同大小的模板, 对加入某一固定噪声水平噪声的图像进行处理, 观察上述滤波器处理的结果。

f) 利用 `imnoise` 命令在图像 `electric.jpg` 上加入椒盐噪声(salt & pepper)

g) 重复 c)~ e) 的步骤

h) 输出全部结果并进行讨论。

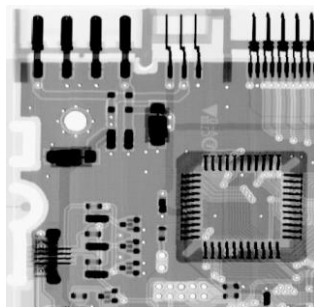
(五) 思考题/问答题

- (1) 简述高斯噪声和椒盐噪声的特点。
- (2) 结合实验内容, 定性评价平均滤波器/中值滤波器对高斯噪声和椒盐噪声的去噪效果?
- (3) 结合实验内容, 定性评价滤波窗口对去噪效果的影响?

(六) 实验报告要求

描述实验的基本步骤, 用数据和图片给出各个步骤中取得的实验结果, 并进行必要的讨论, 必须包括原始图像及其计算/处理后的图像。

(七) 实验图像



`electric.jpg` (原始图像)

二、图像增强-傅里叶变换(1)

(一) 实验目的

熟悉及掌握图像的变换原理及性质，实现图像的傅里叶变换。

(二) 相关函数

(1) 图像的变换

➤ fft2

fft2 函数用于数字图像的二维傅立叶变换，如：

```
i=imread('e:\w01.jpg');  
j=fft2(i);
```

由于 MATLAB 无法显示复数图像，因此变换后的结果还需进行求模运算，即调用 abs 函数。

之后常常还进行对数变换，即调用 log 函数，以减缓傅里叶谱的快速衰减，更好地显示高频信息。

➤ ifft2

ifft2 函数用于数字图像的二维傅立叶反变换，如：

```
i=imread('e:\w01.jpg');  
j=fft2(i);  
k=ifft2(j);
```

➤ fftshift

用于将变换后图像频谱中心从矩阵的原点移动到矩阵的中心

```
B=fftshift (i)
```

利用 fft2 计算二维卷积

利用 fft2 函数可以计算二维卷积,如：

```
a=[8,1,6;3,5,7;4,9,2];  
b=[1,1,1;1,1,1;1,1,1];  
a(8,8)=0;  
b(8,8)=0;  
c=ifft2(fft2(a).*fft2(b));  
c=c(1:5,1:5);
```

利用 conv2(二维卷积函数)校验，如：

```
a=[8,1,6;3,5,7;4,9,2];  
b=[1,1,1;1,1,1;1,1,1];
```

```
c=conv2(a,b);
```

(三) 实验内容

1、对一幅图像进行平移，显示原始图像与处理后图像，分别对其进行傅里叶变换，显示变换后结果，分析原图的傅里叶谱与平移后傅里叶频谱的对应关系。

```
s=imread('1.jpg');%读入原图像
i=rgb2gray(s);
i=double(i);
j=fft2(i);                %傅里叶变换
k=fftshift(j);            % 直流分量移到频谱中心
l=log(abs(k));            %对数变换
m=fftshift(j);            %直流分量移到频谱中心
RR=real(m);               %取傅里叶变换的实部
II=imag(m);               %取傅里叶变换的虚部
A=sqrt(RR.^2+II.^2);      %计算频谱幅值
A=(A-min(min(A)))/(max(max(A)))*255; % 归一化
b=circshift(s,[800 450]); %对图像矩阵 im 中的数据进行移位操作
b=rgb2gray(b);
b=double(b);
c=fft2(b);                %傅里叶变换
e=fftshift(c);            % 直流分量移到频谱中心
l=log(abs(e));            %对数变换
f=fftshift(c);            %直流分量移到频谱中心
WW=real(f);               %取傅里叶变换的实部 B
ZZ=imag(f);               %取傅里叶变换的虚部
B=sqrt(WW.^2+ZZ.^2);      %计算频谱幅值
B=(B-min(min(B)))/(max(max(B)))*255; % 归一化
subplot(2,2,1);imshow(s);title('原图像')
subplot(2,2,2);imshow(uint8(b));title('平移图像')
subplot(2,2,3);imshow(A);title('离散傅里叶频谱');
subplot(2,2,4);imshow(B);title('平移图像离散傅里叶频谱')
```



2、对一幅图像进行旋转，显示原始图像与处理后图像，分别对其进行傅里叶变换，显示变换后结果，分析原图的傅里叶谱与旋转后傅里叶频谱的对应关系。

```
s=imread('1.jpg');%读入原图像
i=rgb2gray(s);
i=double(i);
j=fft2(i); %傅里叶变换
k=fftshift(j); % 直流分量移到频谱中心
l=log(abs(k)); %对数变换
m=fftshift(j); %直流分量移到频谱中心
RR=real(m); %取傅里叶变换的实部
II=imag(m); %取傅里叶变换的虚部
A=sqrt(RR.^2+II.^2); %计算频谱幅值
A=(A-min(min(A)))/(max(max(A)))*255; % 归一化
b=imrotate(s, -90); %对图像矩阵 im 中的数据进行移位操作
b=rgb2gray(b);
b=double(b);
c=fft2(b); %傅里叶变换
e=fftshift(c); % 直流分量移到频谱中心
l=log(abs(e)); %对数变换
f=fftshift(c); %直流分量移到频谱中心
WW=real(f); %取傅里叶变换的实部 B
ZZ=imag(f); %取傅里叶变换的虚部
B=sqrt(WW.^2+ZZ.^2); %计算频谱幅值
B=(B-min(min(B)))/(max(max(B)))*255; % 归一化
subplot(2,2,1);imshow(s);title('原图像')
subplot(2,2,2);imshow(uint8(b));title('平移图像')
subplot(2,2,3);imshow(A);title('离散傅里叶频谱');
subplot(2,2,4);imshow(B);title('平移图像离散傅里叶频谱')
```

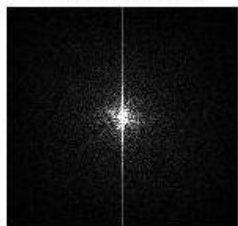
原图像



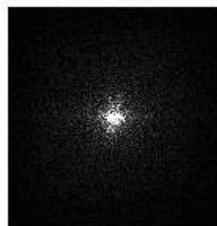
平移图像



离散傅里叶频谱



平移图像离散傅里叶频谱



三、图像增强—傅立叶变换(2)

(一) 实验目的

1. 掌握怎样利用傅立叶变换进行频域滤波
2. 掌握频域滤波的概念及方法
3. 熟练掌握频域空间的各类滤波器
4. 利用 MATLAB 程序进行频域滤波

(二) 实验原理及知识点

频域滤波分为低通滤波和高通滤波两类, 对应的滤波器分别为低通滤波器和高通滤波器。频域低通过滤的基本思想:

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

$F(u, v)$ 是需要钝化图像的傅立叶变换形式, $H(u, v)$ 是选取的一个低通过滤器变换函数, $G(u, v)$ 是通过 $H(u, v)$ 减少 $F(u, v)$ 的高频部分来得到的结果, 运用傅立叶逆变换得到钝化后的图像。

理想地通滤波器 (ILPF) 具有传递函数:

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

其中, D_0 为指定的非负数, $D(u, v)$ 为 (u, v) 到滤波器的中心的距离。 $D(u, v) = D_0$ 的点的轨迹为一个圆。

n 阶巴特沃兹低通滤波器 (BLPF) (在距离原点 D_0 处出现截至频率) 的传递函数为

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

与理想地通滤波器不同的是, 巴特沃兹率通滤波器的传递函数并不是在 D_0 处突然不连续。

高斯低通滤波器 (GLPF) 的传递函数为

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

其中, σ 为标准差。

相应的高通滤波器也包括: 理想高通滤波器、 n 阶巴特沃兹高通滤波器、高斯高通滤波器。给定一个低通滤波器的传递函数 $H_{lp}(u, v)$, 通过使用如下的简单关系, 可以获得相应高通滤波器的传递函数: $H_{hp} = 1 - H_{lp}(u, v)$

(三) 利用 MATLAB 实现频域滤波的程序

```
f=imread('room.jpg');  
f=rgb2gray(f);  
F=fft2(f);           %对图像进行傅立叶变换
```



```

%对变换后图像进行队数变化，并对其坐标平移，使其中心化
S=fftshift(log(1+abs(F)));
S=gscale(S);          %将频谱图像标度在 0-256 的范围内，需要加入 gscale.m 文件
imshow(S)              %显示频谱图像
h=fspecial('sobel');   %产生空间'sobel'模版
freqz2(h)              %查看相应频域滤波器的图像
PQ=paddedsize(size(f)); %产生滤波时所需大小的矩阵，需要加入 paddedsize.m 文件
H=freqz2(h,PQ(1),PQ(2)); %产生频域中的'sobel'滤波器
H1=ifftshift(H);       %重排数据序列，使得原点位于频率矩阵的左上角
imshow(abs(H),[])      %以图形形式显示滤波器
figure,imshow(abs(H1),[])
gs=imfilter(double(f),h); %用模版 h 进行空域滤波
gf=dftfilt(f,H1);       %用滤波器对图像进行频域滤波，需要加入 dftfilt.m 文件
figure,imshow(gs,[])
figure,imshow(gf,[])
figure,imshow(abs(gs),[])
figure,imshow(abs(gf),[])

f=imread('number.jpg'); %读取图片
f=rgb2gray(f);
PQ=paddedsize(size(f)); %产生滤波时所需大小的矩阵
D0=0.05*PQ(1);         %设定高斯高通滤波器的阈值
H=hpfilter2('gaussian',PQ(1),PQ(2),D0); %产生高斯高通滤波器，需要加入 hpfilter2.m 和 lpfilter.m
g=dftfilt(f,H);         %对图像进行滤波
figure,imshow(f)        %显示原图像
figure,imshow(g,[])     %显示滤波后图像

```

(四) 实验步骤

1. 调入并显示所需的图片；
2. 利用 MATLAB 提供的低通滤波器实现图像信号的滤波运算，并与空间滤波进行比较。
3. 利用 MATLAB 提供的高通滤波器对图像进行处理。
4. 记录和整理实验报告。

(五) 实验报告内容

1. 叙述实验过程；
2. 提交实验的原始图像和结果图像。

(六) 实验报告要求

描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果，并进行必要的讨论，必须包括原始图像及其计算/处理后的图像。

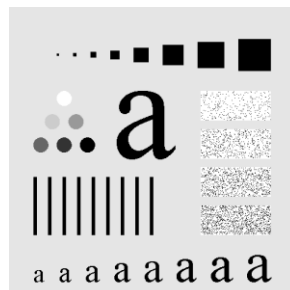
(七) 思考题

1. 结合实验，评价频域滤波有哪些优点？
2. 在频域滤波过程中需要注意哪些事项？

(八) 实验图片



room.jpg



number.jpg

四、图象退化和复原(1)

(一) 实验目的

本次实验进行的主要是对图像复原进行有关讨论。在图像形成过程中，由于成像系统等各种外部因素的影响，可能导致图像的质量下降，发生所谓的“图像退化”。与图像增强相类似，图像复原的目的是改善图像的质量，便于人眼辨识，增强表达效果。在进行图像复原的过程中，必要的先验知识十分重要，可以非常方便的将已退化的图像恢复本来的面目，但是有些时候图像的先验知识我们并不容易获取，所以与此相对应有盲卷积等方法来帮助我们进行图像复原。引起图像退化的因素很多，包括有光学系统、运动等造成的图像模糊，电路传递中、光学因素的影响噪声等等。掌握了解相关的数学模型对于图像复原至关重要，许多复原的方法也是建立在比较严格的数学推导上的。

通过本次实验，主要期望达到以下目的：

- 1、熟练掌握数字图像复原的基本概念和基本理论原理；
- 2、深入理解图像退化的影响原因和对应的处理方法，加强综合分析问题、解决实际问题的能力；
- 3、掌握各种使用的图像复原的相关技术，使经处理后的图像更加适应人眼的视觉与心理；
- 4、巩固使用 **matlab** 软件处理数字图像问题的能力，加强巩固编程能力 通过实验，了解到图像恢复和图像增强的区别还在于图像增强不考虑图像时如何退化的，而是试图采用各种技术来增强图像的视觉效果，因此图像增强可以不顾增强后的图像是否失真，只要看着舒服就基本可以满足要求，而图像复原则对图像退化的机制和过程邓先验知识要求较为必要，很多情况下先验知识如果全面准确，甚至可以完全恢复原始图像，达到非常好的效果，据此找出相应的你处理方法，从而得到恢复的图像。对于一幅图像，可以先进行复原处理，在进行图像增强处理，两者相互结合，图像的效果可以达到更为理想的效果。

(二) 核心代码及运行后截图

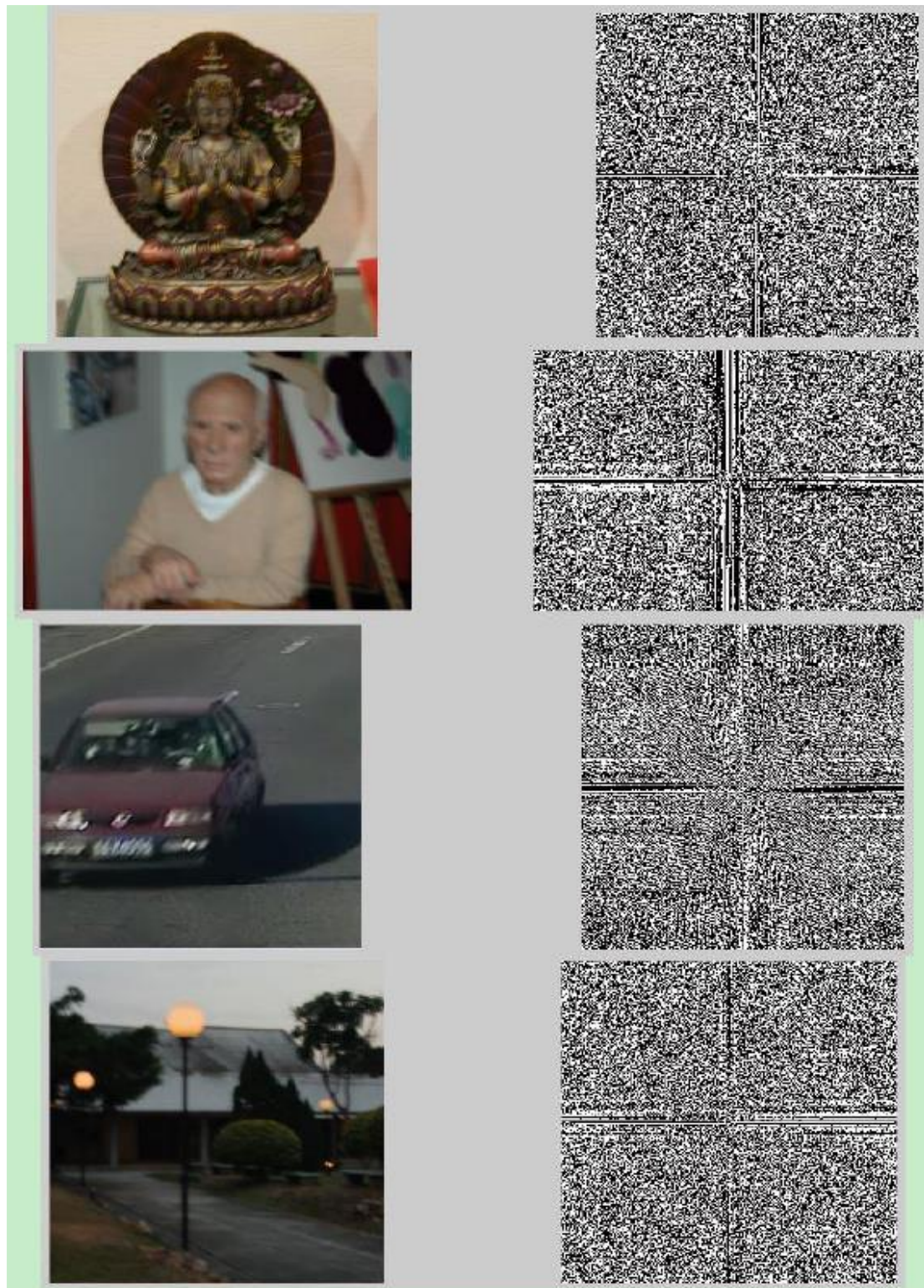
1. 频谱噪声分析

在进行图像复原之前，首先对各个图像做傅里叶变换在频谱中观察分布情况，由此判断噪声或模糊类型。

代码如下：

```
i1=imread('statBlur.png');
i2=imread('s1.bmp');
i3=imread('picassoBlurImage.png');
i4=imread('nvblurImage.png');
o1=fftshift(fft2(rgb2gray(i1)));o2=fftshift(fft2(rgb2gray(i2)));
o3=fftshift(fft2(rgb2gray(i3)));o4=fftshift(fft2(rgb2gray(i4)));
figure,subplot(221),imshow(i1),subplot(222),imshow(o1),subplot(223),i
mshow(i2),...
subplot(224),imshow(o2);
figure,subplot(221),imshow(i3),subplot(222),imshow(o3),subplot(223),i
mshow(i4),...
subplot(224),imshow(o4);
```

结果如下：



分析说明：有傅里叶频谱可以看到，在毕加索频谱的数值中心轴线处出现了明显的亮条纹现象，推测可能是原图像存在水平方向的运动模糊和竖直方向较少的运动模糊，在汽车的频谱中也可以推测原图像中存在水平方向的少许运动模糊，在后面的进一步处理中我们可以看到佛像存在着高斯噪声，若要获得较为好的复原图像，可先对图像进行噪声处理，在进行图像复原，可以达到较为理想的效果。

2. 四幅图像分别处理

注意事项：本次实验中涉及到一些人为设置的参数，大家可以自行调节进行实验，实验代码已经经过测试，不存在代码错误的问题。

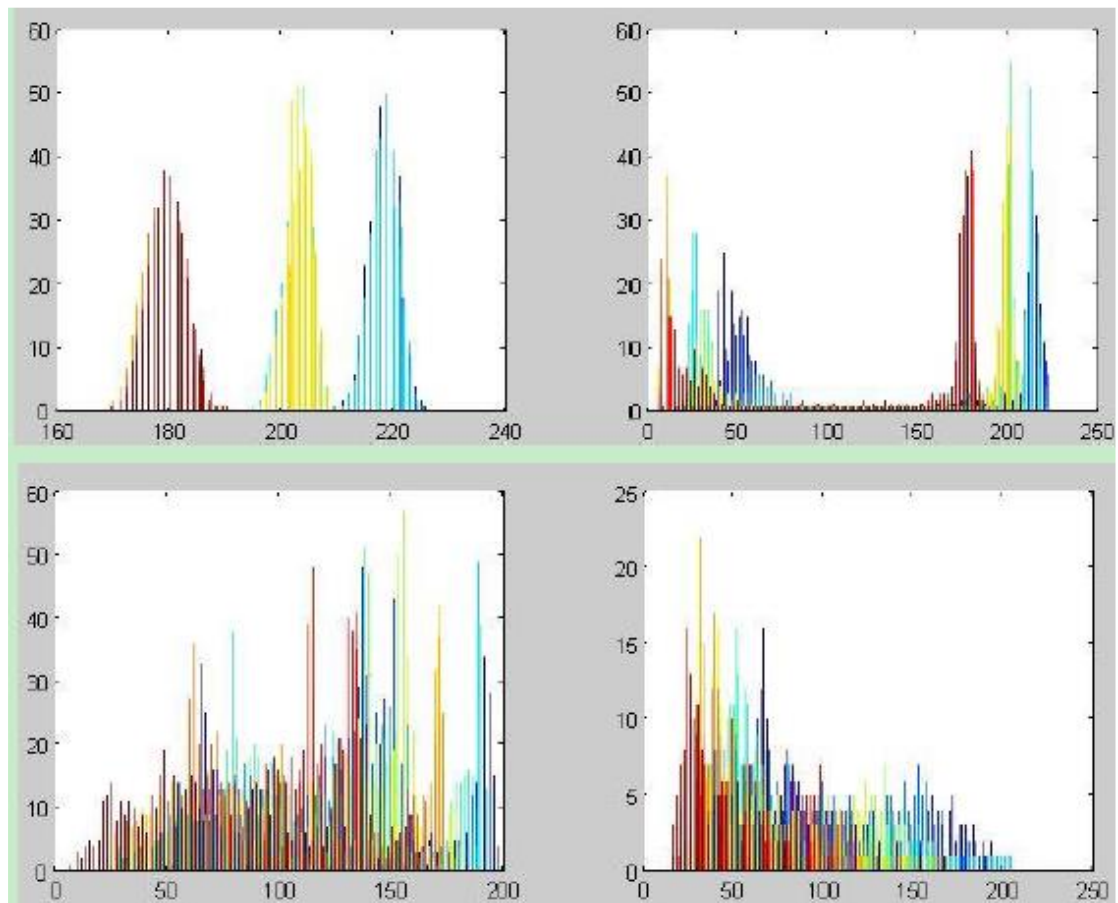
2.1 佛像复原

2.1.1 噪声分析

首先分析佛像图像是否具有噪声，在图像中较为平坦均匀处选取矩形，做直方图处理，观察直方图的分布来确定是否含有噪声以及噪声的类型。

代码如下：

```
im=imread('statBlur.png');[m,n,h]=size(im);
f11=ones(190,162,3);f22=ones(130,130,3);f33=ones(100,100,3);f44=ones(
70,70,3);
for i=1:190
for j=1:162
for k=1:3
f11(i,j,k)=im(i,j,k);
end
end
end
for i=1:130
for j=501:630
for k=1:3
f22(i,j-500,k)=im(i,j,k);
end
end
end
for i=721:870
for j=11:170
for k=1:3
f33(i-720,j-10,k)=im(i,j,k);
end
end
end
for i=761:830
for j=561:630
for k=1:3
f44(i-760,j-560,k)=im(i,j,k);
end
end
end
figure,subplot(221),hist(f11,100),subplot(222),hist(f22,100),subplot(
223),...
hist(f33,100),subplot(224),hist(f44,100);
运行结果：
```

分析说明：可以发现在第一幅图中，R、G、B三个分量均呈现钟形分布，这是高斯噪声的特点，所以初步判断原始图像中含有高斯噪声，所以对图像进行高斯噪声去除处理后再进行图像还原可以得到较为良好的效果。

2.1.2 大气湍流处理

函数代码如下：

```
function output=daqituan( image ,k )% k=0.00011
%函数daqituan 对输入的图像进行大气湍流处理
%参数image 输入的待处理图像
%参数k 大气湍流模型常数i=fft2(double(image));
G=fftshift(i);[m, n, h] = size(i);
H=zeros(m,n);
for i=1:m
for j=1:n
if sqrt((i-m/2)^2+(j-n/2)^2)<100
H(i,j)=exp(-k*(i^2+j^2)^(5/6));
end
end
end
for i=1:3
out(:, :, i) = G(:, :, i) .* H;
end
```

```

out = real(ifft2(ifftshift(out)));
figure,subplot(121),imshow(image),title('原始图像'),subplot(122),...
imshow(out,[]),title('大气湍流复原图像');
end

```

运行结果：



分析说明：有运行结果可知，复原的效果较为理想。

2.2 路灯复原

2.2.1 选取部分进行噪声分析

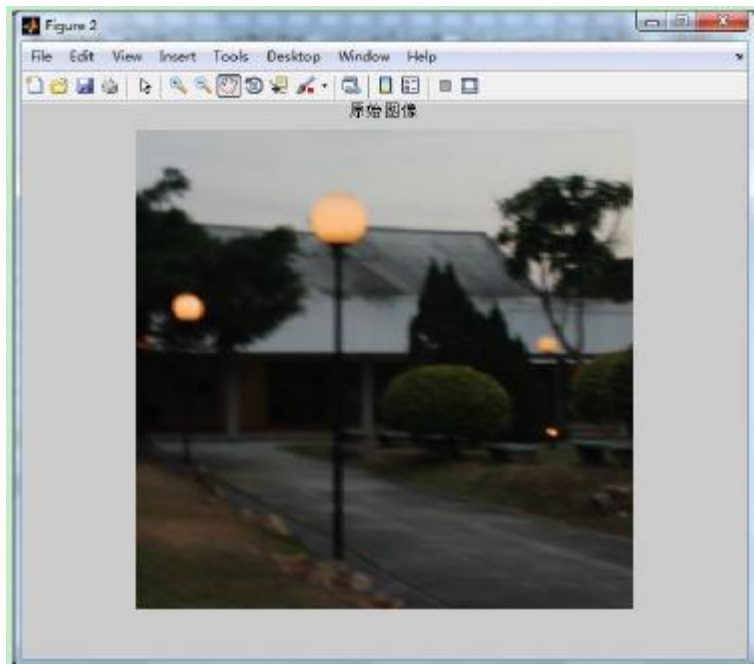
代码如下：

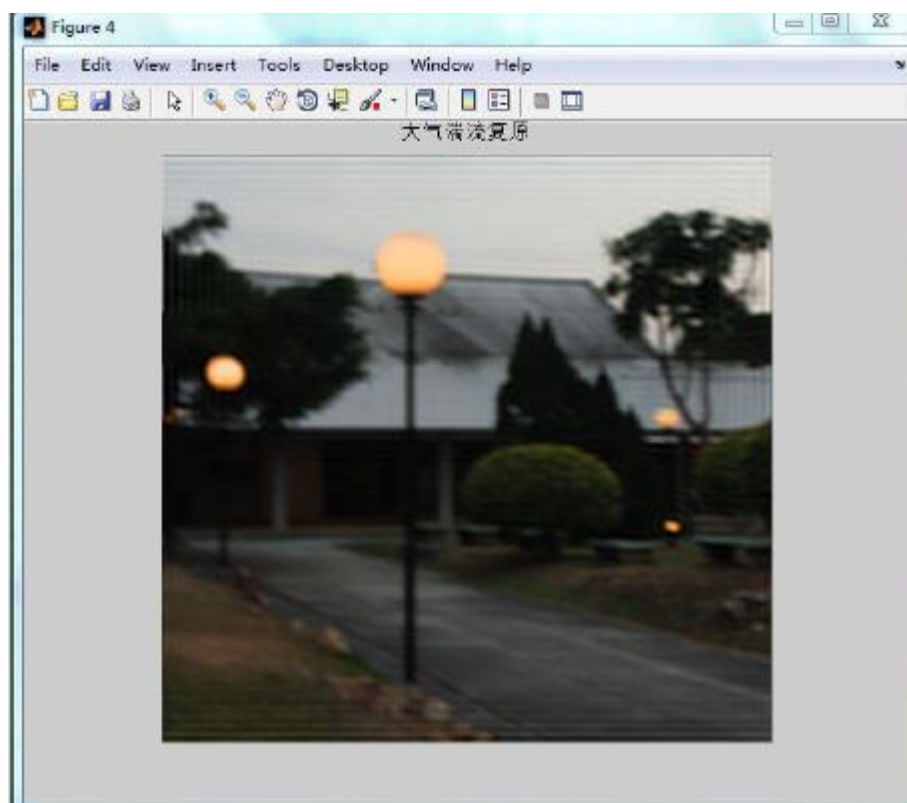
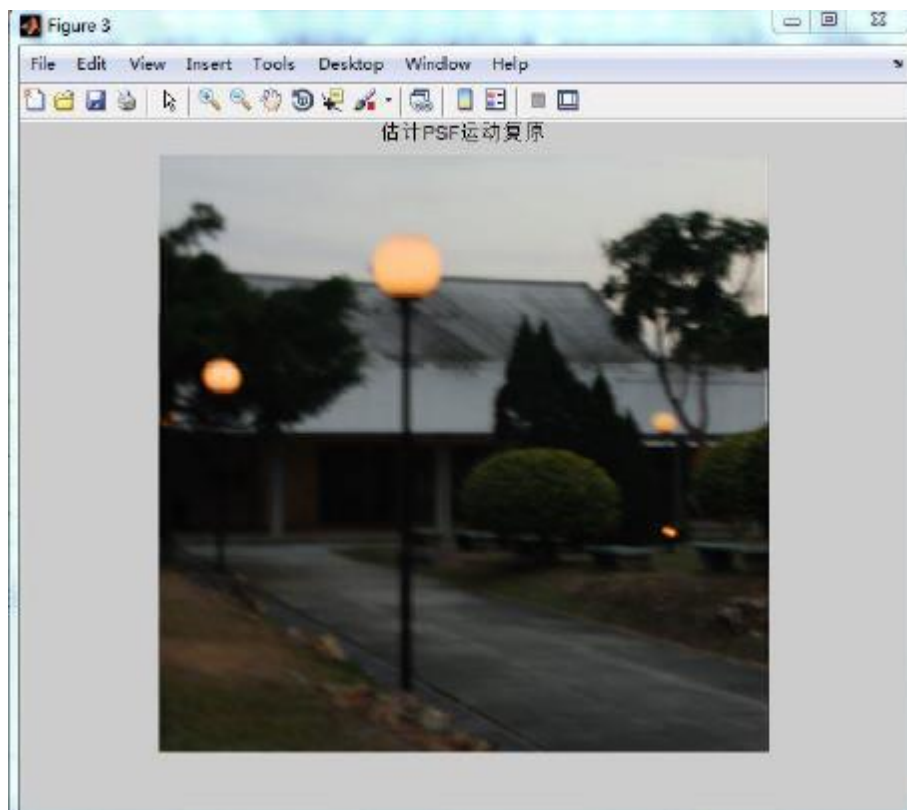
```

im=imread('nvblurImage.png');
figure,imshow(im),title('原始图像');
PSF = fspecial('motion',9,0);
fr1 = double(deconvwnr(im,PSF,0.3));
[m,n,h]=size(fr1);
for i=1:m
for j=1:n
for k=1:3
fr1(i,j,k)=(fr1(i,j,k))^1.06;
end
end
end
figure,imshow(uint8(fr1)),title('估计PSF运动复原');
k=0.000135;
i=fft2(double(im));
G=fftshift(i);
[m, n, h] = size(i);
H=zeros(m,n);
for i=1:m
for j=1:n
if sqrt((i-m/2)^2+(j-n/2)^2)<65

```

```
H(i,j)=exp(-k*(i^2+j^2)^(5/6));  
end  
end  
end  
for i=1:3  
out(:, :, i) = G(:, :, i).*H;  
end  
out = real(ifft2(ifftshift(out)));  
figure,imshow(out,[]),title('大气湍流复原');  
运行结果:
```





分析说明：在原图像中，从路灯的杆子中可以明显看到有“重影”现象，推测是运动模糊或大气湍流模糊，实验中也是对此进行探讨的，本例中所使用的PSF的值由于缺少先验知识，是在实验中不断试出来的效果较为好的PSF，所以和真实的PSF值一定有一些偏差，不过也可以明显看到杆子的重影效果减少了，图像变清晰了一些，经过大气湍流处理的图像中，同样

可以看到杆子及背景的树枝的“重影”明显减少，基本上达到预期的效果。

2.3 汽车复原

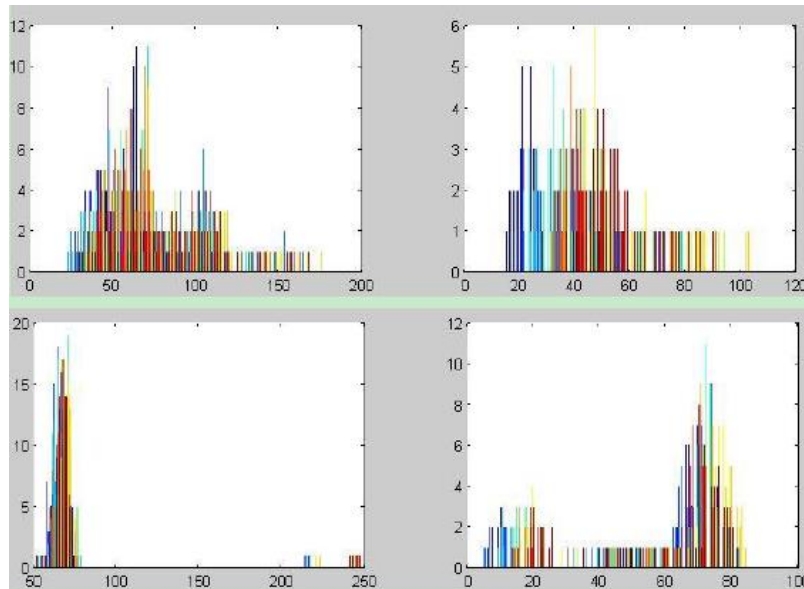
同样也是先分析原始图像中是否有噪声的存在。

代码如下：

```
im = imread('s1.bmp');
[m,n,h]=size(im);
%f1=zeros(m,n);f2=zeros(m,n);f3=zeros(m,n);...
f11=ones(40,30,3);f22=ones(20,16,3);f33=ones(46,31,3);f44=ones(30,27,
3);
for i=61:100
for j=21:50
for k=1:3
f11(i-60,j-20,k)=im(i,j,k);
end
end
end
for i=71:90
for j=121:136
for k=1:3
f22(i-70,j-120,k)=im(i,j,k);
end
end
end
for i=211:256
for j=25:55
for k=1:3
f33(i-210,j-24,k)=im(i,j,k);
end
end
end
for i=201:230
for j=145:171
for k=1:3
f44(i-200,j-144,k)=im(i,j,k);
end
end
end
figure,subplot(221),hist(f11,100),subplot(222),hist(f22,100),subplot(
223),...
hist(f33,100),subplot(224),hist(f44,100);
w1=fspecial('gaussian');
im2=imfilter(im,w1,'conv','replicate');
PSF = fspecial('motion',1.5,40);
```

```
fr1 = deconvwnr(im,PSF);fr2 = deconvwnr(im,PSF,0.3);
figure,subplot(121),imshow(fr1),title('直接估计维纳复原'),subplot(122),...
imshow(fr2),title('高斯去噪后复原');
```

结果如下：

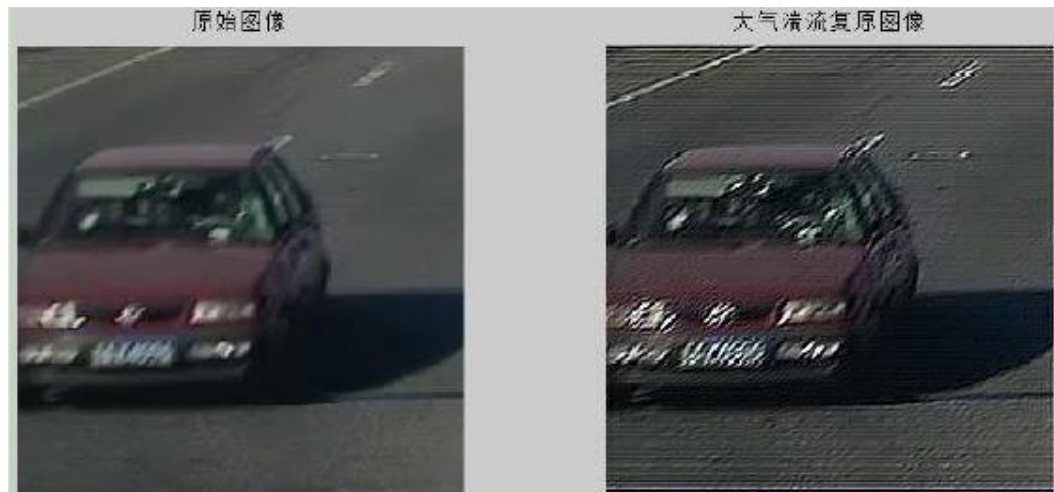


分析说明:可以发现在图中，R、G、B三个分量也基本呈现钟形分布，这是高斯噪声的特点，所以初步判断原始图像中含有高斯噪声，所以对图像进行高斯噪声去除处理后再进行图像还原可以得到较为良好的效果。



分析说明：经过处理后可以基本辨认车牌号为4996，相比较原图像而言清楚了一些，但效果还不是非常理想，这可能是由于还是没有找到精确的模糊过程导致复原时效果很有限。

同理也可以使用大气湍流进行复原处理，运行结果如下：



2.4 毕加索人像复原

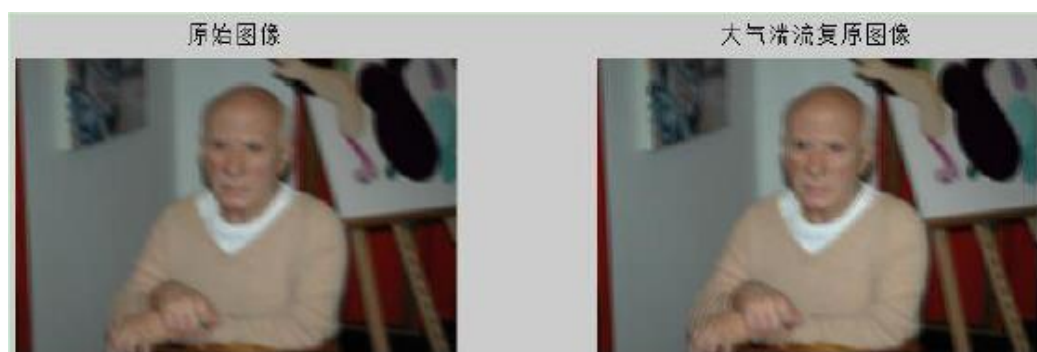
代码如下：

```
image=imread('picassoBlurImage.png');k=0.000185;
i=fft2(double(image));
G=fftshift(i);
[m, n, h] = size(i);
H=zeros(m,n);
for i=1:m
for j=1:n
if sqrt((i-m/2)^2+(j-n/2)^2)<40
H(i,j)=exp(-k*(i^2+j^2)^(5/6));
end
end
end
for i=1:3
out(:,:,i) = G(:,:,i).*H;
end
out = real(ifft2(ifftshift(out)));
figure,subplot(121),imshow(image),title('原始图像'),...
subplot(122),imshow(out,[]),title('频域半径为40复原图像');
PSF = fspecial('motion',1.5,40);
fr1 = deconvwnr(image,PSF);fr2 = deconvwnr(image,PSF,0.3);
figure,subplot(121),imshow(fr1),title('直接估计维纳复原'),...
subplot(122),...
imshow(fr2),title('高斯去噪后复原');
```

运行结果：



同理也可以使用大气湍流进行复原处理，运行结果如下：



五、图象退化和复原(2)

(一) 实验目的

1. 了解图象退化的几种原因;
2. 掌握对相应退化原因的复原方法。

(二) 实验内容

1. 使用函数 `fspecial()` 和 `imfilter()` 模拟产生退化图象;
2. 使用平滑滤波器和维纳滤波器复原图象。

(三) 实验步骤

1. 加性噪声退化图象

- a. 先用 `imnoise()` 函数给图象加噪声, 如对图象增加高斯白噪声

```
x=imread('cameraman.jpg');  
x=imnoise(x,'gaussian')  
imshow(x)
```

- b. 使用平滑滤波器对其进行滤波

```
h=fspecial('average')  
y=imfilter(x,h);  
figure  
imshow(y)
```

- c. 使用维纳滤波器进行处理

```
w=wiener2(x)  
figure  
imshow(w)
```

2. 运动退化图象

- a. 使用 `fspecial()` 和 `imfilter()` 模拟产生运动退化图象

```
clear  
I = imread('flowers.jpg');  
figure;imshow(I);title('Original Image');  
% create PSF  
LEN = 31;  
THETA = 15;  
PSF = fspecial('motion',LEN,THETA);  
% blur the image  
Blurred = imfilter(I,PSF,'circular','conv');  
figure; imshow(Blurred);title('Blurred Image');
```

b.使用维纳滤波器进行处理

```
% deblur the image  
wnr1 = deconvwnr(Blurred,PSF);  
figure;imshow(wnr1);  
title('Restored, True PSF');
```