

# 数字图像处理

## 第六次作业

班级：自动化 64

姓名：李明哲

学号：2160504096

提交日期：2019.04.02

## 摘要

本次实验的目的为进行图像恢复，总体分为对高斯噪声、运动模糊和椒盐噪声进行图像恢复，本次实验使用 Matlab 实现，其中部分功能使用了 Matlab 工具箱函数实现。分别使用了算数均值滤波器、几何均值滤波器、谐波均值滤波器、逆谐波均值滤波器等均值滤波器和中值滤波器、最大值和最小值滤波器、中点滤波器、修正的阿尔法滤波器等滤波器对被椒盐噪声和高斯噪声污染的图像进行图像恢复。同时，本次实验也用维纳滤波器和约束最小二乘法滤波对运动模糊和高斯噪声污染的滤波器进行图像恢复。最后，对每次的图像恢复结果进行了比较，特别的，对有些复杂的滤波器进行 Matlab 工具箱函数和自己实现的结果进行对比。

**关键词：**图像恢复，均值滤波器，统计滤波器，维纳滤波器，约束最小二乘法滤波器。

## 1. 对被高斯噪声和椒盐噪声污染的图像恢复

### 1.1 实验操作

本次实验中，首先先将图像加上高斯噪声污染，选取大约均值为 0，方差为 10 的高斯噪声对图像进行污染。随后用均值滤波器和统计滤波器对图像分别进行恢复。

其中算数均值滤波器的公式为：

$$f'(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$$

其中几何均值滤波器的公式为：

$$f'(x, y) = \left[ \prod_{(s, t) \in S_{xy}} g(x, y) \right]^{\frac{1}{mn}}$$

其中谐波均值滤波器的公式为：

$$f'(x, y) = \frac{nm}{\sum_{(s, t) \in S_{xy}} \frac{1}{g(s, t)}}$$

其中逆谐波均值滤波器的公式为：

$$f'(x, y) = \frac{\sum_{(s, t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{xy}} g(s, t)^Q}$$

其中中值滤波器的公式为：

$$f'(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\}$$

其中最大值滤波器的公式为：

$$f'(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\}$$

其中最小值滤波器的公式为：

$$f'(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

其中中点滤波器的公式为：

$$f'(x, y) = \frac{1}{2}(\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\})$$

其中修正的阿尔法滤波器的公式为：

$$f'(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g_r(s, t)$$

其中 d 的可选范围是 0 到 mn-1。

输入噪声用的是 Matlab 中的 imnoise 函数，高斯噪声和椒盐噪声的参数分别对应为 ‘gaussian’ 和 ‘salt&paper’。本次实验中椒盐噪声的参数 P 选择为 0.1，高斯噪声参数为 0 和 0.01。

## 1.2 实验结果

以下两图作为为图像加高斯噪声并恢复的结果：





以下两图为为图像加椒盐噪声并恢复的结果：





### 1.3 结论

对于被高斯噪声污染的图像使用算数均值滤波器、中值滤波器、重点滤波器和修正的阿尔法滤波器效果较好，其中算数均值滤波器对图像噪声的去除最好，但是作为代价，会损失一部分图像的细节。用其余的滤波器对图像恢复的帮助不大。比如几何均值滤波器，他们都不同程度的受到高斯噪声中像素强度值为 0 的影响。

对于被椒盐噪声污染的图像使用算数均值滤波器，中值滤波器、修正的阿尔法滤波器效果更好，尤其是对于后两中滤波器来说，几乎完美地消除了噪声，和原图像基本一样。对于反谐波滤波器来说，当  $Q$  大于 0 时，可以消除胡椒噪声，当  $Q$  小于 0 时，可以消除盐粒噪声，如果用错了滤波器，可能会造成严重的后果。比如  $Q=-1.5$  时，亮噪声几乎被清除干净，而暗噪声依然存在。但  $Q=1.5$  时，由于盐粒噪声的存在，导致图像产生了灾难性的滤波后果。所以一定要选择正确的滤波器。

## 2. 维纳滤波器与约束最小二乘法滤波器

### 2.1 实验操作

本次实验首先对图像进行了退化与加噪声处理，首先将图像进行运动模糊，然后再加入均值为 0，方差为 10 的高斯噪声。然后对其进行维纳滤波与约束最小二乘法滤波。

实验时，运动模糊的公式为：

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

其中， $T$  取 1， $a$  和  $b$  取 0.1。

维纳滤波的公式为：

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

约束最小二乘法滤波的公式为：

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

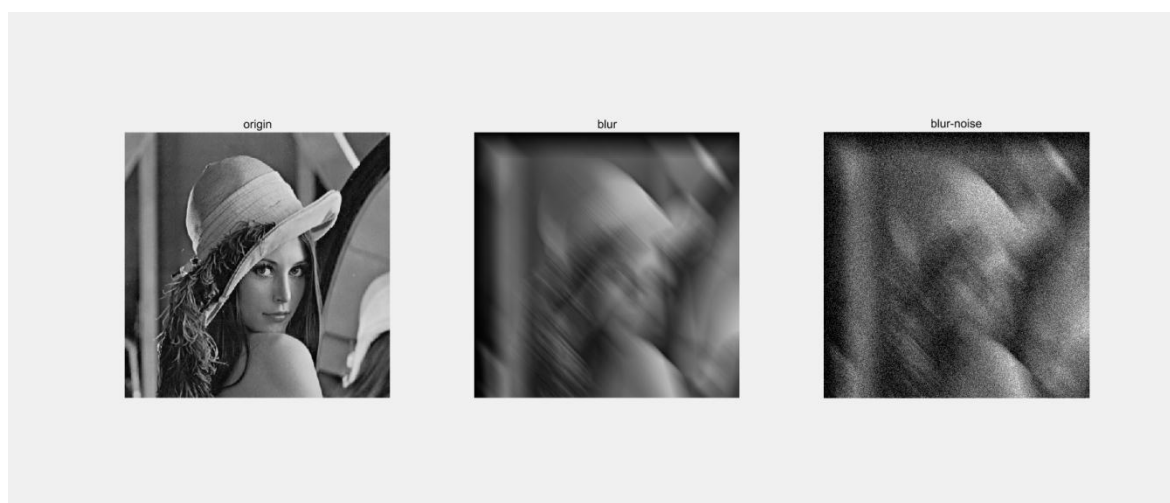
其中定义空间域的信噪比为：

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f'(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - f'(x, y)]^2}$$

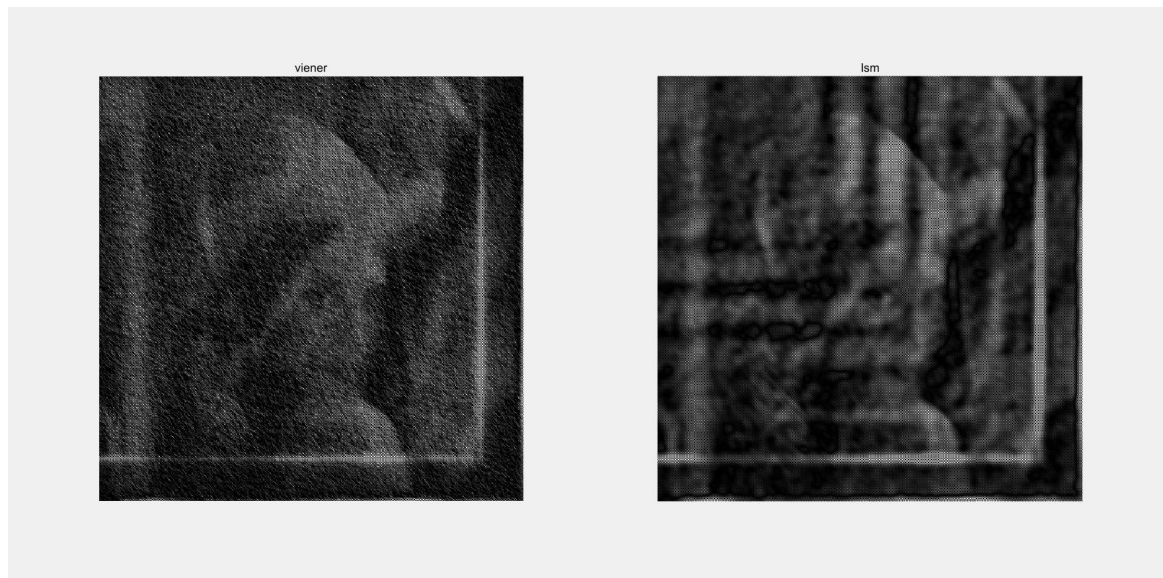
在运用上述公式进行图像恢复之后，发现结果不是很好，虽然采用了交互式选择参数，但是最后效果仍然很不好。最后选择了 Matlab 工具箱里的相关函数，运用 fspecial 中 ‘motion’ 来模拟运动模糊。运用 deconvwnr 来进行维纳滤波，利用 deconvreg 来进行约束最小二乘法滤波，并选择参数为 0.01\*512\*512。

## 2.2 实验结果

以下是加噪声的结果：

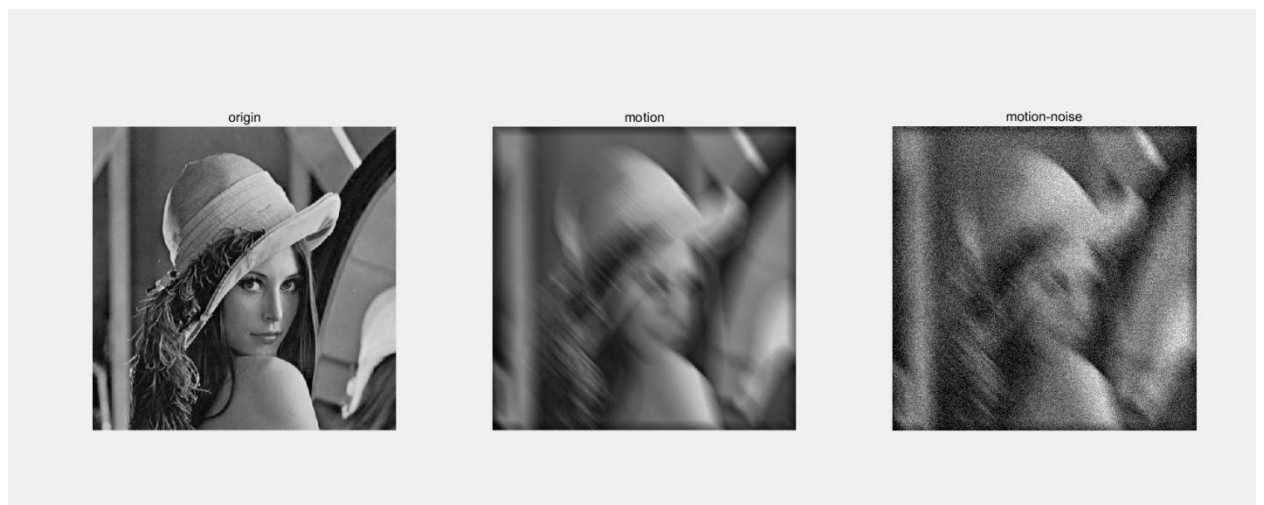


以下是自己书写的两种滤波器结果，结果很差，选择  $K$  为 0.013， $\gamma$  为 0.5.



$\text{SNR}_{\text{viener}} = 2.9894$      $\text{SNR}_{\text{lsm}} = 3.1690$

以下是用 Matlab 工具箱函数写的运动模糊过程和两种滤波器结果，并交互式的选择了参数：





SNR\_vienner =

2.0836	2.5082	2.8406	3.1413	3.4334	3.7190	4.0030
4.2789	4.5458	4.8021				

SNR\_lsm =

3.4084	3.6249	3.8596	4.1662	4.7611	4.8391	4.8684
4.8717	4.8717	4.8717				

## 2.3 结论

两次试验都采用信噪比 SNR 作为实验最后的评判标准之一，当然，另一个评判标准就是图像恢复的整体质量，这样才能够交互式的选择 K。利用书上提供的方程实现频域滤波的结果并不好，实验中也进行了交互式选择参数，但因



为结果都不太好，所以没有列出。最终选择了  $K$  为 0.013， $\gamma$  为 0.5，但是结果可以基本辨认出图像本来面目，不再模糊。但是图像中充斥着大量的噪声与摩尔纹（将图像放大看更加明显），这些都是需要解决的问题。可能是参数的问题，也可能是实验本身原理实现的问题，需要进一步查证。

所以之后选择了 Matlab 工具箱函数做实验，所得结果好了很多。并且都进行了参数的交互式选择。从维纳滤波器来看，当参数过小时，图像被噪声污染严重，参数过大时对运动模糊的消除并不是很好，综合考虑两个因素，选取了参数为 0.05，对于约束最小二乘法滤波，参数过小时，图像失真严重，过大时模糊严重，最终选择了参数为 1.0，维纳滤波和约束最小二乘法滤波能够较好的处理现实中真是的图像恢复问题。

同时，我们发现 SNR 与最终参数的选择并不一致，这也就更加证实了 SNR 最小，不一定图像视觉效果最优，这也是两种滤波器的缺点，为了克服这一缺点，更加需要参数的交互式选择。

### 3. 心得体会

本次图像恢复实验让我对课上学习的知识进行了一次实践。了解了 Matlab 中有关数字图像处理的相关基础函数，包括 `fspecial`、`deconvreg`、`deconvwnr` 等函数。在使用这些函数时，一定要仔细阅读它的使用文档，了解每一个参数的意义，确保正确的使用，同时要对他们的源代码进行研究，找出为什么自己写的函数运行效果很差，Matlab 里面的函数运行效果很好的原因。同时，我也对课上学习的图像恢复的知识有了更深的了解，更深的了解到了各种算法的实现效果与适用情况，了解到了它们的优缺点。这些都是数字图像处理的基础方面。