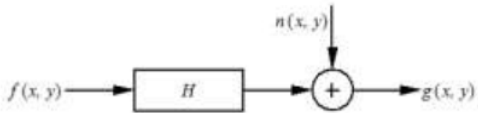


贵州大学实验报告

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	实验四 数字图像复原处理				
实验目的	1. 通过本实验的学习使学生熟悉和掌握数字图像复原中的逆滤波和维纳滤波的原理及应用。				
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	<p>1、 图像退化模型</p> <p>空域：</p> $g(x,y) = H[f(x,y)] + n(x,y)$  <p>2、 逆滤波原理</p> <p>频域：</p> $G(u,v) = F(u,v)H(u,v) + N(u,v)$ $\hat{I}(u,v) = J(u,v) / H(u,v) \quad u,v=0,1,\dots,N-1$ $\hat{f}(x,y) = \mathcal{F}^{-1} \left\{ \frac{G(u,v) - N(u,v)}{H(u,v)} \right\} = \mathcal{F}^{-1} \left\{ F(u,v) - \frac{N(u,v)}{H(u,v)} \right\}$ <p>3、 维纳滤波</p> <p>复原公式：</p>				

	$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{ H(u,v) ^2 + S_n(u,v)/S_f(u,v)} \right] G(u,v)$ $= \left[\frac{1}{H(u,v)} \frac{ H(u,v) ^2}{ H(u,v) ^2 + S_n(u,v)/S_f(u,v)} \right] G(u,v)$ <p>如果噪声为 0，则维纳滤波退化为逆滤波。</p> <p>如果噪声功率谱和信号功率谱都是未知或不能估计时可近似为：</p> $\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{ H(u,v) ^2}{ H(u,v) ^2 + K} \right] G(u,v)$
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. 编写逆滤波复原程序。读取图像，人工产生一个模糊图像，通过逆滤波方法对该模糊复原。人工产生一个模糊且带有噪声的图像，过逆滤波方法对该模糊复原。比较无噪声和有噪声时逆滤波的复原效果。计算复原后图像的 PSNR； 2. 编写维纳滤波复原程序。人工产生一个模糊且带有噪声的图像，利用维纳滤波方法对该模糊复原。计算复原后图像的 PSNR，与逆滤波的复原效果比较。
实验内容	1. 利用逆滤波对图像进行复原； 2. 利用维纳滤波对图像进行复原。
实验数据	<p>Initialization</p> <pre>close all clear clc im = imread(' ../data/jingjing.jpg');</pre>

```
imGray = rgb2gray(im);  
  
[rows, cols] = size(imGray);
```

Inverse Filtering

```
figure  
  
subplot(2,4,1)  
  
imshow(imGray)  
  
title('Original Image')  
  
  
% gaussian low-pass  
  
imBlurred = imgaussfilt(imGray, 2);  
  
subplot(2,4,2)  
  
imshow(imBlurred, [])  
  
title('Gaussian Blurred Image')  
  
  
  
imNoise = imnoise(imBlurred, 'salt & pepper', 0.02);  
  
subplot(2,4,3)  
  
imshow(imNoise, [])  
  
title('Salt & Pepper Noise Image')  
  
  
  
Y = fft2(imNoise);  
  
subplot(2,4,4)  
  
imshow(10.*log10(abs(fftshift(Y))+1), [])  
  
title('Y(u,v)')  
  
  
  
H = fft2(imBlurred)./fft2(imGray);  
  
subplot(2,4,5)  
  
imshow(10.*log10(abs(fftshift(H))+1), [])
```

```

title('H(u,v)')

N = fft2(imNoise - imBlurred);

subplot(2,4,6)

imshow(10.*log10(abs(fftshift(N))+1), [])

title('N(u,v)')

X = fft2(imGray);

subplot(2,4,7)

imshow(10.*log10(abs(fftshift(X))+1), [])

title('X(u,v)')

x = ifft2( X - N./(H+1) );

subplot(2,4,8)

imshow(abs(x), [])

title('Inverse Filtering Result')

fprintf('PSNR without noise: \n\t%f, \nwith noise: \n\t%f\n',...

        psnr(imBlurred, imGray), psnr(imNoise, imGray))

fprintf('Inverse Filter PSNR without noise: \n\t%f, \nwith noise: \n\t%f\n',...

        psnr(uint8(ifft2(fft2(imBlurred)./H)), imGray), psnr(uint8(x), imGray))

```

PSNR without noise:

26.555532,

with noise:

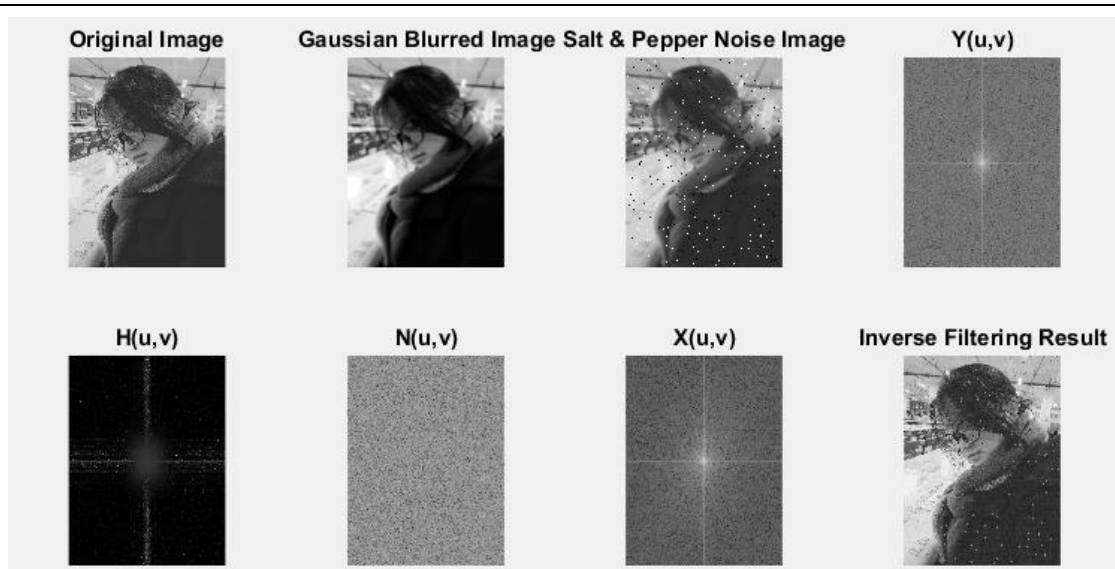
20.665432

Inverse Filter PSNR without noise:

Inf,

with noise:

29.955409



Wiener Filtering

from snr function, we could know that $P = \text{rssq}^2(x)$ ----- % function re =
 signalPower(in) re = rssq(in(:))^2; end ----- %

```

Sn = signalPower(imGray);

Sf = signalPower(imNoise - imBlurred);

F = (1./H .* ( abs(H).^2./( abs(H).^2 + 0.11 ) )) .* Y;

% F = (1./H .* ( abs(H).^2./( abs(H).^2 + Sn/Sf ) )) .* Y;

f = ifft2(F);

figure

subplot(2,2,1)

imshow(imGray, [])

title('Original Image')

subplot(2,2,2)

imshow(imNoise, [])

title('Noise Image')

subplot(2,2,3)

imshow(10.*log10(abs(fftshift(F))+1), [])

title('Wiener Filter F-Frequency Domain')

```

```
subplot(2,2,4)

imshow(f, [])

title('Wiener Filter Re-construction')


fprintf('PSNR without noise: \n\t%f, \nwith noise: \n\t%f\n',...
        psnr(imBlurred, imGray), psnr(imNoise, imGray))

fprintf('Wiener Filter PSNR without noise: \n\t%f, \nwith noise: \n\t%f\n',...
        psnr(uint8(iff2(fft2(imBlurred)./H)), imGray), psnr(uint8(f), imGray))
```

PSNR without noise:

26.555532,

with noise:



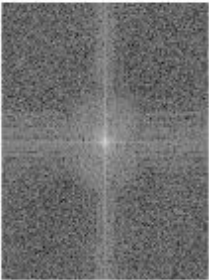

20.665432

Wiener Filter PSNR without noise:

Inf,

with noise:

21.072115

	<div><div><div>Original Image</div></div><div><div>Noise Image</div></div></div> <div><div>Wiener Filter F-Frequency Domain</div></div> <div><div>Wiener Filter Re-construction</div></div>
--	---