

# 数字图像处理第六次作业

姓名：边策

班级：自动化 63

学号：2160504062

摘要：本次项目主要实现了退化图像实现及其复原的几种算法。图像退化包含退化函数和加性噪声两部分。对于污染图像噪声参数未知情况下，我们可以对噪声参数进行估计。如果只存在加性噪声，则可以采用空间滤波的方法进行处理。在退化函数和加性噪声同时存在的情况下，我们可以采取维纳滤波和约束最小二乘方滤波的方法

# 报告正文

## 问题讨论

### 1. 第一问

在测试图像上产生高斯噪声lena图-需能指定均值和方差；并用多种滤波器恢复图像，分析各自优缺点；增加噪声我们可以通过

`fn=imnoise(f,'gaussian',0,0.01)`来对图像f进行增加高斯噪声得到被噪声污染的图像fn。我们通过3\*3算术均值滤波，3\*3几何均值滤波，3\*3中值滤波，对被污染的图像进行空域卷积滤波，和第三次一样调用`imfilter`函数就可以对污染图像fn滤波。根据matlab版数字信号处理，我们得到几何均值滤波的函数。

```
function f=gmean(g,m,n)
g=single(g);
f=exp(imfilter(log(g),ones(m,n),'replicate')).^(1/m/n);
f=uint8(f);
end
```

### 2. 第二问

在测试图像lena图加入椒盐噪声（椒和盐噪声密度均是0.1）；用学过的滤波器恢复图像；在使用反谐波分析Q大于0和小于0的作用；

我们加入椒盐噪声`fn=imnoise(f,'salt & pepper',0.1)`  
同第一问我们开始时先使用上述的三种滤波方式，然后我们通过编写逆调和均值滤波，通过改变Q观察对椒盐噪声的影响

```
function f=charmean(g,m,n,q)
g=single(g);
f=imfilter(g.^(q+1),ones(m,n),'replicate');
f=f./(imfilter(g.^q,ones(m,n),'replicate')+eps);
f=uint8(f);
end
```

逆调和均值滤波如上。

### 3. 第三问

#### 1. 推倒维纳滤波器

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$\hat{f}(x, y) = w(x, y) * g(x, y)$$

$$\text{误差 } e = \hat{f}(x, y) - f(x, y) \quad \text{使 } E\|e\| \text{ 最小.}$$

由正交原理知

$$E\{e(n_1, n_2) g^*(m_1, m_2)\} = 0$$

$$\text{进一步分解} = E\{f(n_1, n_2) g^*(m_1, m_2)\} = E\{\hat{f}(n_1, n_2) g^*(m_1, m_2)\}$$

$$= E\{w(n_1, n_2) * g(n_1, n_2) g^*(m_1, m_2)\}$$

$$= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} w(k_1, k_2) E\{g(n_1 - k_1, n_2 - k_2) g^*(m_1, m_2)\}$$

$$\textcircled{1} P_g(n_1, n_2) = E\{g(k_1, k_2) g^*(k_1 - n_1, k_2 - n_2)\}$$

$$\text{则上式可得 } P_{fg}(n_1 - m_1, n_2 - m_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} w(k_1, k_2) P_g(n_1 - k_1, n_2 - k_2 - m_2)$$

$$\Rightarrow P_{fg}(n_1, n_2) = W(n_1, n_2) * P_g(n_1, n_2).$$

$$W(w_1, w_2) = \frac{P_{fg}(w_1, w_2)}{P_g(w_1, w_2)}$$

$$\textcircled{2} \text{求 } P_{fg} \quad \cancel{P_{fg}(n_1, n_2)} = P_{fg}(w_1, w_2) = \sum_{n_1} \sum_{n_2} P_{fg}(n_1, n_2) e^{-j(w_1 n_1 + w_2 n_2)}$$

因为  $f$  与  $\eta$  不相关

$$\cancel{E\{f(n_1, n_2) \eta^*(m_1, m_2)\}} = P_{fg} = E\{f(n_1 + l_1, n_2 + l_2) g^*(l_1, l_2)\}$$

$$= E\{f(n_1 + l_1, n_2 + l_2) [h^*(l_1, l_2) * f^*(l_1, l_2) + \eta^*(l_1, l_2)]\}$$

$$\Rightarrow P_{fg} = h^*(w_1, w_2) P_f(w_1, w_2) \quad ? \text{ 怎么来的}$$

$$A(x_1, x_2) \cdot A^*(y_1, y_2) = \Gamma_A(x_1 - y_1, x_2 - y_2)$$

NO.

DATE.

$$r_g(n_1, n_2) = E \{ g(n_1, n_2) \cdot g^*(l_1, l_2) \}$$

$$= E \{ [h(n_1, n_2) * f(n_1, n_2) + \eta(n_1, n_2)] \cdot [h^*(l_1, l_2) * f^*(l_1, l_2) + \eta^*(l_1, l_2)] \}$$

$$= E \{ \sum_{m_1, m_2} h(m_1, m_2) f(n_1 + l_1 - m_1, n_2 + l_2 - m_2) \cdot \sum_{k_1, k_2} h^*(k_1, k_2) f^*(l_1 - k_1, l_2 - k_2) + \eta(n_1, n_2) \cdot \eta^*(l_1, l_2) \}$$

$$= E \{ \sum_{m_1, m_2} \sum_{k_1, k_2} h(m_1, m_2) f(n_1 + l_1 - m_1, n_2 + l_2 - m_2) h^*(k_1, k_2) f^*(l_1 - k_1, l_2 - k_2) + r_g(n_1, n_2) \}$$

$$= E \{ \sum_{m_1, m_2} \sum_{k_1, k_2} h(m_1, m_2) h^*(k_1, k_2) r_f(n_1 - m_1 + k_1, n_2 - m_2 + k_2) + r_g(n_1, n_2) \}$$

$$= E \{ \sum_{k_1, k_2} h^*(k_1, k_2) \underbrace{h(n_1 + l_1, n_2 + l_2) * r_f(n_1 + k_1, n_2 + k_2)}_{R(n_1 + k_1, n_2 + k_2)} + r_g(n_1, n_2) \}$$

$$= E \{ h^*(l_1 - n_1, l_2 - n_2) * h(n_1, n_2) * r_f(n_1, n_2) + r_g(n_1, n_2) \}$$

在左右变换

$$\Rightarrow P_g(w_1, w_2) = |H^*(w_1, w_2) H(w_1, w_2) - P_f(w_1, w_2) + P_g(w_1, w_2)|$$

$$W = \frac{P_f}{P_g + P_f} = \frac{|H^*(w_1, w_2) H(w_1, w_2) - P_f(w_1, w_2)|}{|H^*(w_1, w_2) H(w_1, w_2) - P_f(w_1, w_2) + P_g(w_1, w_2)|}$$

$$= \frac{||H^*(w_1, w_2) H(w_1, w_2)||^2}{||H^*(w_1, w_2) H(w_1, w_2)||^2 + \frac{P_g}{P_f}} = k$$

2. 首先对图像进行模糊，根据书中公式

$$H(u, v) = \frac{1}{\pi(u^2 + v^2)} \sin(\pi(u^2 + v^2)) e^{-j\pi(u^2 + v^2)}$$

通过调用matlab自带的fspecial('motion', len, theta)函数之后增加高斯噪声，得到被污染的图像。利用维纳滤波器，维纳滤波器回复图像表达式如下

$$F(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

## 结果及分析

### 1. 第一问

原图



高斯噪声



3\*3算术均值滤波



3\*3几何均值滤波



3\*3中值滤波



通过上述结果我们可以看出算术均值滤波器平滑一幅图像的局部变化,虽然模糊了结果,但降低了噪声。



几何均值滤波器对图像的平滑效果一样,但这种处理中丢失的图像细节更少。效果上更好一些。

中值滤波并没有很好的消除高斯噪声,但也没有使图像变模糊

## 2. 第二问

在这一问中,我们用了逆调和均值滤波,针对不同的 $Q$ 得到结果如下:

原图



椒盐噪声



3\*3算术均值滤波



3\*3几何均值滤波



3\*3中值滤波



3\*3逆调和值 $Q=-1$ 滤波



3\*3逆调和值 $Q=-0.5$ 滤波3\*3逆调和值 $Q=0$ 滤波3\*3逆调和值 $Q=0.5$ 滤波3\*3逆调和值 $Q=1$ 滤波

通过观察我们发现对于逆谐波均值滤波器来说，不能同时消除两种噪声，当  $Q$  值为正时，滤波器能够消除胡椒噪声；当  $Q$  值为负时，该滤波器能够消除盐粒噪声。当  $Q=0$  时，此滤波器可以简化为算术均值滤波器，因为本次实验并没有设置谐波均值滤波器，但是当  $Q=-1$  时对盐粒噪声消除较好。 $|Q|$  的值越大对其中一种噪声滤波效果越好，但对另一种噪声的滤波效果越差。相反，我们发现对空域进行非线性滤波即中值滤波后，效果极好

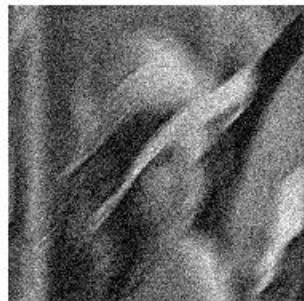
### 3. 第三问



原始图像



模糊图像



滤波复原图像 $k=0.01$



滤波复原图像 $k=0.03$



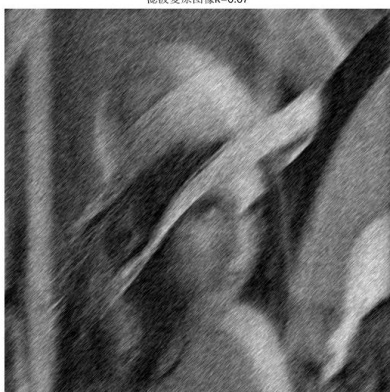
滤波复原图像 $k=0.05$



滤波复原图像 $k=0.11$



滤波复原图像 $k=0.07$



滤波复原图像 $k=0.09$



滤波复原图像 $k=0.13$



滤波复原图像 $k=0.15$





通过观察我们可以发现通过调节  $k$  的值对图像的恢复效果是不同的，当  $k$  值较小时对退化的效果回复较好，当  $k$  值较大时，对噪声的恢复效果较好，在实际应用中应该估算好噪声的能量与原图像的信噪比，进行带入  $k$  才能得到更好的结果。

## 附录

### 参考文献:

1. 数字图像处理: MATLAB 版: 本科教学版/ (美) Gonzalez,R.C, woods, R.E, Eddins, S.L. 著; 阮秋琦译.—2 版.—北京: 电子工业出版社, 2014.1
2. 数字图像处理: 第三版/ (美) 拉斐尔·冈萨雷斯, (美) 理查德·伍兹著; 阮秋琦等译.—北京: 电子工业出版社, 2017.5

### 代码

#### Six1.m

```
clear;
[f,map_f]=imread('C:\Users\15pr\Desktop\ÐÂ½"îÄ¼p¼Ð
(2)\µÚ6'î×÷Òµ\lena.bmp');
figure;
imshow(f);
title('Ô-í¼')
fn=imnoise(f,'gaussian',0,0.01);
figure;
imshow(fn);
title('„ßË¹ÔëÉù')
```

%%%%%%%%%

```
w1=fspecial('average',[3,3]);
f1=imfilter(fn,w1,'conv','replicate','same');
figure
imshow(f1);
title('3*3ËäËð¾ùÖµÂË²')
```

%%%%%%%%%

```
f2=gmean(fn,3,3);
figure
imshow(f2);
title('3*3¼, °î¾ùÖµÂË²')
```

%%%%%%%%%

```
f3=medfilt2(fn,[3 3],'symmetric')
figure
imshow(f3);
title('3*3ÖÐÖµÖµÂË²')
```

#### six2.m

```
clear;
[f,map_f]=imread('C:\Users\15pr\Desktop\ÐÂ½"îÄ¼p¼Ð
(2)\µÚ6'î×÷Òµ\lena.bmp');
figure;
imshow(f);
title('Ô-í¼')
fn=imnoise(f,'salt & pepper',0.1);
```

```

figure;
imshow(fn);
title('½·ÑîÔëÉù')
%%%%%%%%%%
w1=fspecial('average',[3,3]);
f1=imfilter(fn,w1,'conv','replicate','same');
figure
imshow(f1);
title('3*3ËäÊõüÖµÂË²')
%%%%%%%%%%
f2=gmean(fn,3,3);
figure
imshow(f2);
title('3*3¼,°îüÖµÂË²')
%%%%%%%%%%
f3=medfilt2(fn,[3 3],'symmetric');
figure
imshow(f3);
title('3*3ÖÐÖµÖµÂË²')
%%%%%%%%%%
f4=charmean(fn,3,3,-1);
figure
imshow(f4);
title('3*3&mu÷°íÖµQ=-1ÂË²')
%%%%%%%%%%
f5=charmean(fn,3,3,-0.5);
figure
imshow(f5);
title('3*3&mu÷°íÖµQ=-0.5ÂË²')
%%%%%%%%%%
f6=charmean(fn,3,3,0);
figure
imshow(f6);
title('3*3&mu÷°íÖµQ=0ÂË²')
%%%%%%%%%%
f7=charmean(fn,3,3,0.5);
figure
imshow(f7);
title('3*3&mu÷°íÖµQ=0.5ÂË²')
%%%%%%%%%%
f4=charmean(fn,3,3,1);
figure
imshow(f4);
title('3*3&mu÷°íÖµQ=1ÂË²')

```

```

six3.m
clc;clear;
img=imread('lena.bmp');
I=im2double(img);
len=45;%ÖËŦÎ»ÒÆÎ^25,öİñËØ
theta=45;%¼ÇŦËÎ^45
psf=fspecial('motion',len,theta);%²úÉúµãÀ©É¢°-Êýpsf
I1=imfilter(I,psf,'conv','circular');%ÖËŦÎÄ£°ý
k=0.5;
I1=imnoise(I1,'gaussian',0,0.01);
figure;
subplot(1,2,1),imshow(I),title('Ô-Ê¼íîñ');
subplot(1,2,2),imshow(I1),title('Ä£°ýíîñ');
% % for(k=0.01:0.02:0.20)
I2=deconvwnr(I1,psf,k);%Î-ÄÉÂË²°,Ô-
figure;
imshow(I2),title(['ÂË²°,Ô-íîñk=',num2str(k)]);
% % end
%¼ÓËë,ßË¹ÔëÉù

```

```

function f=charmean(g,m,n,q)
g=single(g);
f=imfilter(g.^(q+1),ones(m,n),'replicate');
f=f./(imfilter(g.^q,ones(m,n),'replicate')+eps);
f=uint8(f);
end
function f=gmean(g,m,n)
g=single(g);
f=exp(imfilter(log(g),ones(m,n),'replicate')).^(1/m/n);
f=uint8(f);
end

```