数字图像处理

第六次作业

姓名：张勤东

班级：自动化 63

学号：2160504080

日期：2019 年 4 月 1 日

摘要：本报告主要解决了数字图像处理中关于图像恢复的一些问题。在测试图像上产生指定均值和方差的高斯噪声，使用高斯滤波器、均值滤波器、统计排序滤波器等多种滤波器恢复图像，并分析各滤波器在处理高斯噪声的优缺点。在测试图像上产生椒盐噪声，使用高斯滤波器、均值滤波器、统计排序滤波器等多种滤波器恢复图像，并分析各滤波器在处理椒盐噪声的优缺点。推导维纳滤波器，使用模糊滤波器对图像进行退化并加入高斯噪声，并分别使用维纳滤波和约束最小二乘滤波恢复图像。

1. 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；

解决思路：

在原始图像上加上高斯噪声，得到只存在噪声的退化图像。

在空域这一过程为：

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

在频域这一过程为：

$$G(u,v) = F(u,v) + N(u,v)$$

在只存在加性噪声的情况下，可以使用空域滤波或频域滤波的方法对退化图像进行处理，得到恢复图像。

其中，使用空域滤波器恢复图像的工作原理是：选择一个邻域，然后对该邻域包围的图像像素进行预定义的操作，产生一个新像素，并将新像素的值赋给邻域中心点，移动邻域重复以上操作直至所有的点都被重新赋值。由此得到的重新赋值的图像即为滤波恢复的图像。

使用频域滤波器恢复图像的工作原理是：先根据以下公式，计算出退化图像对应的频域的谱。

$$G(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

然后，将该频谱$G(u,v)$与对应滤波器的频谱$H(u,v)$相乘得到滤波后的频谱：

$$\hat{F}(u,v) = H(u,v) \times G(u,v)$$

最后，根据以下公式，计算出经过低通滤波器处理后的图像。

$$\hat{f}(x,y) = \text{real}\{\mathcal{F}^{-1}[\hat{F}(u,v)]\}$$

高斯噪声又称正态噪声，高斯随机变量 z 的 PDF 为：

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(z-\bar{z})^2/2\sigma^2}$$

使用的滤波器原理如下：

1.高斯滤波器

选取的空域高斯滤波器可有如下公式产生：

$$f(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x_1^2+x_2^2}{2\sigma^2}}$$

根据公式可以算出其对邻域内各点的计算系数，实现高斯低通滤波的功能。

2.算术均值滤波器

$$f(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t)$$

3.几何均值滤波器

$$f(x,y) = [\prod_{(s,t)\in S_{xy}} g(s,t)]^{\frac{1}{mn}}$$

4.谐波均值滤波器

$$f(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

5.中值滤波器

$$f(x,y) = \text{median}_{(s,t)\in S_{xy}}\{g(s,t)\}$$

6.最大值滤波器

$$f(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$$

7.最小值滤波器

$$f(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

8.中点滤波器

$$f(x,y) = \frac{1}{2}[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}]$$

在 matlab 中，使用 imread()函数读入图像，使用 ind2gray()函数，将索引图像和调色板读为一幅灰度图像。使用 imnoise()函数加入均值为 0，方差为 100 的高斯噪声。使用 imgaussfilt()实现高斯滤波。使用 medfilt2()实现中值滤波。使用 fspecial()和 imfilter()函数，选择参数为'average'，进行均值滤波。根据以上公式实现其他均值滤波器和统计排序滤波器。最后，用 imshow()显示滤波后图像。

处理结果：

加入噪声后图像为

原图像

退化图像

使用方差为 1 的高斯滤波器

退化图像

恢复图像

使用方差为 4 的高斯滤波器

退化图像

恢复图像

使用大小为 5*5 的算术均值滤波器

退化图像

恢复图像

使用大小为 15*15 的算术均值滤波器

退化图像

恢复图像

使用大小为 5*5 的几何均值滤波器

退化图像

恢复图像

使用大小为 5*5 的谐波均值滤波器

退化图像

恢复图像

使用大小为 5*5 的中值滤波器

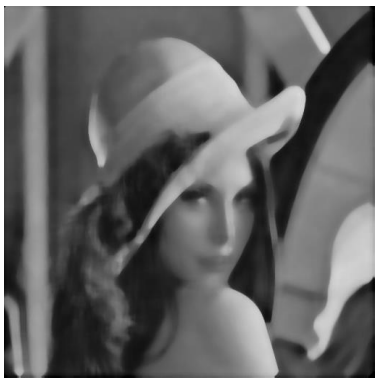
退化图像

恢复图像

使用大小为 15*15 的中值滤波器

退化图像

恢复图像



使用大小为 5*5 的最大值滤波器

退化图像

恢复图像

使用大小为 5*5 的最小值滤波器

退化图像

恢复图像

使用大小为 5*5 的中点滤波器

退化图像

恢复图像

结果分析：

观察比较原始图像和加入高斯噪声的图像，可以发现加入均值为 0、方差为 100 的高斯噪声对原图像产生了一定的影响，尤其是在原始图像较为光滑的地方高斯噪声的影响比较明显。

观察以上各种滤波器恢复的图像，总体而言，加入滤波器后对高斯噪声有一定的平滑作用，但是也在一定程度上导致图像细节变得模糊。

对比不同方差的高斯滤波器，可以发现，方差越大的高斯滤波器，对高斯噪声的滤除效果越明显，但也导致图像整体变得模糊。高斯滤波器对高斯噪声的效

果较为不错。

对比不同尺寸大小的算术均值滤波器和中值滤波器，可以发现，滤波器尺寸越大，低通效果越明显，高斯噪声消除越明显，但是图像会越模糊，细节信息丢失较多。并且相同尺寸的中值滤波器滤波后引起的模糊会更少。

观察几何均值滤波器和谐波均值滤波器的恢复图像，可以发现，这两个滤波器有较为不错的效果。

最大值滤波器对含有较亮噪声较为敏感，最小值对含有较暗噪声较为敏感。经过最大值滤波器滤波后图像会偏亮，而经过最小值滤波器滤波后图像会偏暗。中点滤波器滤波后，图像灰度不会发生大的变化，但是平滑高斯噪声效果较差。总体上，最大值滤波器、最小值滤波器和中点滤波器不适合处理高斯噪声。

2. 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；

解决思路：

在原始图像上加上椒盐噪声，得到只存在噪声的退化图像。

在空域这一过程为：

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

在频域这一过程为：

$$G(u,v) = F(u,v) + N(u,v)$$

在只存在加性噪声的情况下，可以使用空域滤波的方法对退化图像进行处理，得到恢复图像。

其中，使用空域滤波器恢复图像的工作原理是：选择一个邻域，然后对该邻域包围的图像像素进行预定义的操作，产生一个新像素，并将新像素的值赋给邻域中心点，移动邻域重复以上操作直至所有的点都被重新赋值。由此得到的重新赋值的图像即为滤波恢复的图像。

椒盐噪声又称脉冲噪声，双极脉冲噪声的随机变量 z 的 PDF 为：

$$p(z) = \begin{cases} P_a, & z = a \\ P_b, & z = b \\ 1 - P_a - P_b, & 其他 \end{cases}$$

如果，$b > a$，则灰度级 b 在图像中显示为一个亮点；反之，灰度级 a 将在图像中显示为一个暗点。当P_a和P_b近似相等时，则脉冲噪声值将类似于在图像上随机分布的胡椒和盐分微粒。

使用的滤波器原理如下：

1.高斯滤波器

选取的空域高斯滤波器可有如下公式产生：

$$\mathrm{f(x,y)} = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x_1^2+x_2^2}{2\sigma^2}}$$

根据公式可以算出其对邻域内各点的计算系数，实现高斯低通滤波的功能。

2.算术均值滤波器

$$\mathrm{f(x,y)} = \frac{1}{\mathrm{mn}} \sum_{(s,t)\in S_{xy}} g(s,t)$$

3.几何均值滤波器

$$\mathrm{f(x,y)} = [\prod_{(s,t)\in S_{xy}} g(s,t)]^{\frac{1}{mn}}$$

4.谐波均值滤波器

$$\mathrm{f(x,y)} = \frac{\mathrm{mn}}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

5.逆谐波均值滤波器

$$\mathrm{f(x,y)} = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

6.中值滤波器

$$\mathrm{f(x,y)} = \mathrm{median}_{(s,t)\in S_{xy}}\{\mathrm{g(s,t)}\}$$

7.最大值滤波器

$$\mathrm{f(x,y)} = \mathrm{max}_{(s,t)\in S_{xy}}\{\mathrm{g(s,t)}\}$$

8.最小值滤波器

$$\mathrm{f(x,y)} = \mathrm{min}_{(s,t)\in S_{xy}}\{\mathrm{g(s,t)}\}$$

9.中点滤波器

$$\mathrm{f(x,y)} = \frac{1}{2}[\mathrm{max}_{(s,t)\in S_{xy}}\{\mathrm{g(s,t)}\} + \mathrm{min}_{(s,t)\in S_{xy}}\{\mathrm{g(s,t)}\}]$$

在 matlab 中，使用 imread()函数读入图像，使用 ind2gray()函数，将索引图像和调色板读为一幅灰度图像。使用 imnoise()函数加入密度是 0.1 的椒盐噪声。为了进一步分析各滤波器对盐噪声和椒噪声的滤波效果，根据椒噪声和盐噪声的概率密度函数再分别生成密度均是 0.1 的椒噪声和盐噪声。使用 imgaussfilt()实现高斯滤波。使用 medfilt2()实现中值滤波。使用 fspecial()和 imfilter()函数，选择参数为'average'，进行均值滤波。根据以上公式实现其他均值滤波器和统计排序滤波器。最后，用 imshow()显示滤波后图像。

处理结果：

处理密度为 0.1 的椒盐噪声

加入噪声后图像

原图像

退化图像

使用方差为 1 的高斯滤波器

退化图像

恢复图像

使用大小为 5*5 的算术均值滤波器

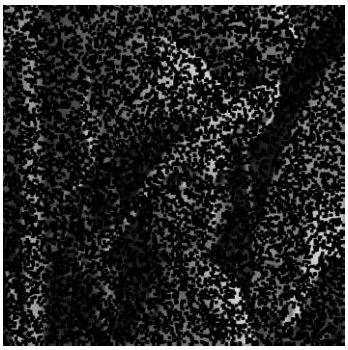
退化图像

恢复图像

使用大小为 5*5 的几何均值滤波器

退化图像

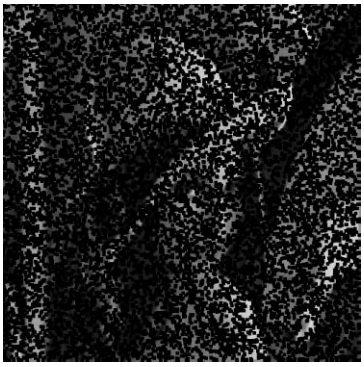
恢复图像



使用大小为 5*5 的谐波均值滤波器

退化图像

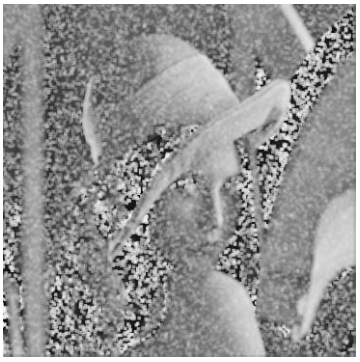
恢复图像



使用大小为 5*5、Q=1.5 逆谐波均值滤波器

退化图像

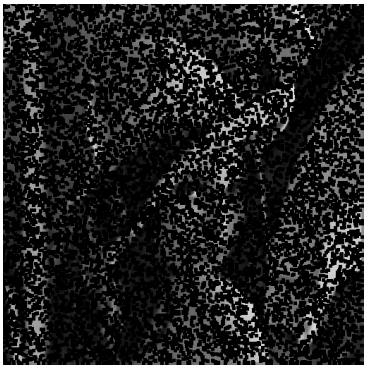
恢复图像



使用大小为 5*5、Q=-1.5 逆谐波均值滤波器

退化图像

恢复图像



使用大小为 5*5 的中值滤波器

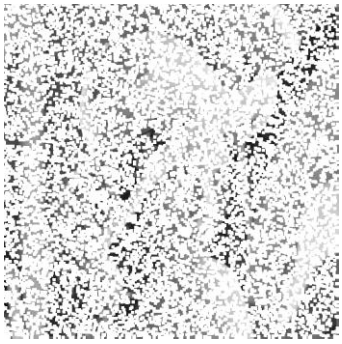
退化图像

恢复图像

使用大小为 5*5 的最大值滤波器

退化图像

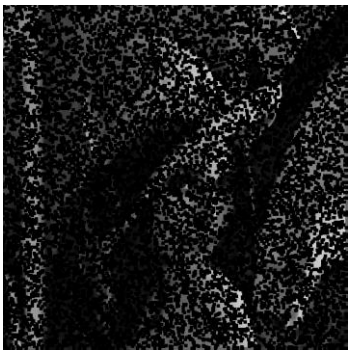
恢复图像



使用大小为 5*5 的最小值滤波器

退化图像

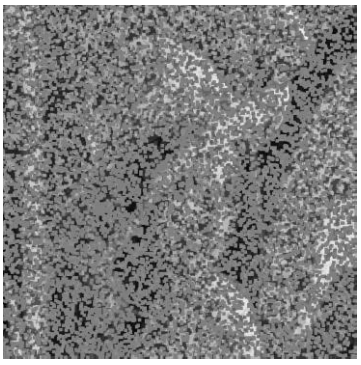
恢复图像



使用大小为 5*5 的中点滤波器

退化图像

恢复图像



处理密度为 0.1 的盐噪声

加入噪声后图像

原图像

退化图像

使用大小为 5*5 的几何均值滤波器

退化图像

恢复图像

使用大小为 5*5 的谐波均值滤波器

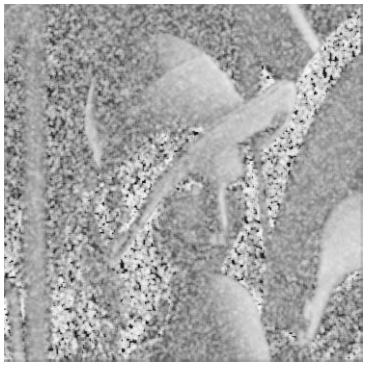
退化图像

恢复图像

使用大小为 5*5、Q=1.5 逆谐波均值滤波器

退化图像

恢复图像



使用大小为 5*5、Q=-1.5 逆谐波均值滤波器

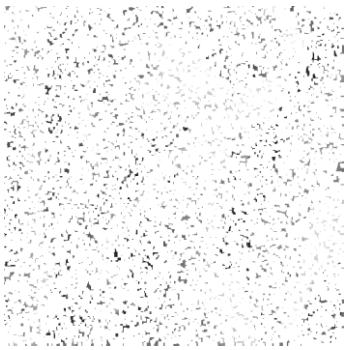
退化图像

恢复图像

使用大小为 5*5 的最大值滤波器

退化图像

恢复图像



使用大小为 5*5 的最小值滤波器

退化图像

恢复图像

使用大小为 5*5 的中点滤波器

退化图像

恢复图像

处理密度为 0.1 的椒噪声

加入噪声后图像

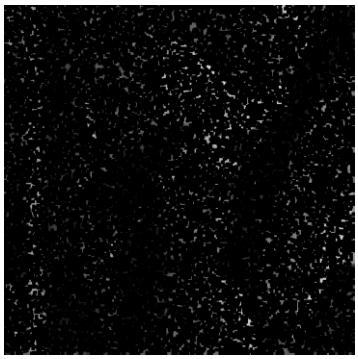
原图像

退化图像

使用大小为 5*5 的几何均值滤波器

退化图像

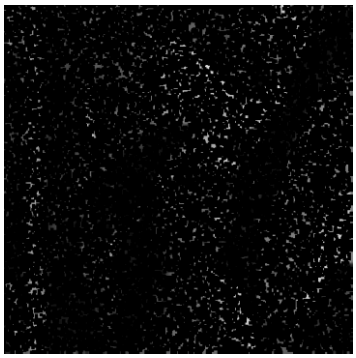
恢复图像



使用大小为 5*5 的谐波均值滤波器

退化图像

恢复图像



使用大小为 5*5、Q=1.5 逆谐波均值滤波器

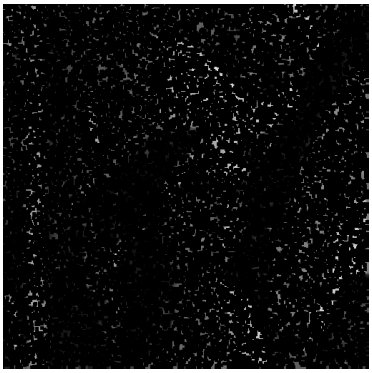
退化图像

恢复图像

使用大小为 5*5、Q=-1.5 逆谐波均值滤波器

退化图像

恢复图像



使用大小为 5*5 的最大值滤波器

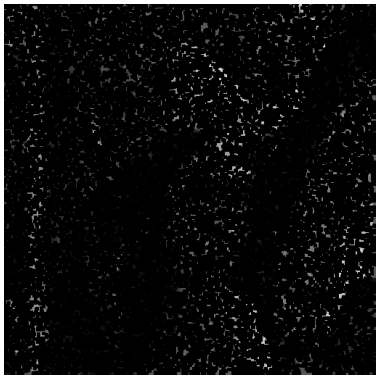
退化图像

恢复图像

使用大小为 5*5 的最小值滤波器

退化图像

恢复图像



使用大小为 5*5 的中点滤波器

退化图像

恢复图像

结果分析：

观察以上分别加入椒盐噪声、盐噪声和椒噪声的图像，可以发现，在图像中加入盐噪声的效果是图中有 10%的像素点的灰度值变为 255，显示为亮点，而加入椒噪声的效果是图中有 10%的像素点的灰度值变为 0，显示为暗点。

观察使用各种滤波器分别处理椒盐噪声、盐噪声和椒噪声的恢复图像，可以发现不同滤波器对盐噪声和椒噪声有不同的效果。

高斯滤波器和算术均值滤波器可以处理椒噪声和盐噪声。使用这两种滤波器可以平滑图像，使盐噪声和椒噪声不明显。但是，却使图像变得模糊，并且去除

椒盐噪声的效果并不是很好。

中值滤波器的处理效果最好，可以处理椒噪声和盐噪声。经过中值滤波器处理的图像，不仅很好地去除了椒盐噪声，并且原图像的细节保留得也十分完整。

几何均值滤波器、谐波滤波器和 Q=-1.5 的逆谐波滤波器可以用来处理盐噪声。这三个滤波器对图像中灰度值为 0 的点十分敏感，不能用来处理椒噪声。并且这三个滤波器处理盐噪声时，会使图像变模糊。三个滤波器处理盐噪声时，逆谐波滤波器效果最好，谐波滤波器次之，几何均值滤波器效果最差。

Q=1.5 的逆谐波滤波器可以用来处理椒噪声。逆谐波滤波器对灰度值为 255 的亮点非常敏感，因此不能处理盐噪声。用逆谐波滤波器处理含有椒噪声的图像后，图像会变得模糊，但总体效果较好。

比较 Q=1.5 和 Q=-1.5 的逆谐波滤波器，可以发现，Q 为正时，逆谐波滤波器可以消除胡椒噪声，Q 为负时，逆谐波滤波器可以消除盐粒噪声，但是该滤波器不能同时消除椒盐噪声。

最大值滤波器对灰度值为 255 的亮点非常敏感，因此不能处理盐噪声，可以在一定程度上用来处理椒噪声，但是处理后图像会偏亮，有轻微的失真。

最小值滤波器对灰度值为 0 的暗点非常敏感，因此不能处理椒噪声，可以在一定程度上用来处理盐噪声，但是处理后图像会偏暗，有轻微的失真。

中点滤波器对椒噪声和盐噪声的处理效果都较差。

3. 推导维纳滤波器并实现下边要求；

(a) 实现模糊滤波器如方程 Eq. (5.6-11).

(b) 模糊 lena 图像：45 度方向，T=1；

(c) 再模糊的 lena 图像中增加高斯噪声，均值= 0 ，方差=10 pixels 以产生模糊图像；

(d)分别利用方程 Eq. (5.8-6)和(5.9-4)，恢复图像；并分析算法的优缺点.

解决思路：

维纳滤波器推导如下：

维纳滤波器要求误差均方最小，即$\min(e^2)$。若误差均方最小，则误差与观测信号不相关，则有：

$$\mathrm{E}\{\mathrm{e}\cdot g^*\}=0$$

根据 e(n1,n2)的定义有：

$$\mathrm{E}\{[\mathrm{w}(\mathrm{n1},\mathrm{n2})*\mathrm{g}(\mathrm{n1},\mathrm{n2})-\mathrm{f}(\mathrm{n1},\mathrm{n2})]\cdot g^*(l1,l2)\}=0$$

展开有：

$$\mathrm{E}\left\{\sum_{m1}\sum_{m2} w(m1,m2)g(n1-m1,n2-m2)\,g^*(l1,l2)\right\} = E\{f(n1,n2)g^*(l1,l2)\}$$

交换运算顺序：

$$\sum_{m1}\sum_{m2} w(m1,m2)E\{g(n1-m1,n2-m2)g^*(l1,l2)\} = E\{f(n1,n2)g^*(l1,l2)\}$$

化简得：

$$\mathrm{w(n1-l1,n2-l2)} * r_g(n1-l1,n2-l2) = r_{fg}(n1-l1,n2-l2)$$

进行傅里叶变换得：

$$\mathrm{W(w1,w2)} = \frac{P_{fg}(w1,w2)}{P_g(w1,w2)} \quad \cdots\cdots(1)$$

由$r_{fg}(\mathrm{n1,n2})$定义有：

$$r_{fg}(\mathrm{n1,n2}) = \mathrm{E\{f(n1}+l1,\mathrm{n2}+l2)\cdot g^*(l1,l2)\}$$

$$r_{fg}(\mathrm{n1,n2}) = \mathrm{E\{f(n1}+l1,\mathrm{n2}+l2)\cdot[f^*(\mathrm{l1,l2})*h^*(\mathrm{l1,l2})+n^*(\mathrm{l1,l2})]\}$$

$$r_{fg}(\mathrm{n1,n2}) = \sum_{m1}\sum_{m2} n^*(m1,m2)E\{f^*(l1-m1,l2-m2)\mathrm{f(n1}+l1,\mathrm{n2}+l2)\}$$

$$r_{fg}(\mathrm{n1,n2}) = n^*(-n1,-n2) * r_f(n1,n2)$$

进行傅里叶变换得：

$$P_{fg}(w1,w2) = H^*(w1,w2)P_f(w1,w2) \quad \cdots\cdots(2)$$

由$r_f(\mathrm{n1,n2})$定义有：

$$r_f(\mathrm{n1,n2}) = \mathrm{E\{}g(\mathrm{n1}+l1,\mathrm{n2}+l2)\cdot g^*(l1,l2)\}$$

$$r_f(\mathrm{n1,n2}) = \mathrm{E\{[h(n1}+\mathrm{l1,n2}+\mathrm{l2)} * \mathrm{f(n1}+\mathrm{l1,n2}+\mathrm{l2)}+\mathrm{n(n1}+\mathrm{l1,n2}+\mathrm{l2)}]$$
$$\cdot[\mathrm{h^*(l1,l2)} * \mathrm{f^*(l1,l2)}+\mathrm{n^*(l1,l2)}]\}$$

$$r_f(\mathrm{n1,n2}) = \sum_{m1}\sum_{m2}\sum_{k1}\sum_{k2} h(m1,m2)\mathrm{h^*(k1,k2)}\,r_f(n1-m1+k1,n2-m2+l2)$$
$$+ r_n(n1,n2)$$

$$r_f(\mathrm{n1,n2}) = h^*(-n1,-n2) * h(n1,n2) * r_f(n1,n2) + r_n(n1,n2)$$

进行傅里叶变换得：

$$P_g(w1,w2) = H^*(w1,w2)H(w1,w2)P_f(w1,w2) + P_n(w1,w2) \quad \cdots\cdots(3)$$

将(2)(3)式代入(1)式得：

$$\mathrm{W(w1,w2)} = \frac{H^*(w1,w2)P_f(w1,w2)}{\|H(w1,w2)\|^2 P_f(w1,w2) + P_n(w1,w2)}$$

得到维纳滤波器。

对图像进行模糊退化并加噪声后，图像退化公式为：

$$\mathrm{g(x,y)} = \mathrm{h(x,y)} * \mathrm{f(x,y)} + \mathrm{n(x,y)}$$

对应频域退化公式为：

$$\mathrm{G(u,v)} = \mathrm{H(u,v)F(u,v)} + \mathrm{N(u,v)}$$

运动模糊退化公式为：

$$\mathrm{H(u,v)} = \frac{\mathrm{T}}{\pi(\mathrm{ua} + \mathrm{vb})}\sin[\pi(\mathrm{ua} + \mathrm{vb})]e^{-j\pi(ua+vb)}$$

高斯噪声又称正态噪声，高斯随机变量 z 的 PDF 为：

$$\mathrm{p(z)} = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(z-\bar{z})^2/2\sigma^2}$$

使用维纳滤波恢复图像公式为：

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

选取合适得 K 值可以实现较好得维纳滤波效果。

约束最小二乘滤波公式为：

$$\hat{F}(u,v) = [\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2}]G(u,v)$$

P(u,v)是如下函数得傅里叶变换：

$$\mathrm{P(x,y)} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

计算 γ 的步骤为：

1. 指定 γ 的初始值
2. 计算$\|r\|^2$。其中，$\mathrm{r} = \mathrm{g} - \mathrm{H}\hat{f}$。
3. 若满足$\|r\|^2 = \|n\|^2 \pm a$，则停止，此时的图像已为对原始图像的最佳估计。若$\|r\|^2 < \|n\|^2 - a$，则增大 γ；若$\|r\|^2 > \|n\|^2 - a$，则减小 γ，返回步骤 2 继续循环。

在 matlab 中，使用 imread()函数读入图像，使用 ind2gray()函数，将索引图像和调色板读为一幅灰度图像。使用 fft2()和 fftshift()函数得到原图像的频谱图。根据运动模糊函数，取 T=1，a=b=0.1，产生相应的频域退化函数。同时，作为对比，使用函数 fspecial，调用参数'motion'，对图像进行运动模糊处理。将模糊处

理后图像加入均值为 0、方差为 10 的高斯噪声。接着，使用维纳滤波器对退化图像进行恢复，K 值取估算的噪声信号比 0.04。使用约束最小二乘滤波，进行迭代运算，计算出合适的 γ 值，对图像进行恢复。

处理结果：

根据书上公式得出的运动模糊函数进行退化

经过运动模糊并加高斯噪声后图像：

维纳滤波后图像：

约束最小二乘滤波得到图像：

计算出 γ =4.1727e-07

使用 matlab 自带的运动模糊函数进行退化

经过运动模糊并加高斯噪声后图像：

维纳滤波后图像：

约束最小二乘滤波得到图像：

计算出 $\gamma = 0.0854$

结果分析：

观察以上图像可知，根据书上公式得出的运动模糊函数可以实现模拟运动的图像退化情况，但是 matlab 自带函数更加符合实际情况。

使用维纳滤波器和约束最小二乘滤波器可以将经过运动模糊和高斯噪声退化的图像进行一定的恢复，并且具有较好的效果。

维纳滤波器和约束最小二乘滤波器两者相比，约束最小二乘滤波器恢复的图像更加接近原图像，细节处比维纳滤波器恢复的更好，并且不需要对图像的信噪比进行估计。但是，维纳滤波器计算更加简单，因此恢复一幅图像需要的时间更短，而约束最小二乘滤波器计算 γ 时迭代收敛速度较慢，需要更多的时间进行计算。

附录

第一题代码

```
[pic,map]=imread('lena.bmp');
t=cell(14,1);
t{1,1}=ind2gray(pic,map);
t{2,1}= imnoise(t{1,1},'gaussian',0,0.0016); %方差为 100

t{3,1}=imgaussfilt(t{2,1},1);
t{4,1}=imgaussfilt(t{2,1},4);

t{5,1}=medfilt2(t{2,1},[5,5]);
t{6,1}=medfilt2(t{2,1},[15,15]);

a1=fspecial('average',5);
a2=fspecial('average',15);
t{7,1}=imfilter(t{2,1},a1);
t{8,1}=imfilter(t{2,1},a2);

q=round(mean(mean(t{2,1})));
t{9,1}=uint8(ones(516,516));
t{9,1}=t{9,1}*q;
for i=1:512
    for j=1:512
        t{9,1}(i+2,j+2)=t{2,1}(i,j);
    end
end

for i=3:514
    for j=3:514
        sxy(1:5,1:5)=t{9,1}(i-2:i+2,j-2:j+2);
        jihe=1;
        xiebo=0;
        for ii=1:5
            for jj=1:5
                jihe=jihe*double(sxy(ii,jj));
                xiebo=xiebo+(1/double(sxy(ii,jj)));
            end
        end
        t{10,1}(i-2,j-2)=jihe^(1/25);
        t{11,1}(i-2,j-2)=25/xiebo;
        t{12,1}(i-2,j-2)=max(max(sxy));
        t{13,1}(i-2,j-2)=min(min(sxy));
```

```
            t{14,1}(i-2,j-2)=(0.5*t{12,1}(i-2,j-2)+0.5*t{13,1}(i-2,j-2));
        end
end

figure(1)
subplot(1,2,1)
imshow(uint8(t{1,1}))
title('原图像')
subplot(1,2,2)
imshow(uint8(t{2,1}))
title('退化图像')

for k=3:14
        figure(k)
        subplot(1,2,1)
        imshow(uint8(t{2,1}))
        title('退化图像')
        subplot(1,2,2)
        imshow(uint8(t{k,1}))
        title('恢复图像')

end
```

第二题代码

加椒盐噪声

```
[pic,map]=imread('lena.bmp');
t=cell(16,1);
t{1,1}=ind2gray(pic,map);
t{2,1}= imnoise(t{1,1},'salt & pepper',0.1);



t{3,1}=imgaussfilt(t{2,1},1);
t{4,1}=imgaussfilt(t{2,1},4);

t{5,1}=medfilt2(t{2,1},[5,5]);
t{6,1}=medfilt2(t{2,1},[15,15]);

a1=fspecial('average',5);
a2=fspecial('average',15);
t{7,1}=imfilter(t{2,1},a1);
t{8,1}=imfilter(t{2,1},a2);
```

```
q=round(mean(mean(t{2,1})));
t{9,1}=uint8(ones(516,516));
t{9,1}=t{9,1}*q;
for i=1:512
    for j=1:512
        t{9,1}(i+2,j+2)=t{2,1}(i,j);
    end
end

q1=1.5;
q2=-1.5;

for i=3:514
    for j=3:514
        sxy(1:5,1:5)=t{9,1}(i-2:i+2,j-2:j+2);
        jihe=1;
        xiebo=0;
        nxb111=0;
        nxb112=0;
        nxb121=0;
        nxb122=0;
        for ii=1:5
            for jj=1:5
                jihe=jihe*double(sxy(ii,jj));
                xiebo=xiebo+(1/double(sxy(ii,jj)));
                nxb111=nxb111+double(sxy(ii,jj))^(q1+1);
                nxb112=nxb112+double(sxy(ii,jj))^q1;
                nxb121=nxb121+double(sxy(ii,jj))^(q2+1);
                nxb122=nxb122+double(sxy(ii,jj))^q2;
            end
        end
        t{10,1}(i-2,j-2)=jihe^(1/25);
        t{11,1}(i-2,j-2)=25/xiebo;
        t{12,1}(i-2,j-2)=max(max(sxy));
        t{13,1}(i-2,j-2)=min(min(sxy));
        t{14,1}(i-2,j-2)=(0.5*t{12,1}(i-2,j-2)+0.5*t{13,1}(i-2,j-2));
        t{15,1}(i-2,j-2)=nxb111/nxb112;
        t{16,1}(i-2,j-2)=nxb121/nxb122;
    end
end

figure(1)
subplot(1,2,1)
```

```
imshow(uint8(t{1,1}))
title('原图像')
subplot(1,2,2)
imshow(uint8(t{2,1}))
title('退化图像')

for k=3:16
    figure(k)
    subplot(1,2,1)
    imshow(uint8(t{2,1}))
    title('退化图像')
    subplot(1,2,2)
    imshow(uint8(t{k,1}))
    title('恢复图像')

end
```

加盐噪声代码

```
salt=rand(512);
for i=1:512
    for j=1:512
        if salt(i,j)>=0.9
            salt(i,j)=1;
        else
            salt(i,j)=0;
        end
    end
end

salt=uint8(255*salt);

[pic,map]=imread('lena.bmp');
t=cell(16,1);
t{1,1}=ind2gray(pic,map);
t{2,1}= t{1,1}+salt;

t{3,1}=imgaussfilt(t{2,1},1);
t{4,1}=imgaussfilt(t{2,1},4);

t{5,1}=medfilt2(t{2,1},[5,5]);
t{6,1}=medfilt2(t{2,1},[15,15]);

a1=fspecial('average',5);
a2=fspecial('average',15);
```

```
t{7,1}=imfilter(t{2,1},a1);
t{8,1}=imfilter(t{2,1},a2);

q=round(mean(mean(t{2,1})));
t{9,1}=uint8(ones(516,516));
t{9,1}=t{9,1}*q;
for i=1:512
    for j=1:512
        t{9,1}(i+2,j+2)=t{2,1}(i,j);
    end
end

q1=1.5;
q2=-1.5;

for i=3:514
    for j=3:514
        sxy(1:5,1:5)=t{9,1}(i-2:i+2,j-2:j+2);
        jihe=1;
        xiebo=0;
        nxb111=0;
        nxb112=0;
        nxb121=0;
        nxb122=0;
        for ii=1:5
            for jj=1:5
                jihe=jihe*double(sxy(ii,jj));
                xiebo=xiebo+(1/double(sxy(ii,jj)));
                nxb111=nxb111+double(sxy(ii,jj))^(q1+1);
                nxb112=nxb112+double(sxy(ii,jj))^q1;
                nxb121=nxb121+double(sxy(ii,jj))^(q2+1);
                nxb122=nxb122+double(sxy(ii,jj))^q2;
            end
        end
        t{10,1}(i-2,j-2)=jihe^(1/25);
        t{11,1}(i-2,j-2)=25/xiebo;
        t{12,1}(i-2,j-2)=max(max(sxy));
        t{13,1}(i-2,j-2)=min(min(sxy));
        t{14,1}(i-2,j-2)=(0.5*t{12,1}(i-2,j-2)+0.5*t{13,1}(i-2,j-2));
        t{15,1}(i-2,j-2)=nxb111/nxb112;
        t{16,1}(i-2,j-2)=nxb121/nxb122;
    end
end
```

```
figure(1)
subplot(1,2,1)
imshow(uint8(t{1,1}))
title('原图像')
subplot(1,2,2)
imshow(uint8(t{2,1}))
title('退化图像')

for k=3:16
    figure(k)
    subplot(1,2,1)
    imshow(uint8(t{2,1}))
    title('退化图像')
    subplot(1,2,2)
    imshow(uint8(t{k,1}))
    title('恢复图像')
end

加椒噪声代码
salt=rand(512);
for i=1:512
    for j=1:512
        if salt(i,j)>=0.9
            salt(i,j)=1;
        else
            salt(i,j)=0;
        end
    end
end

salt=uint8(255*salt);

[pic,map]=imread('lena.bmp');
t=cell(16,1);
t{1,1}=ind2gray(pic,map);
t{2,1}= t{1,1}-salt;

t{3,1}=imgaussfilt(t{2,1},1);
t{4,1}=imgaussfilt(t{2,1},4);

t{5,1}=medfilt2(t{2,1},[5,5]);
t{6,1}=medfilt2(t{2,1},[15,15]);

a1=fspecial('average',5);
```

```
a2=fspecial('average',15);
t{7,1}=imfilter(t{2,1},a1);
t{8,1}=imfilter(t{2,1},a2);

q=round(mean(mean(t{2,1})));
t{9,1}=uint8(ones(516,516));
t{9,1}=t{9,1}*q;
for i=1:512
    for j=1:512
        t{9,1}(i+2,j+2)=t{2,1}(i,j);
    end
end

q1=1.5;
q2=-1.5;

for i=3:514
    for j=3:514
        sxy(1:5,1:5)=t{9,1}(i-2:i+2,j-2:j+2);
        jihe=1;
        xiebo=0;
        nxb111=0;
        nxb112=0;
        nxb121=0;
        nxb122=0;
        for ii=1:5
            for jj=1:5
                jihe=jihe*double(sxy(ii,jj));
                xiebo=xiebo+(1/double(sxy(ii,jj)));
                nxb111=nxb111+double(sxy(ii,jj))^(q1+1);
                nxb112=nxb112+double(sxy(ii,jj))^q1;
                nxb121=nxb121+double(sxy(ii,jj))^(q2+1);
                nxb122=nxb122+double(sxy(ii,jj))^q2;
            end
        end
        t{10,1}(i-2,j-2)=jihe^(1/25);
        t{11,1}(i-2,j-2)=25/xiebo;
        t{12,1}(i-2,j-2)=max(max(sxy));
        t{13,1}(i-2,j-2)=min(min(sxy));
        t{14,1}(i-2,j-2)=(0.5*t{12,1}(i-2,j-2)+0.5*t{13,1}(i-2,j-2));
        t{15,1}(i-2,j-2)=nxb111/nxb112;
        t{16,1}(i-2,j-2)=nxb121/nxb122;
    end
end
```

```
figure(1)
subplot(1,2,1)
imshow(uint8(t{1,1}))
title('原图像')
subplot(1,2,2)
imshow(uint8(t{2,1}))
title('退化图像')

for k=3:16
    figure(k)
    subplot(1,2,1)
    imshow(uint8(t{2,1}))
    title('退化图像')
    subplot(1,2,2)
    imshow(uint8(t{k,1}))
    title('恢复图像')
end
```

第三题代码

```
[pic,map]=imread('lena.bmp');
t=cell(18,1);
t{1,1}=ind2gray(pic,map);

t{2,1}=fftshift(fft2(t{1,1}));
[x,y]=size(t{2,1});

ab=0.1;
for u=1:x
    for v=1:y
        t{3,1}(u,v)=sin(pi*ab*(u+v))*exp(-ab*(u+v)*pi*1j)/(pi*ab*(u+v));

    end
end

% H = fspecial('motion',20,135); %模糊函数，可代替 8-14 行
% t{3,1}=fftshift(fft2(H,512,512)); %模糊函数，可代替 8-14 行

t{4,1}=t{2,1}.*t{3,1};
t{5,1}=abs(ifft2(fftshift(t{4,1})));
c=max(max(t{5,1}));
d=min(min(t{5,1}));
t{6,1}=uint8(255*(t{5,1}-d)/(c-d));
```

```
t{7,1}= imnoise(t{6,1},'gaussian',0,0.00016);
t{8,1}=fftshift(fft2(t{7,1}));

k=0.04;
t{9,1}=(1./t{3,1}).*((abs(t{3,1}).^2)./(abs(t{3,1}).^2.+k)).*t{8,1};

t{10,1}=abs(ifft2(fftshift(t{9,1})));
e=max(max(t{10,1}));
f=min(min(t{10,1}));
t{11,1}=uint8(255*(t{10,1}-f)/(e-f));

p1=[0,-1,0;-1,4,-1;0,-1,0];
p=fftshift(fft2(p1,512,512));

gama1=0;
gama2=0.5;
gama=0.2;
r=0;
n2=10*512*512;
for ss=1:500
    t{12,1}=conj(t{3,1}).*t{8,1}./((abs(t{3,1})).^2.+gama*((abs(p)).^2));
    t{13,1}=t{12,1}.*t{3,1};
    t{14,1}=abs(ifft2(fftshift(t{13,1})));
    cc=max(max(t{14,1}));
    dd=min(min(t{14,1}));
    t{15,1}=uint8(255*(t{14,1}-dd)/(cc-dd));
    t{16,1}=abs(t{7,1}-t{15,1});
    r=sum(sum(t{16,1}.^2));

    if (r<n2)
        gama1=gama;
        gama=0.5*(gama1+gama2);
    else
        gama2=gama;
        gama=0.5*(gama1+gama2);
    end

    if ((n2-1)<=r&&r<=(n2+1))
        gama
        break
    end
end
t{17,1}=abs(ifft2(fftshift(t{12,1})));
```

```
ee=max(max(t{17,1}));
ff=min(min(t{17,1}));
t{18,1}=uint8(255*(t{17,1}-ff)/(ee-ff));

figure(1)
imshow(t{11,1})
figure(2)
imshow(t{18,1})
figure(3)
imshow(t{7,1})
```