

《数字与图像处理》作业 04

作业报告

自 62 2016011411 姜志强

一、图像去噪

1.1 原理介绍:

本次作业我使用空域滤波来进行图像去噪,空域滤波通过对像素点的领域进行某种计算,然后用计算的结果来代替该像素点的值从而实现噪声的去除。

本次作业我尝试了多种滤波器,并选出处理效果比较有代表性的四种进行说明:

均值滤波器:对领域内像素值求期望来代替原有像素点,可以使图像像素值之间更加相近,从而减少尖锐变化,但同时也会增加模糊度。

中值滤波器:用领域内的灰度值的中值来代替原有像素点,可以去除领域内的像素点极值来去除噪声,并且模糊程度低于均值滤波器,但是计算量大,在搜寻相关资料后,找到了一种基于直方图的快速中值滤波,在对一行像素进行中值滤波时,每两次滤波之间的像素邻域有大量重复点,该算法的核心思想就是利用直方图来储存上一次中值滤波后这些重复点的顺序从而减少计算量,经验证在 matlab 上速率提高三倍。

高斯滤波器:利用高斯函数生成滤波模板,可以有效抑制正态分布的噪声,并且在方差与模板大小的合适的情况下模糊程度很低。

自适应局部降噪滤波器:这种滤波器利用局部统计量来进行滤波,计算公式为

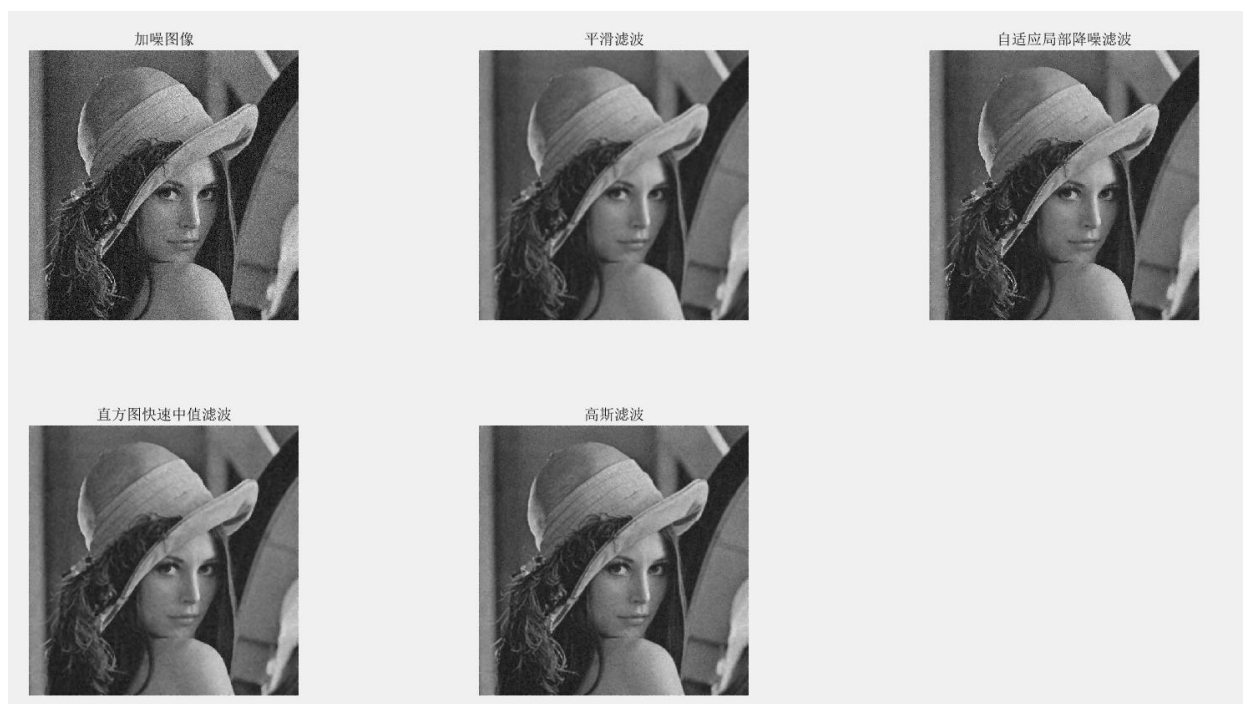
$$f(x, y) = g(x, y) - \frac{\sigma_{\eta}}{\sigma} (g(x, y) - \mu), \text{ 其中 } \sigma、\mu \text{ 为邻域的方差与均值, } \sigma_{\eta} \text{ 为噪声的}$$

方差,对于平滑部分,可以认为邻域方差与噪声方差接近,这样就近似于均值滤波,边缘部分可以认为方差较大,故而接近于原值,这样就实现了去噪并在一定程度上保留边缘,但是噪声的方差需要得知,我通过调参确定了比较合适的值。

除此之外还有自适应中值滤波器以及最大值滤波器但是效果不佳且没有对比性,故没有进行说明,所有滤波器实现代码均在 mylvbo.m 文件中实现。

1.2 计算结果及分析:

单一滤波器处理效果如下：



由于图像在 word 中缩放难以看出效果，可运行 `lvboim.m` 脚本来观察效果。

总体来讲对噪声抑制能力最强的是均值滤波，但是也最模糊，高斯滤波与中值滤波保留了边缘，但是噪声依然比较明显，特别是在背景和肌肤处，可以看见深浅不均匀，这样的效果也合乎滤波器设计原理。

而自适应滤波器则比较奇怪，滤波之后的图像某些部分效果很好，边缘保留十分清晰且比较均匀，比如 `lena` 的头发，但是有些地方又是成块的噪声，总体来说处理后的图像有斑点状的噪声，这也比较好解释，图像每一像素的邻域的噪声的方差不一定是我们所设的噪声的方差，这就导致这些地方噪声残留，而那些与所设噪声方差接近的非边缘地方则相当于平滑滤波，这就形成了最后的斑点状效果，边缘处局部方差大，使得返回值接近与原像素值，从而很好地保留了边缘。

为了可以客观评价去噪效果，我从网上找来的 `lena` 原图，计算各图像与原图的 PSNR、SSIM 来判断，不过作业所给图片是对原图进行截取所得，因此这些指标仅作参考。

	未去噪图像	均值滤波	中值滤波	高斯滤波	自适应
PSNR	13.07	13.58	13.51	13.50	13.40
SSIM	0.45	0.4883	0.4852	0.4819	0.4845

相对于未去噪图像，经过各种滤波之后的图的 PSNR 以及 SSIM 均有所提

高，说明噪声的确得到一定程度抑制，均值滤波的 PSNR 以及 SSIM 最高，均值滤波直接求平均去噪，抑制噪声能力自然很强，但是模糊。

这些指标仅做参考，单就滤波效果来看，高斯滤波器是最好的，中值滤波器次之，但是噪声还是比较明显，且参数难以调整，模板变大一些或者方差调大一些就会导致图像模糊，因此我将自适应局部降噪滤波器的输出的带有斑状噪声的图像进行中值滤波（进行高斯滤波效果也较好），这样一来边缘有较好的保留而来斑状噪声比原图的噪声密度低，滤波后深浅不一的状况会改善，且自适应滤波器的输出图像中没有斑状噪声的部分效果很好，这样最终得出的图像边缘得到了保留且噪声有效的去除，相对于单一滤波器效果更好。

得到的图像如下：



如上，右图的麻点感已经明显去除，且肌肤背景处比较平滑，边缘有较好的保留，虽然依然有一些模糊感，但是整体效果较好。

二、边缘检测

2.1 原理介绍：

边缘检测首先就是图像去噪，经过测试后选择高斯滤波比较合适，计算量小且边缘影响小，均值滤波让边缘变粗，而中值和自适应滤波则计算量大且滤除噪声有些不均匀，由于采用了高斯滤波后的图像，然后主要是利用高通滤波来提取梯度变化大的地方从而找到边缘，尝试使用了一阶微分滤波器，二阶微分滤波器：
一阶微分滤波器：尝试了 robert 以及 sobel 模板，robert 算子使用 2×2 模板来提

取 x 、 y 方向的梯度,不过模板较小对梯度不敏感,sobel 则使用 3×3 模板来提取,对梯度十分敏感,对于噪声却也更敏感,对于高通滤波后的图像需要加阈值 N ,即如果 $f(x,y) < N$ 则令 $f(x,y) = 0$,这样可以去除大部分噪声所形成的梯度,robert 所取的阈值为 25,且进行了二值化,否则看不见细细的边缘,sobel 取阈值为 100,没有进行二值化。

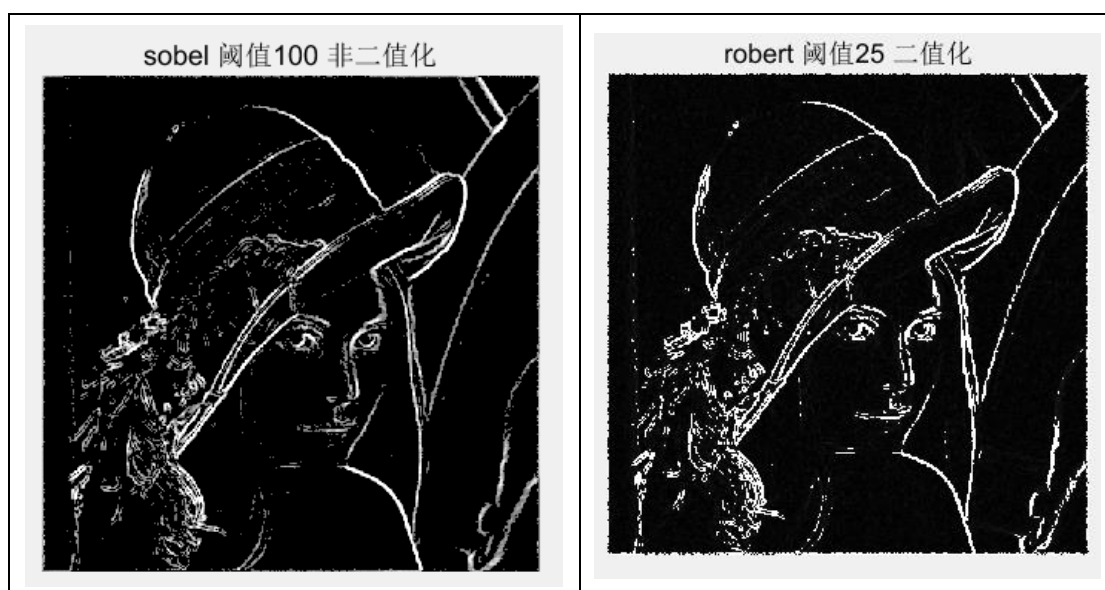
二阶微分滤波器: 尝试了 laplace 算子,它是一种二阶微分算子,可以突出小细节,但是对于噪声的响应很强,单纯加阈值也能看见背景的噪声点,因而不使用单纯加阈值的方法,而是用 N 来代替图像中所有小于 N 的值,并将此图像标定到 0 到 255 之间,这样可以让部分噪声弱化,但是依然看得见噪声点。

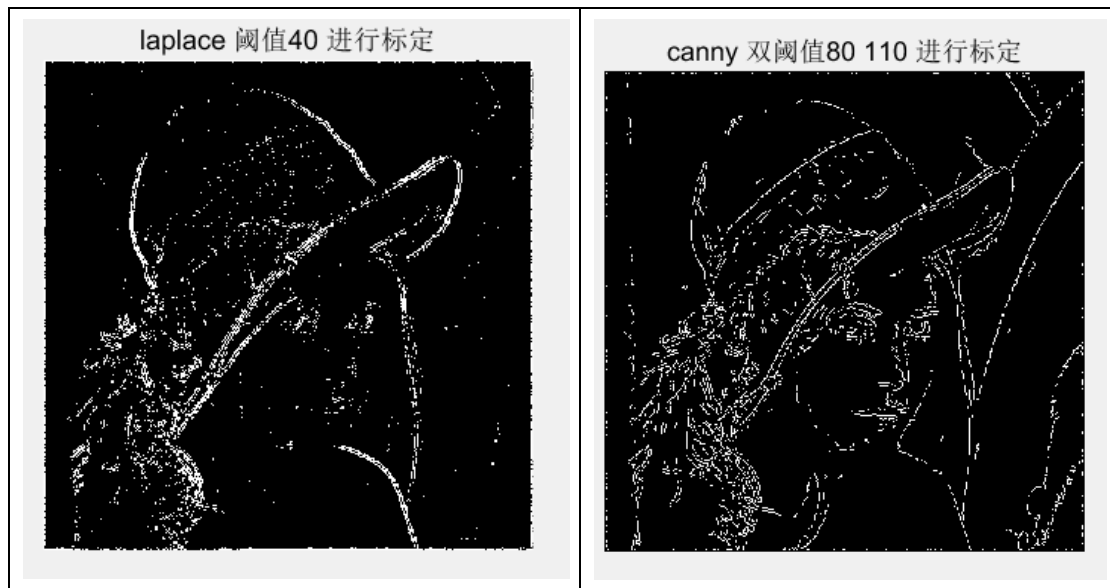
此外,我还参考课本第十章实现了 canny 算法,该算法在高通滤波基础上再进行非极大值抑制以及双阈值,非极大值抑制通过剔除梯度方向上的非极大值来使边缘细化,而双阈值实现了边缘连接,小于低阈值的点认为是噪声,赋值为 0,高于高阈值的点认为是边缘,高低阈值之间的点若与高阈值点相连通则认为是边缘否则认为是噪声,这样可以留下边缘中梯度较小的部分,此部分代码是利用 DFS 来实现。

以上均在 `edge_detect` 内实现。

2.2 计算结果及分析:

处理得到的边缘如下:





可以看见，噪声最大的明显是 laplace 的边缘检测，sobel 与 robert 的边缘检测效果都不错，不过 robert 需要进行二值化才能看见清晰的边缘，否则图像很暗，边缘不清晰，sobel 相比于 robert 提取了更多的细节，不过需要更高的阈值以去掉大部分噪声，否则边缘在噪声白点中难以观察，这些与之前的理论分析一致。

canny 算法，对边缘的提取是最好的，它提取出了其他三种方法难以提取的边缘：图片左侧的竖线，lena 帽檐上边的三角以及 lena 的左半边脸的边缘，且边缘很细。噪声也基本消除，但是看起来效果却不是很好，这主要是因为它提取的边缘断断续续，通过分析我认为，这与我的非极大值抑制算法有关，我的非极大值抑制算法为了方便，只给出 4 个方向的梯度，每个像素点选择最接近的梯度方向进行抑制，而这样不是严格的，可能导致某些边缘由于梯度方向偏差过大而被抑制，应当利用插值来判断各个方向的梯度是否应该被抑制，此外还与阈值选取有一定关系。

总体来讲，个人认为本次作业边缘检测的效果较好的是 sobel 加阈处理，自己写的 canny 需进行改进和调参以解决边缘的断续问题，laplace 则需要滤除噪声效果较强的滤波器。

三、小结

本次作业让我了解了不同空域滤波方法的特性以及其适用范围，大部分情况需要结合不同滤波器混合使用才能有好的滤波效果，此外还需要考虑时间以及实用性，而且图像处理还是得看最终的图片处理效果，以此为依据改进算法。

[ReadMe 内有文件说明以及程序运行说明](#)