

2019 年秋季《图像处理与分析》编程作业-05

【第十三讲-小波域图像去噪】

姓名: 学号:

【问题描述】

以 lena 图像为例, 编程实现小波域维纳滤波(具体算法见十三讲 ppt)小波变换。可以使用 matlab 子带的 dwt2。

问题分析:

(1) 小波域统计分布

图像的小波分解是分别对图片的行和列进行小波变换, 分别对行和列进行低通和高通的处理后, 然后分别对行和列进行采样实现 $1/2$ 压缩, 再重复一次相同的操作即可得到四张子图, 这就完成了一次小波分解, 其流程如下图所示:

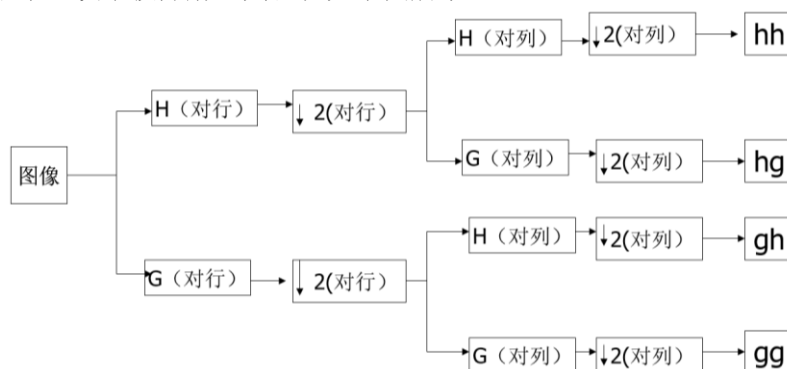


图 1.小波分解原理图

重复进行多次上述操作可以得到多次小波分解, 得到图像的金字塔分解。在图像的小波域中, 除了行列都是低通得到的子图以外, 其他的子图直方图都呈现相似的统计分布。在分析过程中可以考虑高斯模型、广义高斯模型等模型构建其分布, 为了简便计算可以使用高斯分布模型去处理图像小波域分布。示意图如下:

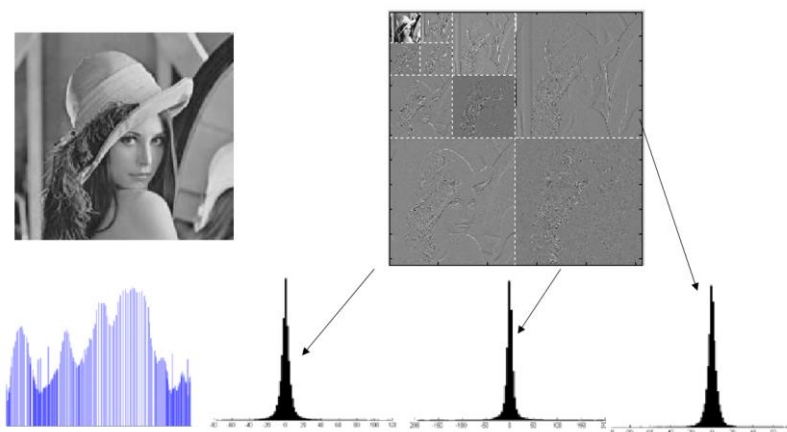


图 2.小波域对的统计特征

(2) 小波域图像维纳滤波原理

图像观测模型为 $Y=X+N$ ，其中 X 为无噪声的原始图像， N 为噪声， Y 为观测到的有噪声图像。

设小波域的先验模型是高斯分布。假定小波域中的概率分布满足下列各式： $P(X_i) \propto \exp(-X_i^2 / \sigma^2)$, $P(Y | X) = P(N)$, $P(N_i) \propto \exp(-N_i^2 / \sigma_n^2)$ ，那么对 X 的估计值为： $\hat{X} = \arg \min \{-\log P(Y | X) - \log P(X)\}$ 。假设图像在小波域内点之间相互独立，可以直接在点上进行处理，则有： $\hat{X}_i = \arg \min \{(Y_i - X_i)^2 / \sigma_n^2 + X_i^2 / \sigma^2\}$ ，令函数 $F(X_i)$ 为： $F(X_i) = (Y_i - X_i)^2 / \sigma_n^2 + X_i^2 / \sigma^2$ ，令其关于 X_i 的偏导为 0，得到：

$$0 = \frac{\partial F}{\partial X_i} = 2(X_i - Y_i) / \sigma_n^2 + 2X_i / \sigma^2 \Rightarrow X_i = \frac{\sigma^2}{\sigma^2 + \sigma_n^2} Y_i$$

对模型中的方差进行估计，取行列均为高通的子图 HH 的小波域中值作为 σ_n ，得到：

$$\sigma = \frac{\text{median}(|HH|)}{0.6745}, \sigma = \frac{1}{M} \sum_i Y(i)^2 - \sigma_n^2$$

将得到的方差 σ_n^2 和 σ^2 对小波域的图像进行映射变换（除了双低通的子图），最后在逆变换会原始图像空间，则完成了小波域的图像去噪。以上就是实验中的图像去噪原理。

实现细节及代码展示：

- 编程语言：Python3.7+
- 所用模块：Numpy1.17.2（用于矩阵处理），PIL5.3.0（用于图像读取等基本操作），matplotlib3.0.2（用于绘制图像等），pywt1.0.1（调用小波变换函数）

(1) 维纳滤波函数

相关变量：

Sigma_n: 噪声方差 σ_n^2 平方根

Sigma: 子图的方差 σ^2

X: 去噪后的小波域子图 $X = \frac{\sigma^2}{\sigma^2 + \sigma_n^2} Y$

M: 小波域子图的像素点总数，及高度×宽度（m×n）

```
1 import numpy as np
2 import pywt
3 from PIL import Image
4 import matplotlib.pyplot as plt
5 def weina(Y, HH): #Y为小波域图像，HH为双高通子图
6     sigma_n=np.median(np.abs(HH))/0.6745
7     X=np.copy(Y)
8     m,n=Y.shape
9     M=m*n
10    sigma2=np.sum(Y**2)/M-sigma_n**2
11    X=Y*sigma2/(sigma2+sigma_n**2)
12    return X
```

(2) 自编写的高斯噪声生成函数:

```

14 def gen_gaussian_noise(signal,SNR):
15     """
16     :param signal: 原始信号
17     :param SNR: 添加噪声的信噪比
18     :return: 生成的噪声
19     """
20     noise=np.random.randn(*signal.shape) # *signal.shape 获取样本序列的尺寸
21     noise=noise-np.mean(noise)
22     signal_power=(1/signal.shape[0])*np.sum(np.power(signal,2))
23     noise_variance=signal_power/np.power(10,(SNR/10))
24     noise=(np.sqrt(noise_variance)/np.std(noise))*noise
25     noise=noise.reshape((signal.shape[0],signal.shape[1]))#噪声序列reshape成与图形长宽相同的矩阵
26     return noise+signal

```

(3) 读入图片 lena.tif 加入噪声并在小波域去噪后重建回原始图像域

相关变量:

X: 输入图像

Img: 根据指定信噪比加入高斯噪声后的图像

cA, (cH, cV, cD): 一次小波变换得到的四张子图

cA2, (cH2, cV2, cD2), (cH1, cV1, cD1): 两次小波变换得到的七张子图

CA3, (CH3, CV3, CD3) (CH2, CV2, CD2), (CH1, CV1, CD1): 三次小波变换子图

w1, w2, w3: 分别为经过一、二、三次小波分解去噪后得到的图像

```

28 X = np.array(Image.open("assignment01_images/lena.tif"))
29 img=gen_gaussian_noise(X,10)#添加高斯噪声, 指定信噪比
30 #一次小波分解进行维纳滤波
31 cA, (cH, cV, cD)=pywt.dwt2(img, 'haar')
32 cH=weina(cH, cD)
33 cV=weina(cV, cD)
34 cD=weina(cD, cD)
35 w1=pywt.idwt2((cA, (cH, cV, cD)), 'haar')#重建
36 #两次小波分解进行维纳滤波
37 cA1, (cH1, cV1, cD1)=pywt.dwt2(img, 'haar')
38 cA2, (cH2, cV2, cD2)=pywt.dwt2(cA1, 'haar')
39 cH1=weina(cH1, cD1)
40 cV1=weina(cV1, cD1)
41 cD1=weina(cD1, cD1)
42 cH2=weina(cH2, cD1)
43 cV2=weina(cV2, cD1)
44 cD2=weina(cD2, cD1)
45 w2_1=pywt.idwt2((cA2, (cH2, cV2, cD2)), 'haar')
46 w2=pywt.idwt2((w2_1, (cH1, cV1, cD1)), 'haar')#重建
47 #三次小波分解进行维纳滤波
48 CA1, (CH1, CV1, CD1)=pywt.dwt2(img, 'haar')
49 CA2, (CH2, CV2, CD2)=pywt.dwt2(CA1, 'haar')
50 CA3, (CH3, CV3, CD3)=pywt.dwt2(CA2, 'haar')
51 CH3=weina(CH3, CD1)
52 CV3=weina(CV3, CD1)
53 CD3=weina(CD3, CD1)
54 CH2=weina(CH2, CD1)
55 CV2=weina(CV2, CD1)
56 CD2=weina(CD2, CD1)
57 CH1=weina(CH1, CD1)
58 CV1=weina(CV1, CD1)
59 CD1=weina(CD1, CD1)
60 w3_1=pywt.idwt2((CA3, (CH3, CV3, CD3)), 'haar')
61 w3_2=pywt.idwt2((w3_1, (CH2, CV2, CD2)), 'haar')
62 w3=pywt.idwt2((w3_2, (CH1, CV1, CD1)), 'haar')#重建

```

效果展示:

根据上述编写程序，运行 `Wiener_wavelet.py` 文件，可以得到如下图：



图 3.小波域维纳滤波去噪结果图（信噪比 20）

图中（a）为原始图像，（b）为加入高斯噪声的图像（这里信噪比设置为 20），（c）（d）（e）分别为进行 1、2、3 次小波分解后维纳滤波去噪的结果。可以发现三次去噪结果均肉眼可见地改善了图片质量，且第 3 次的效果明显好于前两次去噪。这是因为多次小波分解得到的子图数目增多，可以得到更细致的小波域，滤除的噪声成分也更多一些，所以最终的去噪效果更好。为了更突显这种差别，又选择了信噪比设置为 10 的图像进行了去噪，结果如下图所示：



图 4.小波域维纳滤波去噪结果图（信噪比 10）

在这次去噪结果中，由于噪声较大，所以去噪效果整体不够理想，但是可以更加明显地发现 3 次小波分解去噪的效果在某种程度上较大地还原了原始图像的样貌。但是显然噪声过大使得去噪后的图片在局部上出现了“格点”、分辨率不够高，所以对这种信噪比较低的图片还应当对去噪算法进行调整。