

图像处理中的数学方法第一次作业报告



安捷 1601210097
2017 年 3 月 17 日

为了实现本次作业的需求，我基于 MATLAB，实现了 ADMM 算法，利用迭代方法实现了去模糊 + 去噪的图像重建任务。下面从算法简述，参数设置，数值实验结果，结论四个方面总结本次作业。

1 算法简述

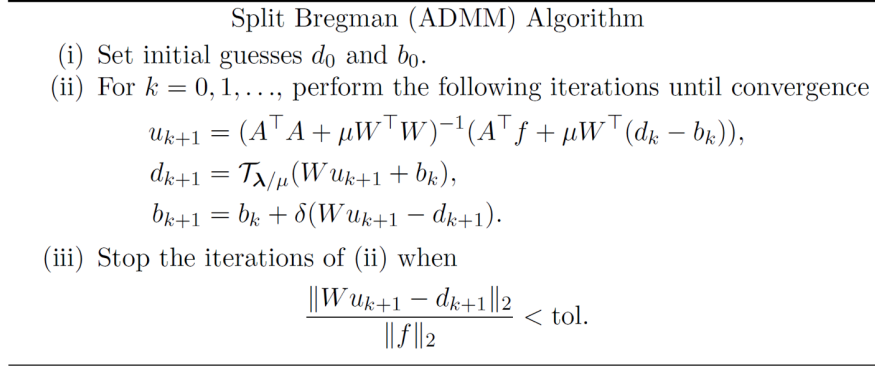


图 1: 算法流程

其中，主要问题在于第一步，在这次作业的开始，我尝试使用线性化的思路来实现本次作业，即，将 A 算子与 W 算子实现的操作转化为一个矩阵，经过线性化之后，上述流程中的操作全部转化为矩阵与向量的操作，这一操作可行，但是存在很严重的两个问题：

1. 存储困难，线性化的操作使得矩阵 A 与 W 的维数是原图像的平方，即 128×128 的图像需要的双精度存储量为约 1200MB， 512×512 的图像需要的双精度存储量为约 80GB，如果不采取稀疏矩阵存储方式，几乎不可存储；
2. 计算耗时长，在我使用的基于 NVIDIA CUDA 的 GPU 加速技术之后，依然速度很慢，主要原因在于矩阵求逆操作到目前为止没有效率很高的快速计算方法，即使采用 GPU 加速技术；

基于以上两点原因，我放弃了这一方法，转为采用老师上课提到的利用 Fourier Transform 求解 Poisson 方程的算法来求解 ADMM 算法的第一步。下面将算法简述如下：首先，原问题可以表述为：

$$(A^T A - \mu W^T W) u = M \quad (1)$$

即有：

$$A * A * u - \mu L * u = M \quad (2)$$

其中, L 是拉普拉斯卷积核, 上式经过 FFT 之后, 有:

$$\hat{A} * \hat{A} * \hat{u} - \mu \hat{L} * \hat{u} = \hat{M} \quad (3)$$

即有:

$$\hat{u} = \frac{\hat{M}}{\hat{A} * \hat{A} - \mu \hat{L}} \quad (4)$$

$$u = \text{ifft2} \left(\frac{\text{fft2}(M)}{\hat{A} * \hat{A} - \mu \hat{L}} \right) \quad (5)$$

上式即我在程序中实际应用的 ADMM 第一步求解方法。

2 参数设置

参数名称	参数值
KERNEL_SIZE	15
SHIFT_LEN	(KERNEL_SIZE - 1) / 2
SIGMA	1.5/2
SCALE	100/200
MU	0.3
LAMBDA	0.06
TAU	LAMBDA / MU
TOL	1e-3
DELTA	0.12
MAX_ITERATION	300

表 1: 参数设置

3 数值实验结果

针对 $SIGMA = 1.5/2.0$ 及 $SCALE = 100/200$ 两组参数共四种情况, 我分别进行了数值实验, 其余参数均按照上一节的参数表设置, 选取 512*512 lena 图进行数值实验, 结果图示如下, 其中 original image 为原始未经任何处理的图像, filtered image 为按照作业要求, 进行高斯平滑处理, 并添加正太随机噪声的图像, reconstruction image 为经过 ADMM 算法恢复之后的图像, error 为按照作业要求进行误差度量得到的每次迭代误差分布:

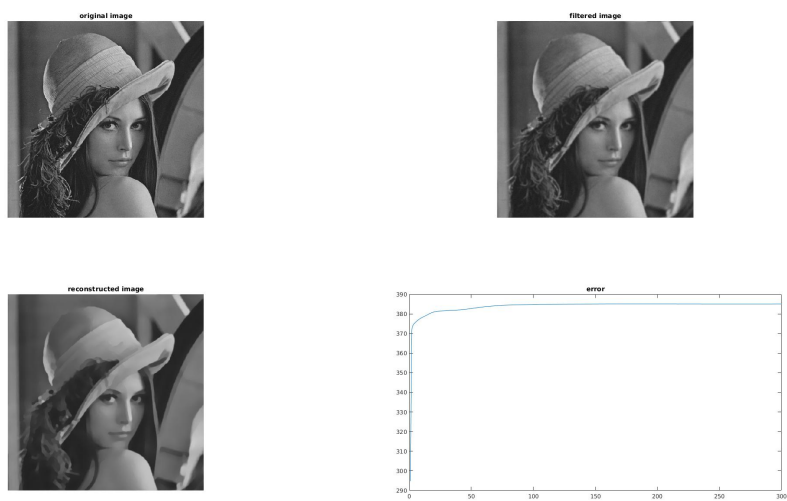


图 2: SIGMA=1.5, SCALE=100 数值实验结果

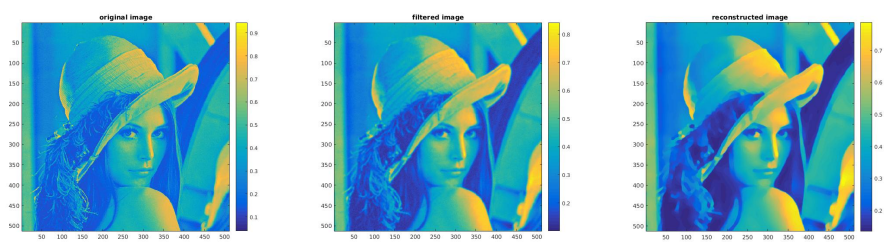


图 3: SIGMA=1.5, SCALE=100 数值实验结果

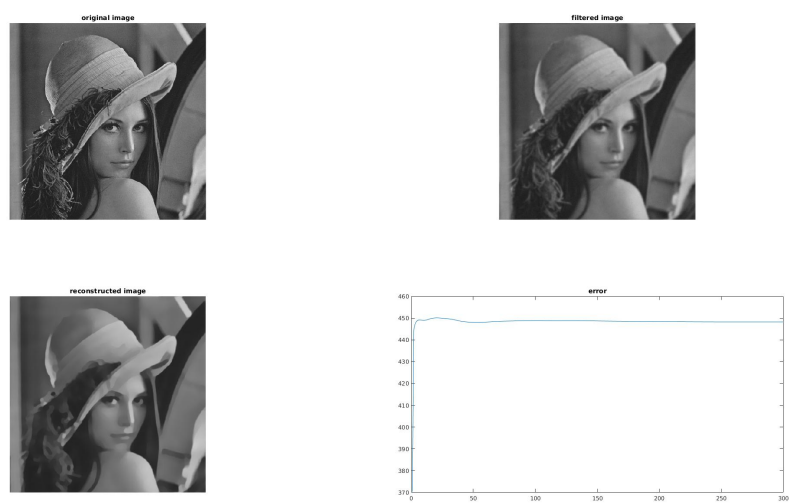


图 4: SIGMA=2.0, SCALE=100 数值实验结果

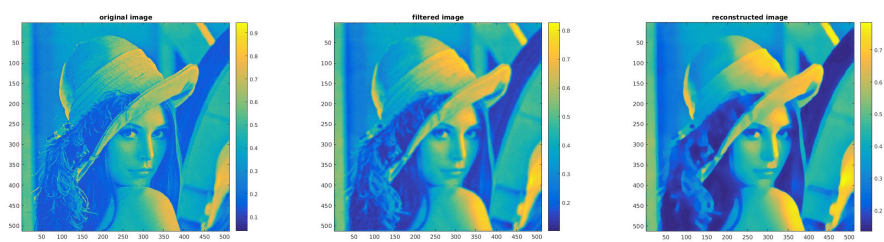


图 5: SIGMA=2.0, SCALE=100 数值实验结果

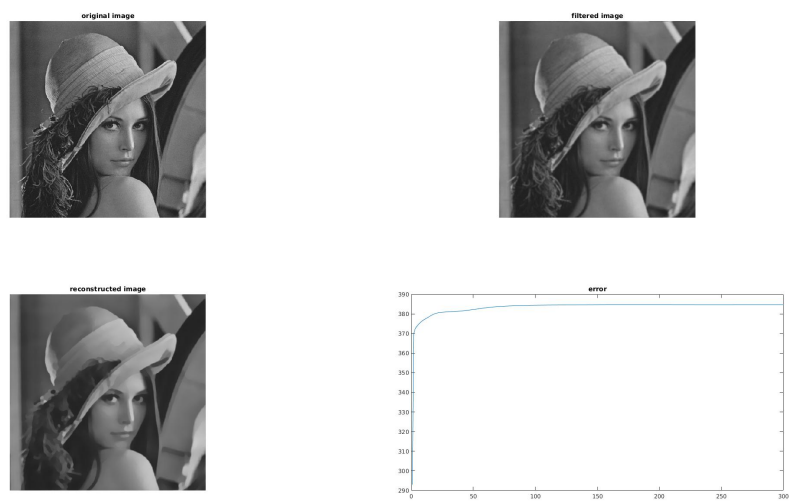


图 6: $\text{SIGMA}=1.5$, $\text{SCALE}=200$ 数值实验结果

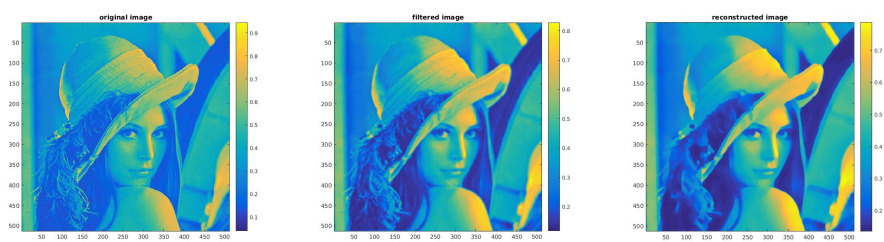


图 7: $\text{SIGMA}=1.5$, $\text{SCALE}=200$ 数值实验结果

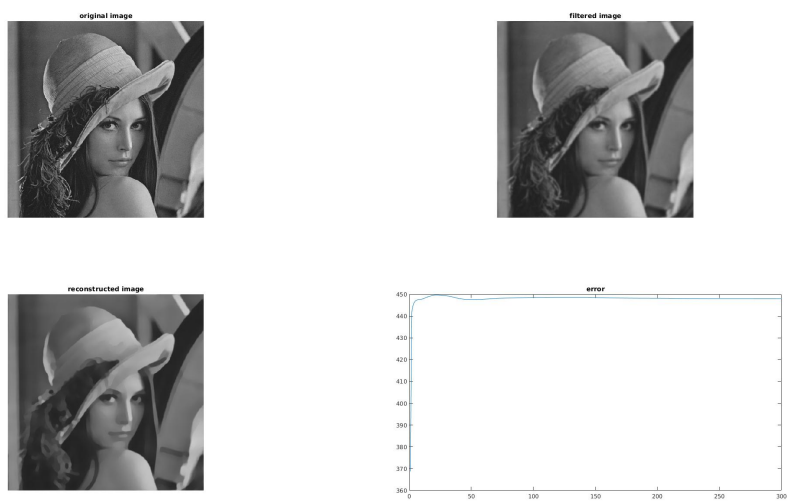


图 8: SIGMA=2.0, SCALE=200 数值实验结果

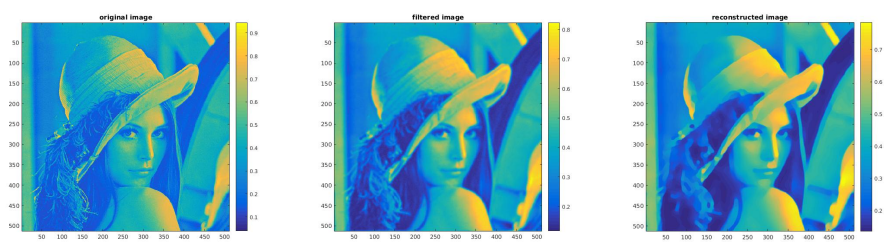


图 9: SIGMA=2.0, SCALE=200 数值实验结果

4 结论

1. 从数值实验的结果来看，SIGMA 的取值直接影响到了最终的误差结果，SIGMA=2.0 时的误差显著大于 SIGMA=1.5 时的情形，与此同时，SCALE 对最终误差的结果影响不大；
2. 每一次数值实验的误差都经历了先增后降或先增后稳定的过程，这是由于我将 filtered image 设置为了迭代初始值，由此可以发现，从信息论的角度，ADMM 算法的迭代过程并没有增加信息量，反而损失了图像的信息，这与图像处理的本质，即：“图像处理不会增加图像的信息量，只会使得图像中的一部分信息更为显著”相一致；
3. 从恢复之后的图像可以看出，恢复之后的图像明显在皮肤、帽子、背景等平滑处均优于 filtered image，在边缘处也起到了很好的保留边缘的作用，这与我们在算法中进行正则约束的目的是一致的，即算法实现达到了算法的设计目的；
4. 算法的处理使得图像的信噪比提高，对比度显著增强，这是 ADMM 算法的优势；
5. 从图像中头发区域的恢复情况可以看出，这一算法没有办法对原本已经丢失的细节进行恢复，这与我们的直观认识是一致的，即图像处理不会凭空产生细节，这是这一算法值得改进的地方；