

数字图像处理实验二

09021227 金桥

2023 年 11 月 4 日

1 实验目标

1.1 第一部分

构建一个灰度值全是 1 的灰度图像，加入一定方差的高斯白噪声（要能看到明显的噪声），判断频谱图是否接近水平；阅读论文，对 Non-Local Means 算法的中心思想进行理解和总结。对 Lena 灰度图像加入高斯噪声，分别对其进行中值处理、均值处理、自适应中值处理、Non-Local Means 处理并进行图像对比。

1.2 第二部分

对 Lena 灰度图像加入一定密度的椒盐噪声（要能看到明显的噪声），分别对其进行中值处理、均值处理、自适应中值处理、Non-Local Means 处理并进行图像对比（这个要结合课上的自适应中值滤波器思路，进行结合）

2 过程与方法

请注意，以下算法均为自主手动实现，没有调用 OpenCV 的方法。

2.1 高斯白噪声生成

高斯白噪声的生成采用了 C++11 自带的高斯随机生成器。仍然基于原有框架，对每个像素点叠加一个高斯分布随机数并保证结果位于 0 与 255 之间。其中生成的随机数均值为 0，标准差为 20.

2.2 椒盐噪声生成

椒盐噪声的实现相对比较简单，使每个像素点有 $\frac{30}{256}$ 的概率变成 0 或者 255 即可。

2.3 中值处理

最开始的想法是遍历每个像素点，将周围范围内的像素点取出排序，然后取出中值。但是发现速度实在太慢。因此采用以下算法：

1. 每行最开始生成一个直方图，内容为范围内的像素点。
2. 处理时从左到右遍历直方图并求和，如果求和大于范围内的像素点数量的一半，则中值为对应值。
3. 向右移动时从直方图中减去左侧的像素点，并将右侧的像素点添加至直方图中。

思考了一下，原来的算法时间复杂度为 $O(n^2(m^2 + m \log m))$ ，换用这种算法则可以降低至 $O(n^2m)$ 。因此速度会变快不少。除此以外，原来的算法要分配空间并排序，这个操作的常数较大，也是原来算法慢的一个原因。

2.4 均值处理

采用跟上面的方法类似逻辑：

1. 每行最开始统计范围内像素点的和。
2. 向右移动时从和中减去左侧的像素点之和，并将右侧的像素点之和添加和中。
均值处理后的值就是当前和除以范围内像素点的数量。

2.5 自适应中值处理

自适应中值的实现实际上就是使得中值处理的窗口可以变化。设置有最大的窗口尺寸以及最小的窗口尺寸。

- 如果当前窗口内灰度值的中值介于最大灰度值与最小灰度值之间：
 - 当前像素也介于最大灰度值与最小灰度值之间，则保持不变。
 - 否则，输出中值。
- 否则，增大窗口大小重新进行，直到当前窗口已是最大。

2.6 Non-Local Means

Non-Local Means 处理后的值由以下公式给出：

$$\text{NLMeans}(A) = \sum w(A, B) \times I(B)$$

其中：

$$w(A, B) = \frac{1}{\text{sum}} e^{-\frac{\text{MSE}(A, B)}{h^2}}$$

$$\text{sum} = \sum e^{-\frac{\text{MSE}(A, B)}{h^2}}$$

$$\text{MSE}(A, B) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A(i, j) - B(i, j))^2$$

其中 A, B 分别是当前点的邻域块以及搜索窗口内其他点的邻域块，并且邻域块的大小为 $m \times n$ 。

Non-Local Means 的计算较多，因此生成图像的速度相对会比较慢一些。生成 512 大小的图像耗时大概是 10 秒左右。

3 结果与分析

3.1 第一部分

构建一个灰度值全是 1 的灰度图像并加入一定方差的高斯白噪声，并生成其傅里叶变换图像：

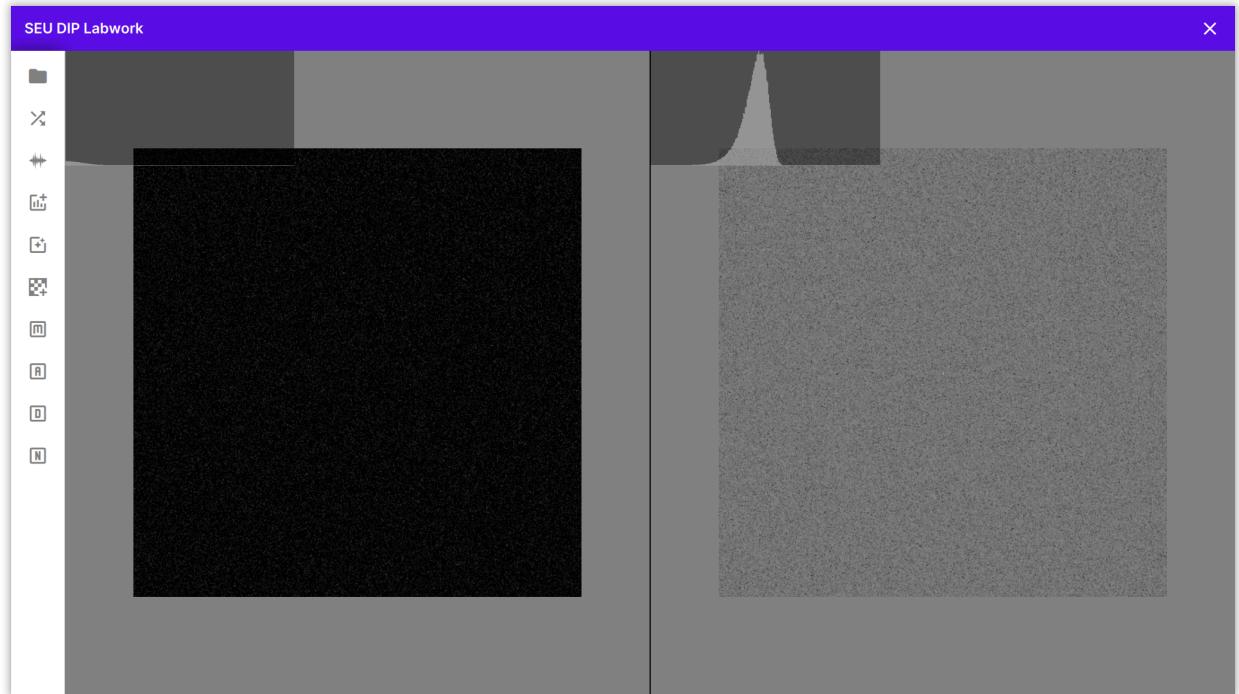


图 1: 左侧为添加高斯白噪声之后的图像，右侧是对应的傅里叶变换图像

可以看到，右侧傅里叶变换后的图像灰度比较均匀（直方图的峰值比较集中），频谱图接近水平。

我个人对于 Non-Local Means 算法的中心思想的理解与总结如下：它主要是利用了整幅图像进行降噪，以图像块为单位寻找相似区域并求平均，从而能够较好的去除均值为 0 的高斯白噪声。

接下来对 Lena 灰度图像加入高斯噪声并进行中值处理、均值处理、自适应中值处理以及 Non-Local Means 处理：



图 2: 左侧为添加高斯白噪声之后的 Lena 图像，右侧是中值处理后的图像



图 3: 左侧为添加高斯白噪声之后的 Lena 图像，右侧是均值处理后的图像



图 4: 左侧为添加高斯白噪声之后的 Lena 图像，右侧是自适应中值处理后的图像



图 5: 左侧为添加高斯白噪声之后的 Lena 图像, 右侧是 Non-Local Means 处理后的图像

可以看到, 中值处理、均值处理以及自适应中值处理对于高斯白噪声的降噪效果并不是很理想, 而 Non-Local Means 方法可以有效的去除高斯白噪声。

3.2 第二部分

对 Lena 图像加入椒盐噪声并进行中值、均值、自适应中值处理以及 Non-Local Means 处理:



图 6: 左侧为添加椒盐噪声之后的 Lena 图像, 右侧是中值处理后的图像

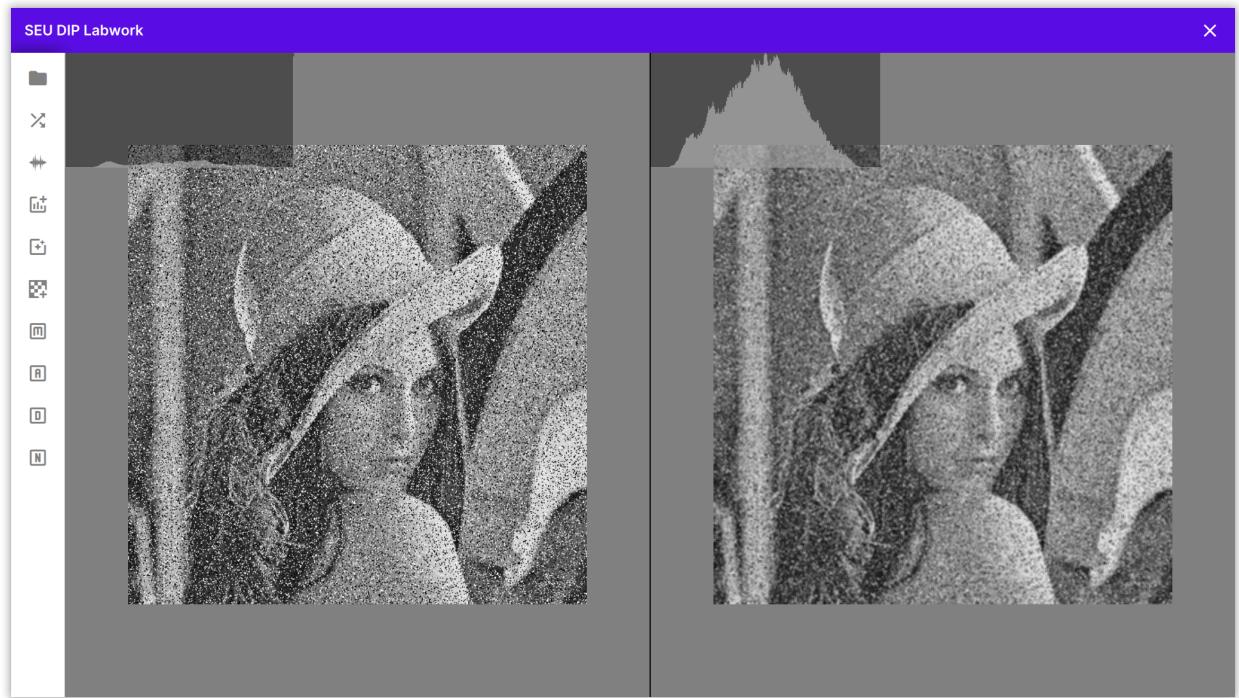


图 7: 左侧为添加椒盐噪声之后的 Lena 图像, 右侧是均值处理后的图像



图 8: 左侧为添加椒盐噪声之后的 Lena 图像, 右侧是自适应中值处理后的图像

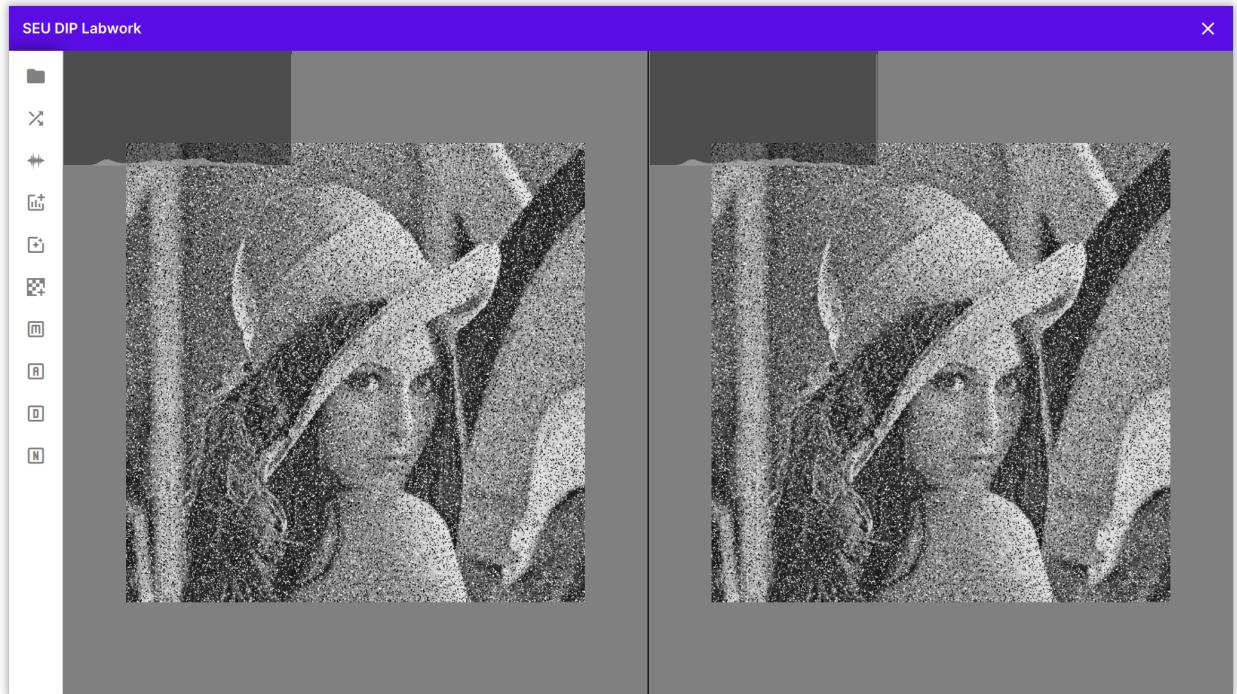


图 9: 左侧为添加椒盐噪声之后的 Lena 图像, 右侧是 Non-Local Means 处理后的图像

可以看到, 均值处理与 Non-Local Means 对于椒盐噪声的去除效果并不是很理想, 而中值处理以及自适应中值处理有效的去除了椒盐噪声。其中, 自适应中值处理较好的保留了图像的细节部分。