**基于卷积和池化的数字图像滤波及图像梯度提取**

李昀哲1

1(上海大学计算机科学与工程学院 上海)

**摘要:**

**[目的]**分析有效的图像滤波方式处理含有不同噪音的数字图像。

**[方法]**根据图像滤波的数学原理，自定义中值、均值滤波，对含有不同噪音的图像进行处理，观察经过处理后，不同滤波后图像的频谱图。

**[结果]**60%密度以上，对于椒盐噪音滤波效果都较差。中值滤波对于椒盐噪音的处理效果较好，均值滤波对高斯噪音处理效果较好，两种滤波对于乘性噪音的处理效果相当。

**[局限]**均值滤波具有线性性质，不能很好地保护图像细节，在图像去噪的同时也破坏了图像的细节部分；中值滤波具有非线性性质，对于脉冲噪音效果好，但对于连续噪音效果较差。

**[结论]**处理高斯噪音，均值滤波是最佳选择，处理椒盐噪音，中值滤波是最佳选择。

**关键词:** 中值滤波；均值滤波; Sobel算子

**Self-defined filters to process digital images with noise based on convolution and pooling, and get gradients of images**

Li Yunzhe1

1(College of Computer Engineering and Science, Shanghai University, Shanghai, China)

**Abstract：**

**[Objective]**To investigate the best way to reduce noise on images.

**[Methods]**According to the mathematics principles, define functions of median and mean filters to process images and get result on frequency domain.

**[Results]**When the density is above 60%, both filters have low performance for salt&pepper noise. Median filter is better than mean filter for Gaussian noise, while mean filter is better than median filter for Salt&pepper noise.

**[Limitations]**On the reason of linearity, mean filter can’t protect details on the images when reduce noise. While median filter has the nonlinear property, it’s better to process impulse.

**[Conclusions]**To deal with Gaussian noise, mean filter is the best way. To deal with Salt&Pepper noise, median filter is the best way. Different filters should be applied according to different situations, for each filters has merits and demerits.

**Keywords：** Median filter;Mean filter;Sobel operator

**1 图像滤波的现实意义**

图像，对于计算机视觉来说是重要的研究对象，通过对图像的精密处理，可以应用于，如：人脸识别、自动驾驶等场景。但在图像在获取过程中，可能会因环境因素的干扰，例如：光照、雨水等的影响，使我们实际得到的图像和想要得到的图像存在偏差，这种偏差称之为“噪音”，噪音使得对图像后续的处理产生困难。

噪声是或大或小的极值点，作用于图像像素的真实灰度值上，造成亮、暗点的干扰[1]，影响图像的质量。

滤波的作用就在于尽可能的去除因外界因素产生的噪音，还原原始图像，以便后续的处理。

**2 滤波数学原理**

**2.1 利用卷积实现均值池化**

在处理之前，需要将原图像转换为灰度图。对于灰度图而言，是一个二维矩阵。对于原始数据的修正就需要借助其周围的数据进行分析，通过分析需要处理的像素点上下左右、左上左下、右上右下八个点的像素值进行卷积操作来修正，因此使用“加权系数”矩阵，如图1所示：

|  |  |  |
| --- | --- | --- |
| １ | １ | １ |
| １ | １ | １ |
| １ | １ | １ |

　　　　　图１加权系数矩阵

Fig1 Weighted coefficient matrix

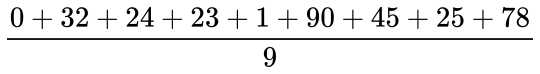
对于一个8x8的矩阵中的某一个像素点，假设使用均值滤波，对(2,2)像素点操作，则对其周围包括自己的9个点求和后取平均，将得到的均值赋给(2,2)像素点，如图2所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 32 | 24 |  |  |  |  |  |
| 23 | 1 | 90 |  |  |  |  |  |
| 45 | 25 | 78 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

图2 对(2,2)像素点作均值池化

Fig2 To mean-pool (2,2)

通过计算



得到(2,2)像素点的值为：35.3。

利用3x3的加权系数矩阵对于边界像素点进行卷积计算时会出现越界的问题，在此前的实验中，当出现越界时，通常通过在边缘外填补“0”达到加权系数矩阵的大小，来完成卷积。

**2.2 滤波模板**

滤波模板也就是上述的“加权系数”矩阵，可以根据需求自己定义，对于边缘检测，Sobel算子就是比较著名的滤波模板，如图3所示，对于边缘而言其周围的像素点和自身都有明显的区别，这就使得加权系数的设置有所不同。

|  |  |  |
| --- | --- | --- |
| 1 | 0 | -1 |
| 2 | 0 | -2 |
| 1 | 0 | -1 |

图3 Sobel算子

Fig3 Sobel operator

**2.3 中值滤波**

中值滤波由Turky在1971年提出，最初用于时间序列分析，后来被用于图像处理，并在去噪复原中取得了较好的效果。中值滤波器是基于次序统计完成信号恢复的一种典型的非线性滤波器，其基本原理是把图像或序列中心点位置的值用该域的中值替代，具有运算简单、速度快、除噪效果好等优点[2]。

也就是说，对于一个像素点，将包含自己的周围的九个点的灰度值进行从小到大排序，将中位数作为结果填入这个像素点，如图4所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  | 0 | 76 | 12 |  |  |
|  |  |  | 87 | 56 | 34 |  |  |
|  |  |  | 21 | 50 | 99 |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

图4 对像素点的中值滤波

Fig4 To median-filter one point

排序得到：

MommyTalk1636112126787

中位数为50，即这个像素点中值滤波后的结果。

1. **Matlab实现思路**

根据上述数学原理，可以理解为对一个灰度图矩阵***A，***自定义窗口的大小，用中值和均值两种滤波方式进行处理，并在矩阵***A***上自左向右、自上向下滑动。

**3.1 图像预处理**

在进行滤波操作前，需要对读入的图像进行预处理，即，先将RGB的图像转为灰度图，应用函数：

MommyTalk1636120545646

再为图像添加不同噪音以观察后续滤波效果。

**3.2 图像尺寸处理**

由于滤波的卷积操作涉及每一个像素点及其周围的像素点，对于边界的点就要为其添加“0”，使得卷积可以正常操作。

原图像的尺寸发生了改变，其中的内容也要重新调整。

**3.3 自定义滤波函数**

3.3.1 均值滤波函数

均值滤波的本质是定义一个元素均为1的“加权系数”矩阵***M***，用***M***和***A***中的元素做卷积。

将卷积后的值求和，再取平均值，将平均值赋给中心点，进行下一点的操作。

3.3.2 中值滤波函数

中值滤波是对处理后的矩阵***M***，按照既定的窗口大小进行分割。对于每一个分割的矩阵，利用函数

MommyTalk1636120972483

将其转换为1行*n*列的矩阵，用函数

MommyTalk1636121081674

求得这些数的中位数，将中位数赋给中间值，进行下一点的操作。

**3.4 滤波后统一处理**

根据滤波后的矩阵和窗口大小，确定最终结果的数值范围，并将矩阵转换为uint8型。

**3.5 图像输出**

首先将uint8类型转换为double类型，并用函数

**MommyTalk1636121815825**

进行傅里叶变换得到频域上的表示。

再利用函数

MommyTalk1636121884833

进行频谱平移。由于得到的是复数，再进行取模的操作。最后再利用频谱对数变换

MommyTalk1636121996808

得到图像的频谱图。

**3.6 结果分析**

比较方法：综合尝试“矩阵相减”、“矩阵相除”，均不适合此场景，误差较大，结合实验7中比较矩阵相关性的方法，得到较准确的相关性比较。故使用函数

MommyTalk1636546365233

计算两个图像的相关性。

3.6.1 不同密度椒盐噪声的比较

原始图像灰度图及频谱图如图5所示

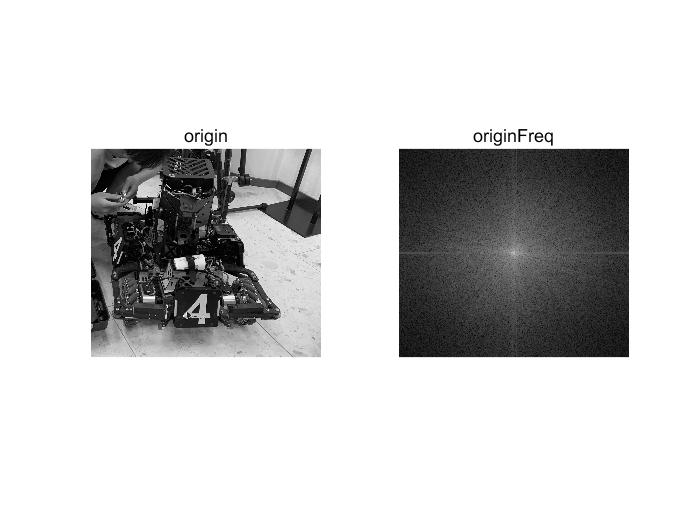


图5 灰度原图及频谱

Fig5 Origin image of gray & image on frequency domain

椒盐噪声在5%分布下的滤波结果,如图6所示：

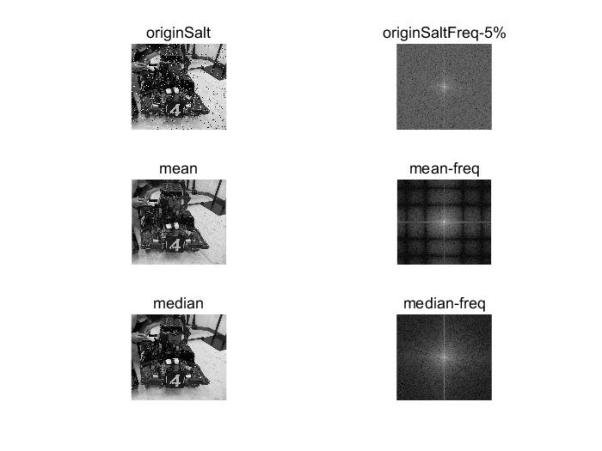


图6 椒盐噪声5%分布率下滤波效果

Fig6 Filter effects of images with 5% salt&pepper noise

椒盐噪声在30%分布下的滤波结果,如图7所示：

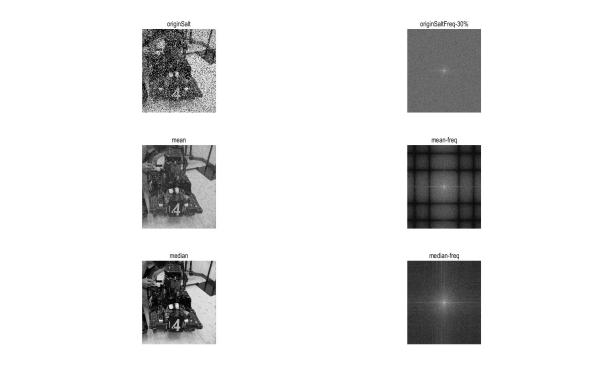


图7 椒盐噪声30%分布率下滤波效果

Fig7 Filter effects of images with 30% salt&pepper noise

椒盐噪声在55%分布下的滤波结果,如图8所示：

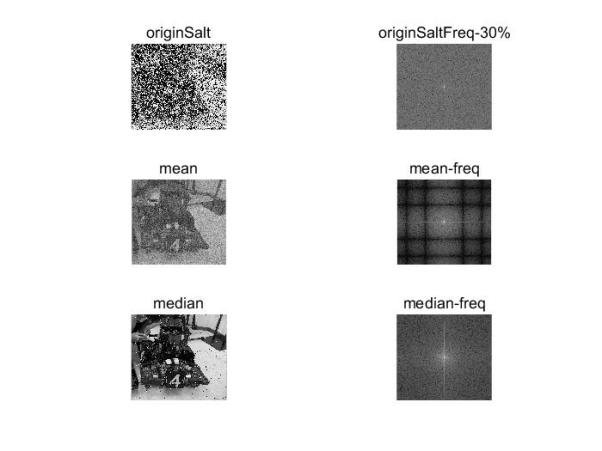


图8 椒盐噪声55%分布率下滤波效果

Fig8 Filter effects of images with 55% salt&pepper noise

椒盐噪声在75%分布下的滤波结果,如图9所示：

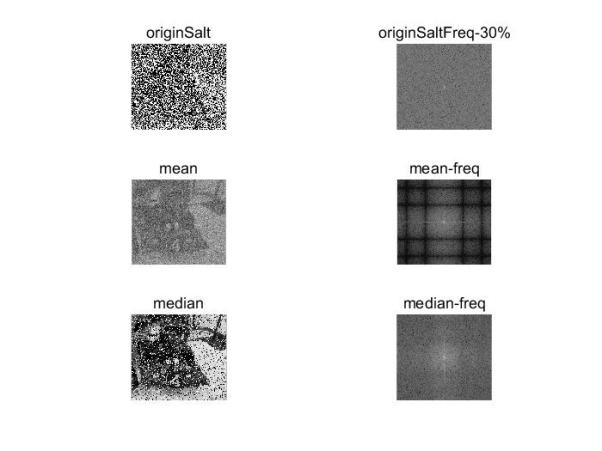


图9 椒盐噪声75%分布率下滤波效果

Fig9 Filter effects of images with 75% salt&pepper noise

不同密度下滤波效果相关性测试结果如表1所示：

表1 两种滤波器对于不同密度椒盐噪声的处理效果相关性

Table1 Related coefficients of images with different densities of salt&peeper noise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法/密度 | 0.05 | 0.3 | 0.55 | 0.75 |
| 均值 | 98.8% | 95.2% | 85.2% | 63.0% |
| 中值 | 99.6% | 99.2% | 95.3% | 67.4% |
| 原噪音 | 91.1% | 57.2% | 32.5% | 16.7% |

由表1分析可知，全密度下，对于校验噪音，中值滤波的效果都优于均值滤波；在30%密度以下，中值滤波几乎可以还原原图，但在60%密度以上，效果则并不理想，需要寻找其他更合适的滤波器。

由原理分析可知，均值滤波本身存在一些局限：不能很好地保护图像细节，在图像去噪的同时也破坏了图像的细节部分，从而使图像变得模糊，不能很好地去除噪声点，特别是椒盐噪声。

3.6.2 不同种类噪声的比较

此实验中采用高斯噪声、椒盐噪声以及乘性噪声进行实验。

滤波原始图像灰度图及频谱图如图10所示

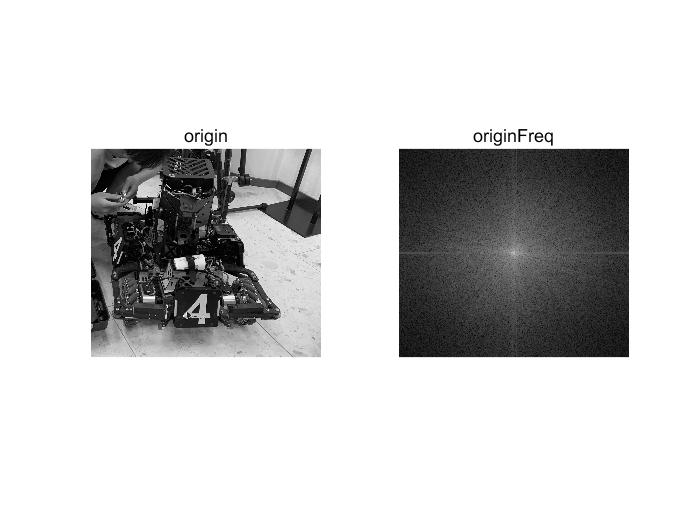


图10灰度原图及频谱

Fig10 Origin image of gray & image on frequency domain

高斯噪声的滤波结果,如图11所示：

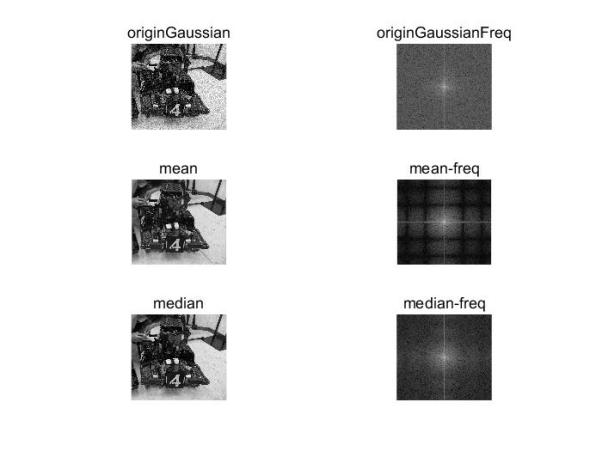


图11高斯噪音滤波及频谱

Fig11 Process Gaussian noise & image on frequency domain

椒盐噪声的滤波结果,如图12所示：

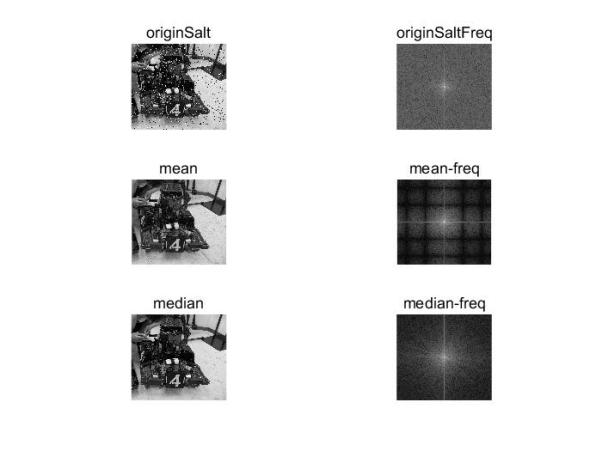


图12椒盐噪音滤波及频谱

Fig12 Process Salt&Pepper noise & image on frequency domain

乘型噪声的滤波结果,如图13所示：

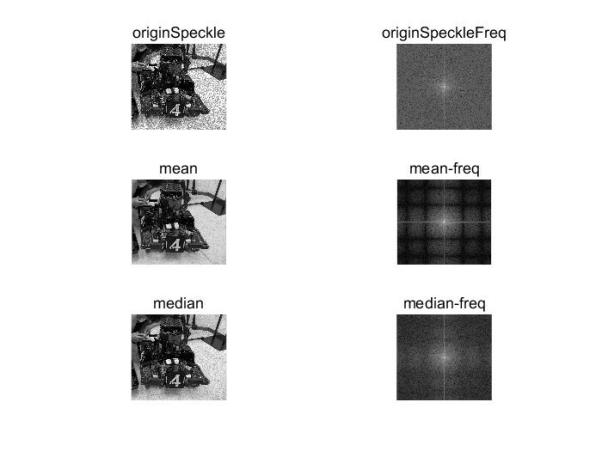


图13乘性噪音滤波及频谱

Fig13 Process Speckle noise & image on frequency domain

表2 两种滤波器对不同种噪声的处理效果

Table2 Results of images with different noise

|  |  |  |  |
| --- | --- | --- | --- |
| 方法/噪声 | 高斯 | 椒盐 | 乘性 |
| 均值 | 99.0% | 98.8% | 98.9% |
| 中值 | 98.2% | 99.6% | 98.6% |
| 原始噪音 | 95.5% | 91.2% | 94.1% |

由表2 可知，高斯噪声下，均值滤波更优；椒盐噪声下，中值滤波更优；乘性噪声下，均值和中值滤波效果都比较理想。

均值滤波采用线性的方法，平均整个窗口范围内的像素值；中值滤波采用非线性的方法，它在平滑脉冲噪声方面非常有效,同时它可以保护图像尖锐的边缘，选择适当的点来替代污染点的值。因此中值滤波对椒盐噪声表现较好，对高斯噪声表现较差。

**3.7 边缘提取初步：Sobel算子提取梯度值**

对于经过中值滤波的图像，使用Sobel算子提取图像的梯度，查看滤波效果。未来将查阅相关边沿数据判断边缘的阈值进而提取边缘。但由于目前时间限制，未进一步开展。原理：利用2.2中提到的Sobel算子，如图13所示：

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

|  |  |  |
| --- | --- | --- |
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

图13 Sobel算子，Gy&Gx

Fig13 Sobel operator, Gy&Gx

与原图像矩阵做卷积，使用二维卷积函数

MommyTalk1636551011836

分别得到水平和竖直方向两个梯度值并取绝对值，再根据公式

MommyTalk1636551132907

进行叠加，得到图像的整体梯度。

得到的结果如图14所示：

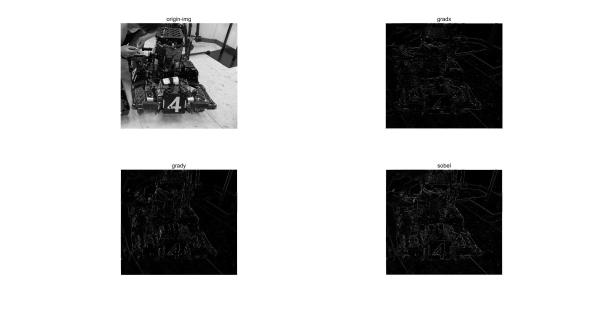


图14 水平、垂直梯度及整体梯度

Fig14 Gradients on Gx, Gy and whole

如需要提取边缘，应进一步对边缘进行处理，应根据得到***grad***设置阈值，从而区别何处是真正的边界。

**4 结论**

对于不同密度的椒盐噪声，由于均值滤波的线性性质，使得滤波时会产生细节的丢失和模糊，因此对于椒盐噪声的滤波效果较差；

对于同种密度的不同噪声，由于中值滤波的非线性，对于平滑脉冲噪声方面十分有效，因此处理椒盐噪音十分有效，但对于较为粘连的高斯噪音，效果就比较差。而对于乘性噪音，二者滤波效果相当；

每一种滤波方式都有其自身的优缺点，对于不同的情形，应当选择最合适的滤波器，并对滤波模板和“加权系数”矩阵加以调整，才能的到最合适的滤波图像。

（致谢：感谢段老师十周以来的指点，不局限于书本和考试的教学方式使我十分收益，也引导我们要去关注实际问题，真正实现理论结合实践。在最后的数据分析方面也使我更深入地思考，量化不同方法的区别，也对未来的研究道路打下基础。信号处理意义不言而喻，望十周之所学能化为今后之所用。）

**参考文献：**

[1] 毛星云：OpenCV3编程入门，电子工业出版社

[2] 图像滤波，百度百科载，

https://baike.baidu.com/item/图像滤波/11042217

(作者: 李昀哲，E-mail:1669077595@qq.com。)