

# 实验六：图像去噪改进算法

09020334 黄锦峰

2022 年 12 月 6 日

## 1 实验目标

1. 手机夜间拍摄低信噪比静态图像，验证平均次数越多，图像质量或者信噪比越高；
2. 自己添加噪声，在高斯滤波器的基础上，设计去噪效果提高的滤波器（如双边滤波器，non-local means 等，任意选择不少于一种）；分析去噪前后，频谱空间信息变化

## 2 实验内容

### 2.1 任务一：验证平均图像效果

#### 2.1.1 平均静态图像原理去噪原理

多幅图像平均法是对同一景物重复采集  $M$  次相加后取平均值的方法来消除噪声的。图像成像的模型可描述为：

$$g(x, y) = f(x, y) + \eta(x, y) \quad (1)$$

其中  $f(x, y)$  是图像的真实值， $\eta(x, y)$  是噪声， $g(x, y)$  是图像的测量值。  
若噪声是与图像无关的高斯白噪声，即  $E[\eta(x, y)] = 0$ ：

$$E[g(x, y)] = f(x, y) \quad (2)$$

$M$  幅重复采集的图像的平均后输出的图像：

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y) \quad (3)$$

平均后输出的图像的方差为：

$$\sigma_{\bar{g}}^2 = \frac{1}{M} \sigma_{\eta}^2 \quad (4)$$

$\sigma_{\eta}^2$  是原始图像的方差， $\sigma_{\bar{g}}^2$  是平均后输出的图像的方差。

可见平均次数越多，图像质量或者信噪比越高，具有一定的去噪效果。

计算信噪比：

$$SNR_{db} = 10 \log_{10} \frac{P_{signal}}{P_{noise}} \quad (5)$$

其中

$P_{signal} = \sum_{i,j} (f(i, j) - \text{mean}\{f(x, y)\})^2$  是信号功率

$P_{noise} = \sum_{i,j} (f(i, j) - g(i, j))^2$  是噪声功率。

### 2.1.2 平均图像效果验证



图 1: 平均图像效果验证

平均 10 次的图像的信噪比: 8.0715

平均 20 次的图像的信噪比: 9.7764

平均 50 次的图像的信噪比: 11.814

平均 100 次的图像的信噪比: 12.7246

## 2.2 任务二：验证高斯滤波器效果

### 2.2.1 双边滤波器原理

双边滤波（Bilateral filter）是一种可以保边去噪的滤波器。之所以可以达到此去噪效果，是因为滤波器是由两个函数构成。一个函数是由几何空间距离决定滤波器系数。另一个由像素差值决定滤波器系数。

双边滤波器中，输出像素的值依赖于邻域像素的值的加权组合，

$$g(x, y) = \frac{\sum_{k,l} f(k, l) w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)} \quad (6)$$

权重系数  $w(i, j, k, l)$  取决于定义域核:

$$d(i, j, k, l) = e^{\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right)} \quad (7)$$

和值域核

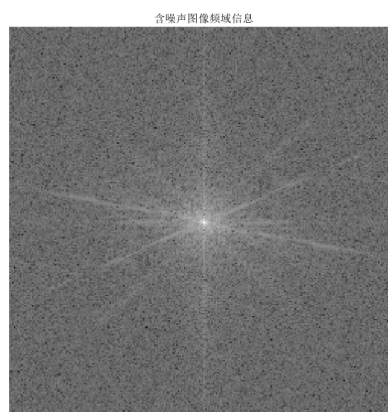
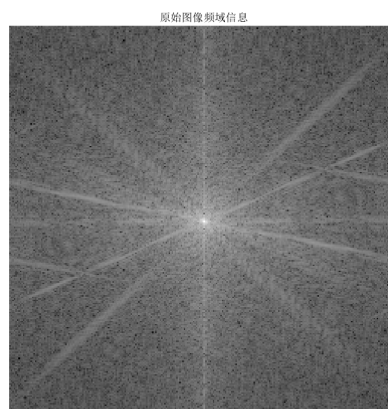
$$r(i, j, k, l) = e^{\left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2}\right)} \quad (8)$$

的乘积

$$w(i, j, k, l) = d(i, j, k, l) r(i, j, k, l) \quad (9)$$

双边滤波器的好处是可以做边缘保存，一般用高斯滤波去降噪，会较明显地模糊边缘，对于高频细节的保护效果并不明显。双边滤波器比高斯滤波多了一个高斯核。它是基于像素颜色分布的高斯滤波函数，所以在边缘附近，当两个像素距离很近时，只有同时当颜色很接近时影响才会较大，反之，虽然距离很近，但颜色差距较大，那么平滑权重也会很小。这样就保证了边缘附近像素值的保持，起到了保护边界的效果。

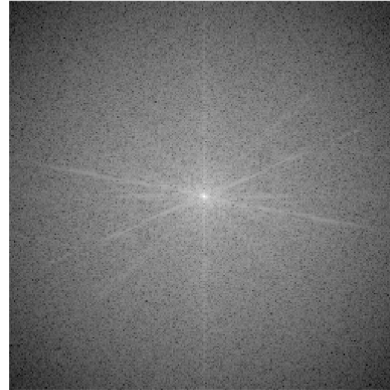
### 2.2.2 滤波器效果验证



高斯低通滤波器处理后图像



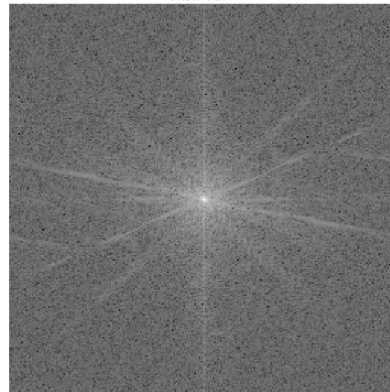
图像频域信息



双边滤波器处理后图像



图像频域信息



观察原图的频域信息，可见图形中的高频信息和低频信息都比较丰富。

通过高斯噪声污染后，图像的高频信息和低频信息都被破坏了，高频信息被破坏的更严重。

使用高斯低通滤波器对污染的图像进行滤波后，图像的高频信息被破坏的更严重，低频信息被破坏的较少。

使用双边滤波器对污染的图像进行滤波后，图像的高频信息被破坏的较少，低频信息被破坏的较少，达到了比较好的效果。

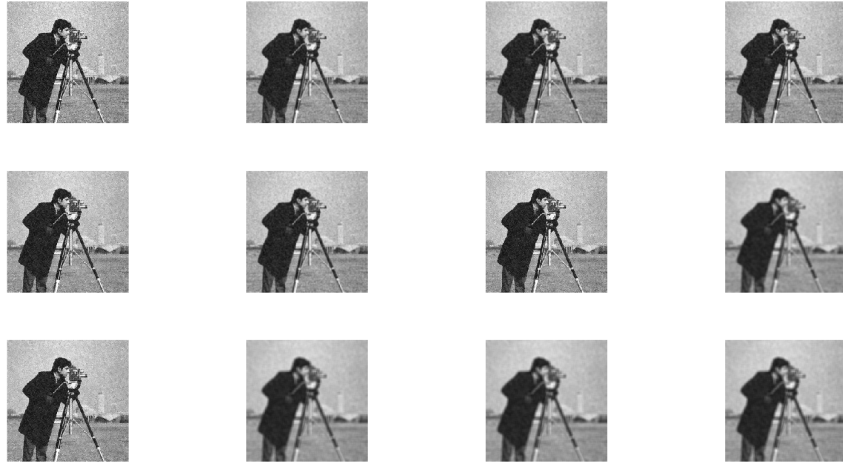


图 2:  $\sigma_d$  和  $\sigma_r$  取不同值的对比图

$\sigma_d$  由上到下依次为: 1、3、10

$\sigma_r$  由左到右依次为: 0.1、10、30、100

- $\sigma_d$  变大, 图像每个区域的权重基本都源于值域滤波的权重, 因此对于空间邻域信息不是很敏感;  $\sigma_r$  变大, 则不太考虑值域, 权重多来自于空间距离, 因此近似于普通的高斯滤波, 图像的保边性能下降。因此如果像更多的去除平滑区域的噪声, 应该提高  $\sigma_d$ , 如果像保持边缘, 则应该减小  $\sigma_r$ 。
- 极端情况, 如果  $\sigma_d$  无穷大, 相当于值域滤波;  $\sigma_r$  无穷大, 相当于空域高斯滤波。

### 3 实验代码

#### 3.1 平均静态图像

Listing 1: Solve\_1.mlx

```
clear
I = imread('cameraman.tif');
[L,H]=size(I); %获取大小
K=zeros(L,H);
n=[10,20,50,100];

figure
for j=1:length(n)
    for i=1:n(j)
        J=imnoise(I,'gaussian',0,0.1); %随机加噪
        J1=im2double(J); %转成double型进行相加
        K=K+J1;
    end
    K=K/n(j);
```

```

        subplot(2,3,(j+2))
        imshow(K),title(['平均',num2str(n(j)),'次的图像']);

        disp(['平均',num2str(n(j)),'次的图像的信噪比: ',num2str(SNR1(im2double(I),K)
        ]))
    end
    subplot(2,3,1),imshow(I),title('原图像');
    subplot(2,3,2),imshow(J),title('加噪声后图像');

    function snr=SNR1(I,I_noise)
    % 计算信噪比
    Ps=sum(sum((I-mean(mean(I)).^2)));
    Pn=sum(sum((I-I_noise).^2));
    snr=10*log10(Ps/Pn);
    end

```

## 3.2 滤波器效果验证

Listing 2: Solve\_2.mlx

```

clear
I = imread('cameraman.tif');

figure,
subplot(121),imshow(I);title('原图像');
f_I=fftshift(fft2(I));      %原图像频域信息
subplot(122),imshow(log(abs(f_I)),[]);title("原始图像频域信息")

R=imnoise(I,'gaussian',0.1);
figure
subplot(121),imshow(R);title('加高斯噪声后图像');
f_R=fftshift(fft2(R));      %原图像频域信息
subplot(122),imshow(log(abs(f_R)),[]);title("含噪声图像频域信息")

% 普通的高斯低通滤波器进行处理
[H,g]=lpfilter('gaussian',50,1,R);
figure
subplot(121),imshow(g,[]);title('高斯低通滤波器处理后图像')
f_g=fftshift(fft2(g));      %原图像频域信息
subplot(122),imshow(log(abs(f_g)),[]);title("图像频域信息")

% 双边滤波器处理图像
g2=im2double(R);
res= bilateral(g2,3,3,0.1);
figure
subplot(121),imshow(res);title('双边滤波器处理后图像')
f_res=fftshift(fft2(res));  %原图像频域信息
subplot(122),imshow(log(abs(f_res)),[]);title("图像频域信息")

```

```

figure
res1= bilateral(g2,3,1,0.1);
subplot(3,4,1);imshow(res1);
res2= bilateral(g2,3,1,10);
subplot(3,4,2);imshow(res2);
res3= bilateral(g2,3,1,30);
subplot(3,4,3);imshow(res3);
res4= bilateral(g2,3,1,100);
subplot(3,4,4);imshow(res4);

res5= bilateral(g2,3,3,0.1);
subplot(3,4,5);imshow(res5);
res6= bilateral(g2,3,3,10);
subplot(3,4,6);imshow(res4);
res7= bilateral(g2,3,3,30);
subplot(3,4,7);imshow(res5);
res8= bilateral(g2,3,3,100);
subplot(3,4,8);imshow(res6);

res9= bilateral(g2,3,10,0.1);
subplot(3,4,9);imshow(res9);
res10= bilateral(g2,3,10,10);
subplot(3,4,10);imshow(res10);
res11= bilateral(g2,3,10,30);
subplot(3,4,11);imshow(res11);
res12= bilateral(g2,3,10,100);
subplot(3,4,12);imshow(res12);

```

### 3.3 低通滤波器

Listing 3: lpfilter.m

```

function [H,g] = lpfilter(filtertype,D0,n,image)
%LPFILTER 此处显示有关此函数的摘要
% filtertype--滤波器的类型
% n--滤波器的阶数
% D0--截止频率
% image--原始图像
% H--滤波器
% g--滤波后的图像

%im=imread(image);
im=image;
[DIM1,DIM2,K]=size(im);%确定原始图像的大小
if K==3
    im1=rgb2hsv(im);
    im2=double(im1(:,:,3));
else
    im2=double(im);

```

```

end

u=0:(DIM1-1);
v=0:(DIM2-1);

idx=find(u>DIM1/2);
u(idx)=u(idx)-DIM1;
idy=find(v>DIM2/2);
v(idy)=v(idy)-DIM2;
[V,U]=meshgrid(v,u);
D=sqrt(U.^2+V.^2);

switch filtertype
    case 'ideal'
        H=double(D<=D0);
    case 'butterworth'
        H=1./(1+(D./D0).^(2*n));
    case 'gaussian'
        H=exp(-(D.^2)./(2*(D0^2)));
    case 'exponent'
        H=exp(-D./(2*D0)./(2*D0));
    case 'trapeziod'
        if D<=D0
            H=1.0;
        elseif D>100
            H=0.0;
        else
            H=(D-100)./(D0-100);
        end
    otherwise
        error('Unkown filter type')
end

F=(fft2(im2));
G=H.*F;
g=real(ifft2(G));
% figure
% subplot(121);imshow(im2,[]);title('Original Image')
% subplot(122);imshow(g,[]);title('Lowpass Filtered Image')
end

```

### 3.4 双边滤波器

Listing 4: bilateral.m

```

function B = bilateral(img,w,sigma_d,sigma_r)
%输出参数:
% img为待滤波图像(double类型,取值在[0,1])
% w为滤波窗口的半径(e.g:3*3窗口的w值为1,w=3时的滤波效果较好)

```



```

% sigma_d为定义域（空间域）核的方差，通常设置为3
% sigma_r为值域核的方差，通常设置为0.1
%输出参数：
% B为滤波后的图像
% 预先计算高斯距离权重
[X,Y] = meshgrid(-w:w,-w:w);

%计算定义域核
G = exp(-(X.^2+Y.^2)/(2*sigma_d^2));

%计算值域核H 并与定义域核G 乘积得到双边权重函数F
dim = size(img);
B = zeros(dim);
for i = 1:dim(1)
    for j = 1:dim(2)
        % 确定作用区域
        iMin = max(i-w,1);
        iMax = min(i+w,dim(1));
        jMin = max(j-w,1);
        jMax = min(j+w,dim(2));
        %定义当前核所作用的区域为(iMin:iMax,jMin:jMax)
        I = img(iMin:iMax,jMin:jMax);%提取该区域的源图像值赋给I

        %计算值域核H.
        H = exp(-(I-img(i,j)).^2/(2*sigma_r^2));

        % Calculate bilateral filter response.
        F = H.*G((iMin:iMax)-i+w+1,(jMin:jMax)-j+w+1);
        %在计算边缘部分的点的时候H的大小会变化，例如在计算第一行第一列的点时，
        %H的大小为4*4，因为7*7的其余部分都在图像外面（没有对应的值）。
        %因此适当的对G进行裁切使得G的大小始终能和H对上
        B(i,j) = sum(F(:).*I(:))/sum(F(:));
    end
end
end
end

```