

基于网格划分的混合噪声滤波算法

唐向红, 郑 阳, 杨全伟

(贵州大学 现代制造技术教育部重点实验室, 贵州 贵阳 550003)

摘 要: 提出了基于网格划分的混合噪声滤波算法. 算法在网格划分基础上定义边缘网格, 通过定位出噪声点的位置, 在边缘网格和非边缘网格内同时对椒盐噪声和高斯噪声进行处理. 实验结果表明, 该算法与已有的混合噪声滤波算法对比, 能够有效滤除混合噪声和保护图像边缘及细节信息, 同时对单一噪声也有不错滤除效果.

关键词: 混合噪声; 网格划分; 边缘网格

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2016)06-0087-05

The Filtering Algorithm for Mixed Noise Based on Grid Division

TANG Xiang-hong, ZHENG Yang, YANG Quan-wei

(Key Laboratory of Advanced Manufacturing Technology,
Ministry of Education, Guizhou University, Guiyang 550003, China)

Abstract: This paper presented an algorithm of removing mixed noise based on grid division. The algorithm define edge grid based on grid division. When all mixed noise positioned at the grid, the algorithm use two different methods to deal with edge grid and non edge grid. Experimental results show the proposed algorithm outperforms the existing algorithms in suppress mixed noise effectively and preserve the image details. At the same time, the algorithm has good effect in filtering single noise.

Key words: mixed noise; grid division; edge grid

1 引言

图像携带着丰富的信息, 在传输和存储过程图像会受到各种噪声干扰. 噪声会对图像的视觉效果和质量产生很大破坏, 同时对后续的图像处理有一定影响.

混合噪声由高斯噪声和椒盐噪声这最常见的两种噪声组成. 对于混合噪声研究, 由于该噪声有滤除困难这一特性, 所以已有的算法只是对滤除一种噪声着重研究. 国内外学者对混合噪声滤除, 进行了一定的研究^[1-6]. 文献[1-3]提出的算法在滤除噪声的同时, 也各自存在着若干局限, 即误判率较高, 图像的边缘及细节信息丢失, 且无法滤除颗粒噪声. 此外, 文献^[4]提出了基于噪声分离和小波阈值自适应

图像去噪算法, 文献^[5]提出了一种去除混合噪声的滤波算法, 两种算法都是通过噪声检测分离噪声, 再进行滤波处理. 但该类算法很容易把图像边缘点误判为噪声点, 使得图像边缘被破坏. 还有学者利用 PCNN 对混合噪声进行处理, 文献^[6]在简化型 PCNN 基础上对混合噪声进行研究, 该算法结合的算法过多, 使得时间复杂度增加, 而且图像边缘及细节信息遭到破坏. 目前, 已有的混合噪声研究都是先滤除椒盐噪声再将处理后的图像进行高斯噪声去噪, 这样会产生二次误差, 对图像复原产生影响.

针对这样的情况, 本文提出基于网格划分的混合噪声滤波算法. 该算法通过网格划分方法, 定义边缘网格和非边缘网格, 在网格内利用简化的 PNCC 对被噪声污染的图像进行预处理, 定位噪声点的位

收稿日期: 2015-08-20; **修回日期:** 2015-10-13

基金项目: 国家科技支撑计划课题(2012BAF12B14); 贵州省重大科技专项(黔科合重大专项字(2013)6019); 贵州省工业攻关项目(黔科合 GY 字[2014]3004)

置. 对边缘网格和非边缘网格采取不同的方法处理, 获得滤波后的图像. 由于算法在网格内是同时对椒盐噪声和高斯噪声进行处理, 所以不会对滤波图像产生二次误差和影响图像复原. 而且本算法充分考虑图像边缘的情况, 这样能够更好地保护图像边缘和细节信息.

2 相关定义

定义 1 边缘网格是图像边缘像素所在的网格. 为保护图像的细节和图像边缘特征, 提出了边缘网格这一概念, 它有利于更快、更好地处理边缘像素. 通常局部灰度有显著变化的区域就极可能为图像边缘. 如何判断噪声所在网格是否为边缘网格. 本文用万有引力边缘检测^[7], 该方法简单、有效. 计算出像素点在其邻域内所受水平及垂直方向万有引力合力 F^x 和 F^y . 计算总合力大小 $|F| = \sqrt{(F^x)^2 + (F^y)^2}$, 通过设定一个阈值合力大于此阈值的为边缘点, 小于此阈值的为非边缘点. 边缘网格就是包含这些边缘点较多的网格.

定义 2 噪声定位 N_{ij} 表示噪声点 p_{ij} 的污染情况, 规定

$$N_{ij} = \begin{cases} 1, \max\{W_D(S_{ij})\} - \delta \geq S_{ij} \\ \text{或 } S_{ij} \leq \min\{W_D(S_{ij})\} + \delta \\ -1, \text{otherwise} \end{cases}$$

即表示噪声点 p_{ij} 污染程度为 $N_{ij} = 1$ 点, 认定该点为椒盐噪声点, 像素 p_{ij} 污染程度为 $N_{ij} = -1$ 点, 该点认定为高斯噪声点. 其中 $W_D(S_{ij})$ 表示以像素 S_{ij} 为中心的设定邻域范围内的区域, δ 表示一个范围参数. $\min\{W_D(S_{ij})\}$ 表示对窗口 $W_D(S_{ij})$ 内的像素灰度最小值同理, $\max\{W_D(S_{ij})\}$ 表示对窗口 $W_D(S_{ij})$ 内的像素灰度最大值.

定义 3 峰值信噪比 (PSNR), 峰值信噪比是最常用的图像逼真度计测参数, PSNR 值越大, 表示恢复图像与原始图像越接近. 其定义为

$$P_{\text{PSNR}} = 10 \lg \frac{M \times N \times 255^2}{\sum_{m=1}^M \sum_{n=1}^N [F(m, n) - F'(m, n)]^2}$$

式中, M, N 表示图像的尺寸; $F(m \times n)$ 表示原始图像的灰度值; $F'(m, n)$ 表示去噪后的图像灰度值.

3 图像网格划分算法描述

本文提出的图像网格划分方法, 如图 1 所示. 基本思想是: 在灰度图中以所要生成的像素大小来确定网格的大小和数量, 然后对图像像素进行网格划

分, 以每个网格来确定每个像素点的邻域范围, 每个网格的中心位置就是所对应的像素点位置, 这样能够保证所生成的所有像素点的准确度. 由于每个像素点都划分到了对应的网格内, 因此在计算每个像素灰度值时不需再次对所有图像像素进行重新遍历, 从而减小计算量. 其具体方法如下所示.

(1) 首先, 得到要生成的灰度图的像素大小, 假设为: $n \times m$. 则相应的定义一个 $n \times m$ 的二维数组, 假设为 Array-P . 该数组中每一个单元都是一个容器, 假设为 p , 该容器用来承放划分到该网格内的图像像素.

(2) 遍历图像像素, 找图像像素的边界值: $\max(x)$ 、 $\min(x)$ 、 $\max(y)$ 、 $\min(y)$.

(3) 分别计算边长 $x_{\text{length}} = \max(x) - \min(x)$ 和 $y_{\text{length}} = \max(y) - \min(y)$, 比较边长大小取大的边长 L , $L = \max(x_{\text{length}}, y_{\text{length}})$. 然后计算出 L/n , L/m , 网格单元边长为 t , $t = \min(L/n, L/m)$ (对应像素形状, 网格的每个单元为正方形).

(4) 以计算出的边长 L 为网格总长, 并画正方形. 令 $\text{start_x} = \min(x)$, 得到 X 方向的起始位置.

(5) 对图像进行遍历, 对每个点 p 计算 $tn = (p[x] - \text{start_x}) \% t + 1$ 和 $tm = (p[y] - \text{start_y}) \% t + 1$, 这里符号 $\%$ 表示整除. 然后将点 p 放入 $\text{Array-P}[tn][tm]$ 单元的容器中.

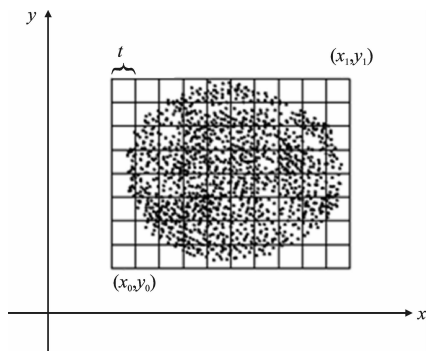


图 1 图像网格划分方法

上述方法仅对图像像素进行了 $2n$ 次循环 (假设像素点个数为 n). 同时本文提出的网格划分方法, 因为画出边长 L 的正方形, 并以最小的边长, 使得划分网格的更均匀, 为下一步混合噪声去除提供了基础.

4 混合噪声滤波算法描述

在上面的图像网格划分的基础上, 本文提出了混合噪声的滤波算法. 基本思想是: 首先根据定义 2

定位噪声点的位置且对噪声点进行判定。遍历噪声点所在的网格,由定义 1 区分网格是否为边缘网格,若噪声点所在网格是非边缘网格,针对网格内有椒盐噪声点,将网格内像素点用线性分类器进行分类,计算不同类的平均值,以最近邻的平均值替换掉噪声点。对高斯噪声点则以网格为模板计算出网格内非噪声点均值大小,并求出该网格内方差,由以上均值和方差求出替换噪声点的灰度值。若噪声点所在网格是边缘网格,先找出网格内边缘点,以最近的边缘点为参考,比较边缘点和噪声点的灰度值大小,若噪声点的灰度值比边缘点大,则用以这个噪声点为中心找与其近邻的 k 个灰度值比边缘点大的像素点,若在网格内找不到,则在其相邻的网格内找与该点近邻的且灰度值比边缘点大的像素点。并且算出 k 个点灰度值的平均值记为 A_{ij}' ,并用 A_{ij}' 替换掉噪声点。若噪声点的灰度值比边缘点小,则用以这个噪声点为中心找与其近邻的 k 个灰度值比边缘点小的像素点,并计算出 A_{ij}' ,并用 A_{ij}' 替换掉噪声点。而对于对高斯噪声点则可以利用边缘点将网格内像素点分为两类,再用类似非边缘网格方法来替换高斯噪声点。

其算法流程如图 2 所示。

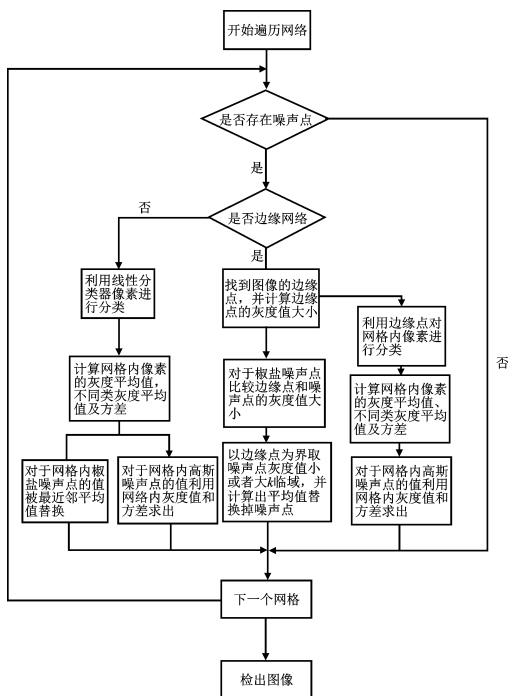


图 2 混合噪声滤波算法流程描述

混合噪声去噪算法对边缘网格和非边缘网格采用不同方法处理,算法具体描述如下:

像素 p_{ij} 为污染程度 $N_{ij} = 1$ 点,因此该像素认

定为椒盐噪声点,像素 p_{ij} 为污染程度 $N_{ij} = -1$ 点,因此该像素认定为高斯噪声点,将 p_{ij} 所在的网格 $G(p_{ij})$ 内非噪声点所有灰度值点的集合,记为 $G_D(p_{ij})$,即 $G_D(p_{ij}) = \{p_{nm} \mid N_{nm} = 0, p_{nm} \in G(p_{ij})\}$ 。

当 p_{ij} 所在网格是非边缘网格时,对像素 p_{ij} 为污染程度 $N_{ij} = 1$ 点,特征向量 $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ 为除去噪声点外各像素的灰度值分量的线性组合,第 i 类线性判别函数的表达式:

$$g_i = w_{i1}x_1 + w_{i2}x_2 + \dots + w_{id}x_d + w_{i0},$$

$$= \sum_{k=1}^d w_{ik}x_k + w_{i0},$$

$$g_i(x) = W_i^T x + w_{i0}, i = 1, 2, \dots, d,$$

$W_i = [w_{i1} \dots w_{id}]^T$ 为全向量, w_{i0} 为阈值。

判别函数: $g(x) = g_1(x) - g_2(x)$,

决策层面: $g(x) = g_1(x) - g_2(x) = 0$

将网格内像素分为两类。

对此求出 p_{ij} 所在网格两类各个类里包含像素点灰度值平均值 A_{ij} , $A_{ij} = \text{average}\{G_D(p_{ij})\}$ 并用与 p_{ij} 最近邻的 A_{ij} 替换 p_{ij} ,记 G_{ij} 为噪声点 p_{ij} 的灰度值。其余像素点不做处理,即 $G_{ij} = A_{ij}$ 。

对像素 p_{ij} 为污染程度 $N_{ij} = -1$ 点,由于计算出 $G_D(p_{ij})$ 像素和两类像素的均值 (μ, μ_1, μ_2) 和方差 $(\sigma', \sigma_1, \sigma_2)$ 。然后分别计算出去噪后的灰度值 $R_1(i, j), R_2(i, j)$ 。用 $R(i, j)$ 值替换 p_{ij} 。若网格内像素分布为 X 行 Y 列,则

$$\mu = \frac{1}{XY} \sum_{i,j} s(i, j),$$

$$\sigma^2 = \frac{1}{XY} \sum_{i,j} s^2(i, j) - \left(\frac{\mu}{XY}\right)^2,$$

$$R(i, j) = \mu + \frac{\sigma^2 - \sigma'^2}{\sigma^2} (s(i, j) - \mu).$$

当 p_{ij} 所在网格是边缘网格时,找到图像边缘点 b_{ij} ,得到 b_{ij} 的灰度值 M_{ij} 。对于边缘网格的椒盐噪声,比较椒盐噪声点 G_{ij} 和的 M_{ij} 大小,判断出 p_{ij} 像素点 k 邻域的取值界限,当 $G_{ij} > M_{ij}$ 时, P_{ij} 像素点的 k 邻域取值为除去噪声点的所有像素点灰度值并且都比 M_{ij} 大,同理当 $G_{ij} < M_{ij}$ 时, P_{ij} 像素点的 k 邻域取值都比 M_{ij} 小。

计算出 P_{ij} 像素点的 k 邻域内所有像素点灰度值平均值 A_{ij}' ,并用 A_{ij}' 取代 G_{ij} ,即 $G_{ij} = A_{ij}'$ 。

对于边缘网格的高斯噪声点,找到高斯噪声点最近边缘点 b_{ij} ,以边缘点 b_{ij} 的灰度值 M_{ij} 为界,将网格分为两类。其中可能会其中一类像素点极少,因而

设定阈值 T , 当分类使得一边过少时, 在周围网格内找到最近邻的像素点使总数达到阈值.

将分类的像素点以类似步骤 5 的方式, 求出 $R'(i, j)$. 并用 $R'(i, j)$ 替换高斯噪声点.

以上算法描述中 2-4 为非边缘网格滤波方法, 5-8 为边缘网格滤波方法. 通过网格化处理像素, 使得在滤波过程中不需要考虑领域大小, 减少了时间复杂.

已有的混合噪声研究都是先滤除椒盐噪声再将处理后的图像进行高斯噪声滤波, 然而在滤除椒盐噪声后图像就已经和未含噪声的原始图像存在差异, 而之后高斯噪声滤除算法却还是利用滤除椒盐噪声后的图像为初始图像, 这样会产生二次误差, 对图像复原产生影响. 由上述算法的 3、4、6、8 可知, 本文算法是以图像中非噪声点为对象同时对椒盐噪声和高斯噪声进行处理, 这样处理就避免了上述误差, 使得图像复原更精确. 此外本算法充分考虑边缘点的情况, 为了保护图像边缘提出了边缘网格滤波方法, 这种方法能够更好的保护图像边缘和细节信息.

5 实验结果

本文实验去噪算法在主频 2.7 GHz、内存 4.0 GB 的 Intel(R) PENTIUM(R) PC 机上采用 matlab 编程语言实现仿真. 本文选用文献[3]、文献[5]和文献[6]中三种算法作为实验对比算法. 实验选用常用的 lena 为基础对比图像. 为评价算法的滤波效果采用主观和客观评价两种标准, 客观评价选用常用的峰值信噪比作为参数; 图像主观评价则是以去噪前后的图像做对比.

实验一 客观评价

本实验选用峰值信噪比作为实验对比参数, 在实验基础图像上人工加入混合噪声, 表 1 所示为不同混合噪声条件下, 各算法滤波后 PSNR 值对比. 为了论证算法对椒盐噪声的滤除能力, 本文用滤除椒盐噪声能力强的自适应中值滤波^[8]为参照, 表 2 所示为上述算法和本文算法滤除椒盐噪声后的 PSNR 值. 表 3 反映了算法对单独对高斯噪声的滤除能力, 其对比算法为经典的维纳滤波算法^[9].

由表 1 可以看出在各种比例混合噪声中, 与其他几种方法对比, 本文提出的算法 PSNR 值均为最大的. 也就是说本文的滤波算法去噪效果均好于其他滤波算法. 由表 2 和表 3 可以得出, 本文在只有椒盐噪声和高斯噪声的情况下, 滤除单一噪声也有不

错去噪效果, 滤除椒盐噪声能力尤为突出. 这是因为算法在网格内是同时对椒盐噪声和高斯噪声进行处理, 图像复原比较准确, 且在算法中充分考虑图像边缘的情况, 图像边缘及细节信息得到有效保护.

表 1 四种滤波算法的 PSNR 值对比

混合噪声	文献[3]	文献[5]	文献[6]	本文算法
$p=0.1, \sigma=10$	26.113 7	25.637 2	26.473 5	27.853 3
$p=0.1, \sigma=15$	25.947 7	25.569 8	26.351 2	27.710 1
$p=0.1, \sigma=20$	25.865 1	25.473 6	26.146 0	27.534 9
$p=0.15, \sigma=10$	25.835 4	25.518 9	26.263 1	27.638 9
$p=0.15, \sigma=15$	25.617 3	25.481 0	26.157 3	27.427 7
$p=0.15, \sigma=20$	25.434 6	25.427 5	26.098 3	27.334 5
$p=0.2, \sigma=10$	25.787 3	25.463 2	26.194 3	27.596 5
$p=0.2, \sigma=15$	25.599 2	25.327 6	26.039 1	27.361 5
$p=0.2, \sigma=20$	25.317 0	25.219 8	25.978 4	27.143 6

表 2 椒盐噪声滤波结果比较

椒盐噪声	自适应中值滤波	本文算法
$p=0.05$	29.547 1	30.724 5
$p=0.1$	28.932 6	29.653 2
$p=0.15$	28.103 2	28.873 5
$p=0.2$	26.037 8	26.473 4

表 3 高斯噪声滤波结果比较

高斯噪声	维纳滤波	本文算法
$p=5$	26.157 4	25.872 7
$p=10$	25.971 1	25.519 6
$p=15$	24.786 2	25.052 2
$p=20$	24.324 5	24.1523

实验二: 主观实验

在进行主观评价时, 在 lena 图像中加入混合噪声(椒盐噪声 $p=0.15$ 和零均值高斯噪声 $\sigma=15$), 图 3 所示为加入混合噪声 lena 图像、文献[3, 5, 6]和本文算法进行滤波后的输出图像. 为了验证本文提出的算法保护图像边缘的能力效果, 在实验基础图像 lena 上取帽子的局部区域, 如图 4 所示, 之后对四种滤波算法处理后获取的局部图进行对比如图 5 所示.

由图 3 可知, 在滤除混合噪声方面, 本文提出的滤波算法相较于其他算法, 滤波效果最好. 图 5 表明本文算法在保护图像边缘方面明显优于其他的滤波算法.

通过主观和客观两个方面的滤波性能效果比较, 基于网格划分的混合噪声滤波算法, 在滤除混合

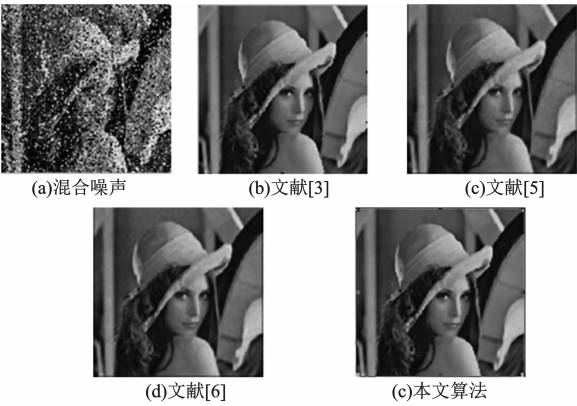


图 3 混合噪声和四种算法滤波后的输出图像



图 4 lena 帽子的局部区域

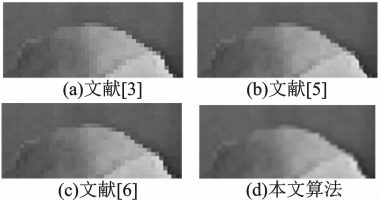


图 5 lena 帽子滤波后的局部图

噪声方面优于现有的混合噪声滤波算法,能够较好的还原图像细节和保护图像边缘及细节信息,同时在滤除单一噪声也有不错去噪效果.

6 结束语

针对现有的混合噪声滤波算法存在的问题,本

文在网格划分的方法的基础上,提出了基于网格划分的混合噪声滤波算法. 由主观和客观实验结果得出,本文提出的算法滤除混合噪声后,能够较好地还原图像,还能很好地保护图像边缘信息,同时在滤除单一噪声方面也有不错去噪效果. 下一步将对细节丰富图像的优化算法进行研究.

参考文献:

[1] 朱士虎,游春霞. 一种改进的均值滤波算法[J]. 计算机应用与软件, 2013, 30(12):97-99.

[2] Zhu W, Han J, Chen P, et al. Adaptive median filter algorithm based on multi-stage noise detection[J]. Opto-Electronic Engineering, 2013, 40(10):63-69.

[3] 胡浩,王明照,杨杰. 自适应模糊加权均值滤波器[J]. 系统工程与电子技术, 2002, 24(2):15-17.

[4] 万千,薛明. 基于噪声分离和小波阈值自适应图像去噪算法[J]. 电子科技, 2011, 24(5):94-96.

[5] 王益艳. 一种去除图像混合噪声的滤波算法[J]. 计算机应用与软件, 2010, 27(10):98-102.

[6] 张艳珠,李媛,李小娟. 简化型 PCNN 的混合噪声图像滤波算法研究[J]. 控制工程, 2013, 20(5):829-832.

[7] 张春雪,陈秀宏. 基于非线性滤波的万有引力边缘检测方法[J]. 计算机应用, 2011, 31(3):763-766.

[8] 王晓凯,李锋. 改进的自适应中值滤波[J]. 计算机工程与应用, 2010, 46(3):175-176.

[9] 尹方平,苏静. 一种改进的核维纳滤波器图像去噪算法研究[J]. 激光与红外, 2010, 40(5):549-553.

作者简介:

唐向红 男,(1979-),博士,副教授,硕士研究生导师. 研究方向为数据挖掘、制造业信息化、物联网技术.

郑 阳(通讯作者) 男,(1990-),硕士研究生. 研究方向为制造业信息化. E-mail:403370748@qq. com.

杨全纬 男,(1989-),硕士研究生. 研究方向为制造业信息化.