# 编程作业 1 简明报告

Haoyu Zhen

2022 年 2 月 18 日

## 1 添加模糊及噪音

### 1.1 模糊化

利用卷积运算即可制造模糊，

```
% Blurred image
PSF = ones([5,5])*0.04;
% blurred = conv2(origin,PSF);
% size = 516;
blurred = conv2(origin,PSF,'same');
size = 512;
```

效果如图 (2) 所示。值得说明的是，我传入了 'same' 这个参数，以使得前后图片大小一致，但这会导致边缘信息的丢失。因此在恢复图像时，图片边缘会出现 “方格” 现象，如图 (5) 所示，若不传入 'same' 参数，后续的逆滤波会取得更好的效果。

### 1.2 噪音化

使用 awgn 函数可直接为图片添加噪音，对应代码如下：

```
% Blurred and noised image
bn30 = awgn(blurred,30,'measured');
bn20 = awgn(blurred,20,'measured');
bn10 = awgn(blurred,10,'measured');
```

效果如图 (3) 所示。

## 2 图像复原

### 2.1 直接逆滤波

在编程上可以用两种方式实现，基于定义，利用傅立叶变换进行除法；或者利用 deconvwnr 函数，并将 nsr 设为 0，(默认值)。我选择了后者作为实现方式，前者在源代码中以注释形式给出：

```
% Implement filtering directly
% F_SPF = fft2(PSF,size,size);
% r1 = ifft2(fft2(bn10)./F_SPF);
% r2 = ifft2(fft2(bn20)./F_SPF);
% r3 = ifft2(fft2(bn30)./F_SPF);
% For simplicity, we could use ...
    deconvwnr(bn, PSF)
df1 = deconvwnr(bn10, PSF);
df2 = deconvwnr(bn20, PSF);
df3 = deconvwnr(bn30, PSF);
```

效果如图 (4) 所示。

### 2.2 Wiener 滤波

使用 MATLAB 中 deconvwnr 函数可一定程度的去除噪音与模糊，代码如下：

```
% Implement wiener filtering
wnr1 = deconvwnr(bn10, PSF, 0.1);
wnr2 = deconvwnr(bn20, PSF, 0.01);
wnr3 = deconvwnr(bn30, PSF, 0.001);
```

效果如图 (5) 所示。

### 2.3 伪-逆滤波

“Image Restoration.pdf” 第 29 页给出了 pseudo-inverse filtering 方法，实现如下，不过源代码中被我所注释并附在于最后。

```
% implement pseudo-inverse filtering
% Here \epsilon = 0.5
F_PSF = fft2(PSF,size,size);
H = (abs(F_PSF)>0.5)./F_PSF;
F_bn30 = fft2(bn30);
F_bn20 = fft2(bn20);
F_bn10 = fft2(bn10);
rr1 = ifft2(F_bn10.*H);
rr2 = ifft2(F_bn20.*H);
rr3 = ifft2(F_bn30.*H);
```

# Appendix

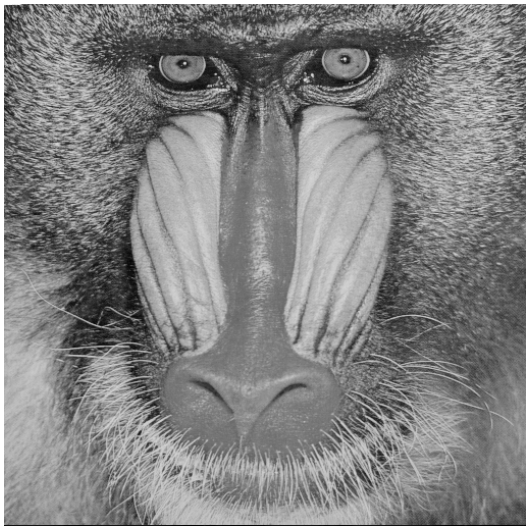
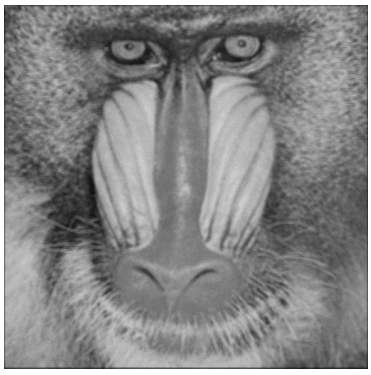
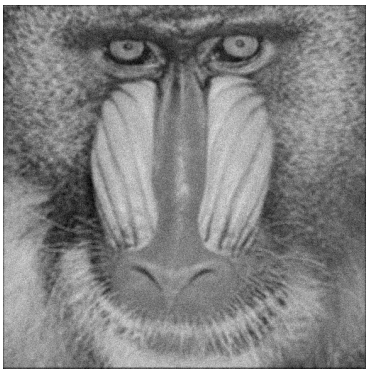


图 1: Original image

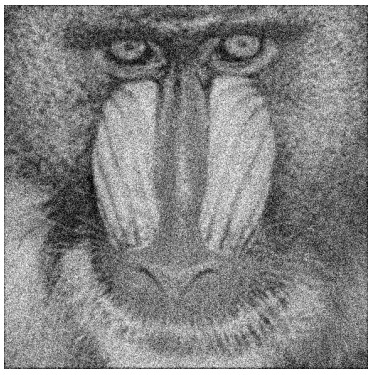
图 2: blurred image



30dB

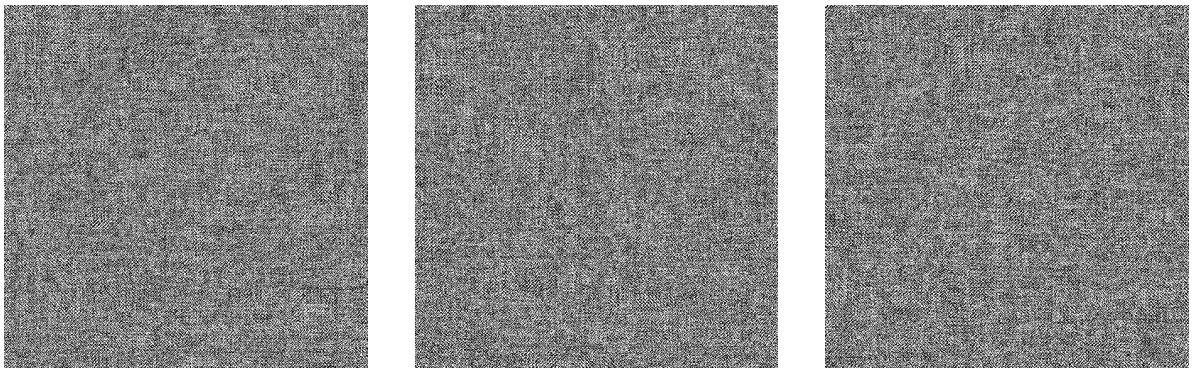


20dB



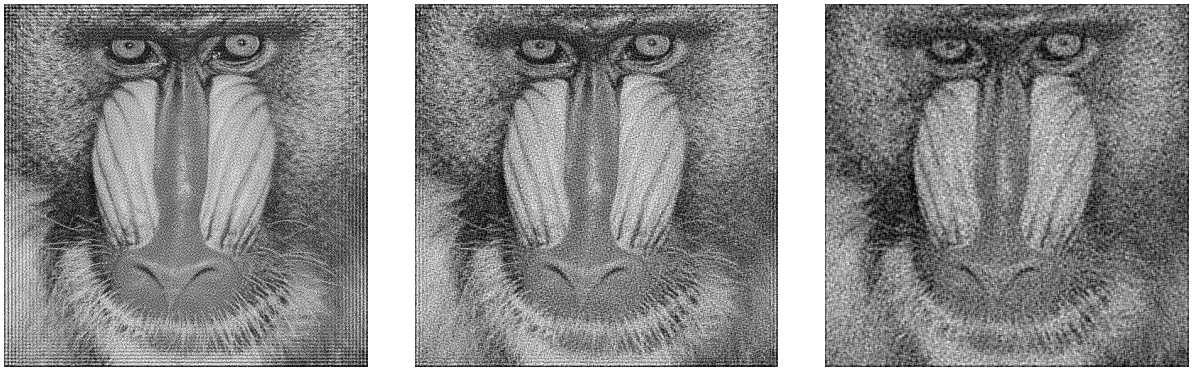
10dB

图 3: Blurred and noised image



Restoration-30dB Restoration-20dB Restoration-10dB

图 4: Naive inverse filtering



Restoration-30dB Restoration-20dB Restoration-10dB

图 5: Wiener filtering