

不同高斯噪声进行不同方式维纳滤波复原的峰值信噪比（PSNR）如下表

峰值信噪比	PSNR		
噪声类型	高斯噪声		
噪声标准差	0.1	0.01	0.001
常数比率的维纳滤波	30.2579	29.5927	28.3471
自相关函数的维纳滤波	39.4152	36.7114	32.5348

问题：应该加的标准差效果越大，它复原效果比较低，PSNR 应该越小才对，我现在的是反的为什么？

答：首先解释一下 PSNR

它说的是处理图像与原图像的失真情况的对比，**PSNR 越大，说明实则失真越小，二者越接近**。但是有一点，它是一个客观测量法，许多实验结果都显示，PSNR 的分数**无法**和人眼看到的视觉品质**完全一致**，有可能 PSNR 较高者看起来反而比 PSNR 较低者差。这是因为人眼的视觉对于误差的敏感度并不是绝对的，其感知结果会受到许多因素的影响而产生变化。

然后，大家说我可能粘错了，但是，我又运行了一边，结果就是这样，我改变了一下：我将高斯标准差选定为 0.05,0.1, 0.5，重新试了一下，结果符合 PSNR 越大，效果越好。

峰值信噪比	PSNR		
噪声类型	高斯噪声		
噪声标准差	0.5	0.1	0.05
常数比率的维纳滤波	29.7198	30.2579	30.4115
自相关函数的维纳滤波	32.4982	39.4152	39.8784



上面图是常数比率的维纳滤波，高斯噪声标准差分别是 0.5,.0.1,0.05



上面图是常数比率的维纳滤波，高斯噪声标准差分别是 0.5,.0.1,0.05，
所以，我的程序没有错，造成问题的原因，

- (1)我原先加的高斯标准差是 0.001, 0.01,0.1, 跳度太大, 0.001 太小了。
(2)0.001, 0.01 会有边缘效果, 虽然纹理清晰了, 但是还是影响质量评价的。

2. 我想说一下, 今天下午大家说的为啥我用的和你们不一样, 我看了看, 大家用的都是 `wiener2` 这个函数, 我用的是 `deconvwnr` 这个函数。我这个是维纳滤波的反卷积, 这个函数的用法就是我今天下午讲的公式, 针对退化模型进行反卷积。Winner2 和 `deconvwnr` 都能完成维纳滤波的功能, `deconvwnr` 强调的是图像复原, Winner2 强调的是图像空间域锐化的作用。因为, 我觉得这个题目用 `deconvwnr` 比较好吧。

下面是我的代码, 供大家参考, 可以与大家做的对比一下, 它的思路, 就是今天下午我讲的 (难点就是那个噪声表示) :

```
I = imread('lena.jpg');%原图像
I=rgb2gray(I);
I=im2double(I);
PSF = fspecial('gaussian',7,10);%高斯低通滤波器
Blurred = imfilter(I,PSF);%产生模糊的图像
noise=imnoise(zeros(size(I)), 'gaussian',0,0.1);%噪声图像
BlurredNoisy = im2uint8(Blurred + noise);%模糊的图像加上噪声
NSR = sum(noise(:).^2)/sum(I(:).^2);%噪声信号功率比
NP = abs(fftn(noise)).^2;
NPOW = sum(NP(:))/prod(size(noise));%噪声功率谱
NCORR = fftshift(real(ifftn(NP))); %噪声自相关函数
IP = abs(fftn(I)).^2;%
IPOW = sum(IP(:))/prod(size(I));%原图像功率谱
ICORR = fftshift(real(ifftn(IP)));%原图像相关函数
ICORR1 = ICORR(:,ceil(size(I,1)/2));
NSR = NPOW/IPOW;%噪声信号功率比
J1=deconvwnr(BlurredNoisy,PSF);%依次改变各个参数进行维纳滤波
J2=deconvwnr(BlurredNoisy,PSF,NSR);
J3=deconvwnr(BlurredNoisy,PSF,NCORR,ICORR);
%J4=deconvwnr(BlurredNoisy,PSF,NPOW,ICORR1);
figure;%显示原图像, 退化的图像和维纳滤波后的图像
subplot(321), imshow(I); title('original image');
imwrite(I, 'original.jpg')
imwrite(Blurred, 'Blurred.jpg')
subplot(322); imshow(BlurredNoisy, []);
imwrite(BlurredNoisy, 'BlurredNoisy0.1.jpg')
title('A = Blurred and Noisy');
subplot(323), imshow(J1, []); title('deconvwnr(BlurredNoisy,PSF)');
imwrite(J1, '直接逆滤波 0.1.jpg')
subplot(324);imshow(J2, []); title('deconvwnr(A,PSF,NSR)');
imwrite(J2, '常数比率的维纳滤波 0.1.jpg')
subplot(325);imshow(J3, []); title('deconvwnr(A,PSF,NCORR,ICORR)');
```

```
imwrite(J3, '使用自相关函数的维纳滤波 0.1.jpg')  
%subplot(326);imshow(J4, []);  
title('deconvwnr(A,PSF,NPOW,ICORR_1_D)');
```