

编程实现小波域维纳滤波

图像小波变换:

python 里面的 pywavelets 包进行的变换是将图像分解成了四个部分, 具体是 Approximation, horizontal detail, vertical detail and diagonal detail coefficients respectively. 其中第一部分对应的是低频部分, 其余的是高频信号, 第二个是水平部分, 第三个是垂直方向的, 最后一部分是对角线形式的分解。

维纳滤波:

维纳滤波类似于求解一个最大后验概率, 这里对于后验概率中的概率估计都是以高斯模型的形式进行估计的, 最后使用的是求导数的方法求得最大的原来的图像  $X_i$

其中, 对于有噪声的部分的条件概率的方差的估计, 是使用的对于 HH 部分, 即 diagonal detail coefficients 部分的数值进行估计:

$$\sigma_n = \frac{\text{median}(|HH|)}{0.6745}$$

然后对于实际的想要滤波得出来的结果的  $X_i$  的方差的估计使用输入的有噪声的图像  $Y(i)$  进行估计:

$$\sigma^2 = \frac{1}{M} \sum_i Y(i)^2 - \sigma_n^2$$

小波域的维纳滤波:

现将图像进行小波分解, 对于高频部分进行维纳滤波处理, 最后再进行反变换变回原来的小波图像, 经过了一次分解之后的情况如下:

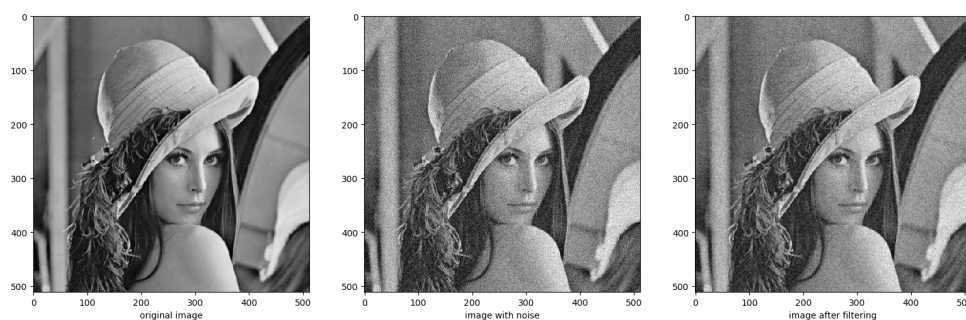


图 1: 只经过一次分解, 然后进行小波域维纳滤波的图像

这样效果实际上并不好, 从图中可以比较直观的看出来, 所以我参考网上其他人写的维纳滤波算法, 尝试进行多尺度的金字塔形式的小波分解, 然后对于每一尺度进行维纳滤波, 最后组合起来, 参考网上资料, 这里使用的是类似于 PPT 中提到的 Mallat 算法类似金字塔结构的小波分解, 然后使用递归的思路进行维纳滤波, 效果如下 (5 层的小波分解):

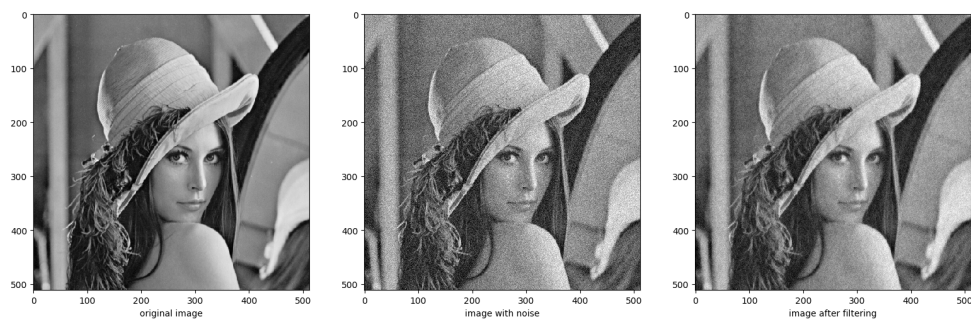


图 2: 只经过一次分解, 然后进行小波域维纳滤波的图像

可以看到, 效果实际上好了一点, 细看的话, 图像实际上是变模糊的, 但是确实相对而言起到了滤波的效果。

附录:

代码 waveletwiener.py 直接运行就可以

需要环境变量:

numpy 1.16.5

opencv-python 4.1.1.26

matplotlib 3.1.1

pywavelets 1.1.1