

对数字图像中最先进的去噪技术的系统回顾

Nishant Bindal • Rajanbir Singh Ghumaan • Prateek Jeet Singh Sohi •

Nikhil Sharma • Hemdutt Joshi • Bharat Garg

Received: 2 January 2021 / Revised: 23 February 2022 / Accepted: 9 March 2022

Published online: 9 April 2022

摘要

数字图像处理是数字信号处理的一个子类别，重点是研究用于增强或恢复的处理技术。被各种类型的噪声破坏的图像的去噪就属于这个类别。去噪主要是为了提高受影响图像的可理解性。用故障设备拍摄的图像或经过长途传输的图像极易受到脉冲噪声的破坏，因此，提出了各种技术来去除图像中的这种噪声。每种技术都有自己的优点和缺点。本文对这些技术进行了全面的比较分析，在广泛的噪声范围内进行了分析。所有的过滤技术都在 MATLAB 中实现，并用标准的基准图像数据进行模拟，并对定性指标即峰值信噪比（PSNR）和结构相似度指数（SSIM）进行评估和比较。因此，本文对各种最先进的噪声去除技术进行了全面的比较分析。

关键词 中位数滤波器, 均值滤波器, 盐和胡椒噪声, 脉冲噪声, 脉冲噪声, 图像修复和去噪

1 简介

图像的失真可能会因噪声的干扰而导致有用的特征丢失。图像中的噪声可能源于图像捕获设备的缺陷、处理技术的失误、传输到远距离时的干扰等因素。若要从视觉上分辨图像中是否存在脉冲噪声，观察者应寻找异常明亮或暗的像素。这些亮或暗的像素对应着像素的最大值（255）或最小值（0）（对于 8 位图像，0 和 255 是可能的最小和最大值）。利用各种图像处理技术去除这些噪声的过程通常被称为去噪。

随着时间的推移，人们提出了各种用于恢复被盐和胡椒噪声破坏的图像的最先进的去噪滤波器。这篇评论文章的主要目的是让读者了解各种不同的过滤器，以及这些过滤器可以被归类的各种类别的知识。这些论文的主要贡献如下。

- 我们总结了大多数最先进的过滤器的工作技术。
- 为了让读者更好地理解，还给出了使用各种技术过滤的选定基准图像（这些图像被密度为 95 % 的盐和胡椒噪声所破坏）的视觉表现。

- 总而言之，本文将成为一个指南，使研究人员了解从 1995 年到 2021 年这四分之一世纪里盐和胡椒噪声去除领域的主要发展。

本文的其余部分组织如下。第 3 节解释了盐和胡椒噪声、其影响和不同的过滤技术。第 4 节向读者介绍了仿真环境和对选定的过滤算法进行的比较分析。本节使用不同的表格和图表向读者介绍了这些过滤器在广泛的噪声密度范围内的比较分析，即低度（不超过 30%）、中度（在 30% 和 90% 之间）和高度（超过 90%）。最后，在第 5 节中向读者提供了从第 4 节得出的结论。

2 评估参数

对于各种过滤器的质量分析，使用了不同的质量评估指标。大体上，这些质量评估可以是主观的或定量的。主观的质量分析是基于人眼从给定图像中提取结构信息的能力。然而，定量质量指标利用数学表达式来计算一个给定图像的质量。数学类很容易实现，并提供统一的预测质量度量。在这种方法中，数学公式被应用于计算噪声/去噪声图像在原始图像上的噪声量。在本文中，以下指标被用于质量分析。

2.1 平均平方误差 (MSE)

平均平方误差或平均平方偏差 (MSD) 是处理后的像素值与准确的像素值之间的平均平方差。MSE 的数学表达式由 (1) 给出。

$$MSE = \frac{1}{M + N} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - y_{i,j})^2 \quad (1)$$

2.2 均方根误差 (RMSE)

RMSE 可以计算为实际像素值和恢复后的像素值之间的平均平方差的根。它也可以通过取 MSE 的平方根来计算。

$$RMSE = \sqrt{\frac{1}{M + N} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - y_{i,j})^2} \quad (2)$$

2.3 峰值信噪比 (PSNR)

PSNR 只是最大可能的信号功率与改变图像定量表示的噪声信号功率之间的比率。PSNR 经常被用来衡量一项技术的修复精度。PSNR 可以被认为是人类对重建质量的一种近似感知。PSNR 的数学表达式为分贝 (dB)，由 (3) 给出。

$$PSNR = 10 \log_{10}(MAX^2 / MSE) \quad (3)$$

其中，MSE 和 MAX 分别为信号的均方误差和最大值。

2.4 结构相似性指数 (SSIM)

顾名思义, SSIM 是用来测量修复/处理后的图像和实际图像之间的结构相似性。SSIM 是以原始/未压缩的图像作为参考, 对图像质量进行的测量。

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4)$$

其中 $\mu_x(\sigma_x)$ 和 $\mu_y(\sigma_y)$ 分别是 x 和 y 方向的平均值 (标准差)。而常数 C1 和 C2 的选择是为了使 SSIM 的最大 (最小) 值为 1(0)。

2.5 图像增强系数 (IEF)

IEF 是实际图像和损坏图像的 MSE 与实际图像和重建图像的 MSE 之比, 如 (5) 所示。它是估计去噪后图像质量提高的准确措施。

$$IEF = \frac{\sum_{i=1}^M \sum_{j=1}^N (\hat{y}_{i,j} - x_{i,j})^2}{\sum_{i=1}^M \sum_{j=1}^N (y_{i,j} - x_{i,j})^2} \quad (5)$$

(3) 至 (5) 中使用的术语:M, N -图像的维度

$x_{i,j}$ -没有任何形式的噪声的实际图像

$y_{i,j}$ -去除噪音的图像

$\hat{y}_{i,j}$ -损坏的图像

MAX -函数给出图像的最大允许像素值。

本文主要围绕脉冲噪声及其去噪方法展开。脉冲噪声也可称为数据下降噪声, 因为它在统计上将像素值的强度下降到最大或最小的可能值。因此, 也被称为”盐和胡椒噪声”。本文试图解决许多研究人员在脉冲噪声去除技术方面的著名技术。

3 不同的盐和胡椒的噪音消除器

3.1 前提概念

术语”窗口”或”内核”指的是预定大小的像素的邻域。通常情况下, 在过滤过程中采取三种窗口尺寸。3×3、5×5 和 7×7。在去噪过程中, 考虑到较高的窗口尺寸, 噪声效应会减少, 但图像会失去其特征。在低噪声密度下, 小的处理窗口可以有更好的性能。在高噪声密度下, 大的窗口尺寸会有更好的结果, 但会增加图像的模糊效果。

滤波是一种增强图像的技术。在这种技术中, 恢复的图像中的像素是通过对噪声及其邻近的像素进行一些操作来计算的。在图像过滤中, 有一些操作被应用, 如平滑、锐化和边缘增强。滤波技术可以在空间域或频率域进行。在空间域中, 这些操作直接应用于图像的像素上。另一方面, 在频域中, 滤波操作是通过使用傅里叶变换或任何其他图像函数的变换将空间域映射到频域来进行的。然而, 本文介绍了各种最先进的空间滤波技术, 如图 1 所示。用于消除盐和胡椒噪声的各种滤波技术分为以下几类。

3.2 线性滤波器

在这些过滤技术中，恢复的像素被计算为以噪声像素为中心的特定大小的窗口的像素值的线性组合。这些技术通常伴随着模糊效应，并且在去除诸如信号相关噪声方面表现不佳。最常见的线性滤波器是下面讨论的平均数或平均滤波器。

3.2.1 均值滤波器

平均值过滤器是最简单的过滤器之一。它是一个基于平均老化技术的线性滤波器。处理窗口通常是一个正方形的形状。这个滤波器计算中心噪声像素周围的处理窗口的平均值，中心像素的值被计算的值所取代。损坏图像的平均值在 S_{xy} 定义的区域内计算 [1, 5, 6, 24]。

$$f(x, y) = \frac{1}{MN} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (6)$$

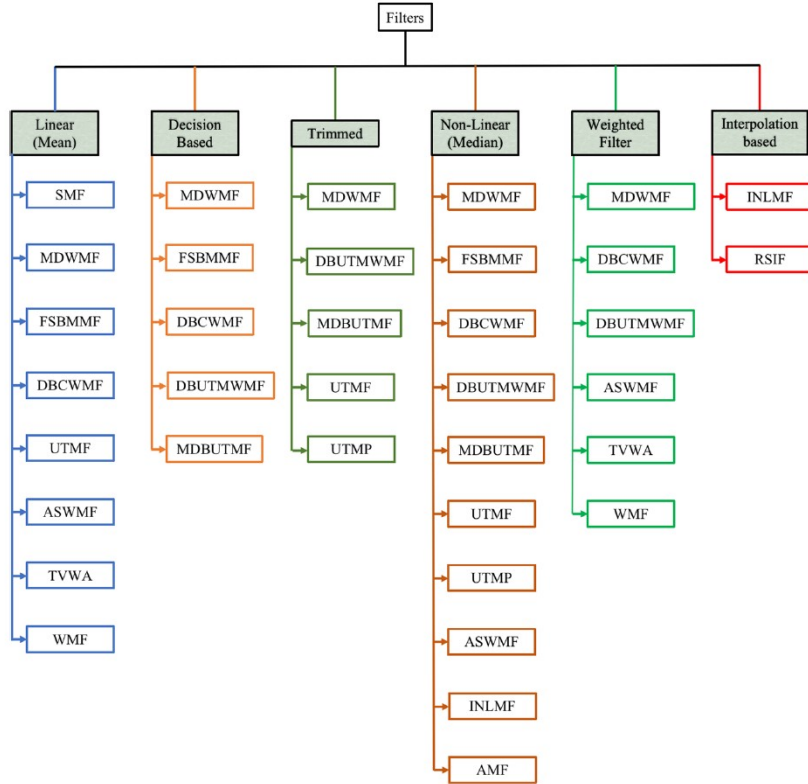


图 1: 各种盐 and 胡椒噪声去除技术的分类流程图

其中， $f(x, y)$ 是去噪图像， $g(x, y)$ 是 $M \times N$ 维的原始图像。

3.2.2 加权平均滤波器

这可以看作是简单的平均滤波技术的进步。均值滤波和加权均值滤波的主要区别在于，局部邻域内的指定像素被乘以权重矩阵中指定的权重 [29, 32]。这种滤波器的数学表达式如下。

$$f(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) g(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)} \quad (7)$$

3.3 非线性滤波器

一些非线性滤波器 [2-4, 7, 10-12, 16, 18-20, 29, 31] 被开发出来, 以提供比线性滤波器更好的性能并克服其缺点。顾名思义, 这一类的滤波器不满足叠加。本节对文献中存在的各种非线性滤波器进行了全面回顾。

3.3.1 中值滤波器

它也可以被称为顺序统计滤波器。它是简单而最强大的非线性滤波器。像平均数滤波器一样, 它也通过平滑图像来消除噪音。它用所选窗口或内核的中值替换图像的像素值。这也降低了图像中不同像素之间的强度变化。一个给定的窗口/向量的中值的计算方法是, 首先将它们按升序排列, 然后选择重新排列的窗口/向量的中间值。如果所选窗口/向量中的像素数是偶数, 那么中值是通过平均两个中间像素值来计算。以下是这个过滤器的方程式。

$$f(x, y) = \text{Median}[g(s, t)] \quad (8)$$

其中 S_{xy} 是计算中值的区域。与线性滤波器相比, 中值滤波器在消除盐和胡椒噪声方面更有效。中值滤波器比线性滤波器的主要优点是, 前者可以有效地消除输入图像中非常大的噪声影响。标准中值滤波器通常在低噪声密度下更有效。在较高的噪声下, 增加处理窗口的大小会导致输出模糊, 而小窗口的噪声抑制效果不佳。

3.3.2 切换式中值滤波器

这是对标准中值滤波器的修改。滤波器中使用的噪声检测算法决定了损坏的像素、未损坏的像素以及校正算法如何使用中值公式 [12, 20, 29, 31]。

3.3.3 渐进式开关中值滤波器

这个过滤器根据预定义的固定阈值对像素的检测作出决定。主要的挑战是如何固定阈值以准确地检测出有噪声的像素。这种滤波器在高密度下不能正确地保留边缘, 因为这种滤波器在执行其操作时不考虑局部细节 [12, 20, 29, 31]。

3.3.4 修剪均值滤波器

这个过滤器使用基于决策的算法。它首先使用 3×3 的窗口进行去噪。如果处理窗口没有非噪声像素, 那么中心像素就被窗口的平均值取代, 否则就被窗口的中值取代 [3]。

3.3.5 非对称性修剪中值滤波器

在这个滤波器中, 对排序窗口两侧的损坏像素进行不对称的修剪。修剪后的窗口的中位数被计算出来, 并取代了处理后的像素。因此, 如果产生的修剪后的窗口是空的, 它将不起作用, 这主要发生在高噪声密度下。因此, 它在高噪声密度下不能很好地工作, 而且也不能保留边缘细节 [11]。

3.3.6 最小滤波器

在这个滤波器中，像素值被具有最小强度水平的相邻像素的值所取代。由于其程序，它也被称为第 0 百分位数滤波器。这意味着它适合图像中最暗的像素。

3.3.7 最大滤波器

在这个滤波器中，像素值被损坏的处理像素附近的最亮的像素所取代。它也被称为第 100 百分位数滤波器。它可以去除图像中的胡椒粉噪音。

3.4 基于决策的算法 (DBA)

在高噪声密度下，被破坏的中心像素被无噪声的邻近像素所取代。当与邻近像素重复替换时，也会出现条纹效应 [3, 4, 11, 26]。

3.4.1 改进 DBA

这个滤波器是为了避免在 DBA 中观察到的条纹效应。在这里，中断的中心像素的值是通过计算不对称修剪的处理窗口的中值来计算的。然而，在高噪声密度下，该滤波器的效率很差或非常低。

3.5 自适应滤波器

顾名思义，这些 [1, 2, 12, 16, 19, 21, 32]. 类型的过滤器的行为随着图像区域和过滤器区域的统计特征变化而变化。这些过滤器很难实现，而且比其他过滤器的性能更好。这种去噪算法所涉及的步骤是：

1. 分析: 在这个步骤中类似的图像块被收集成组。
2. 处理: 得到的组通过阈值进行过滤。
3. 合成: 然后, 过滤后的区块将用于最终的图像重建，该图像重建是由过滤后的区块上的操作计算出来的。

3.5.1 自适应中值滤波器

最常用的自适应滤波器是”自适应中值滤波器” [16]。它在过滤图像的同时保留了图像的细节。它在含有高密度盐和胡椒噪声的图像上也表现良好。这个滤波器的一个有趣的特点是，它可以在操作过程中根据预定的条件改变其窗口大小。这个过滤器涉及的步骤如下：

1. 它会计算出受噪声影响的窗口的极端值。
2. 检查计算出的中位数本身是否是一个极端值（即盐和胡椒的噪音）。

3. 如果中值也是一个被破坏的值，那么就用增加的窗口大小重申这一过程，并计算中值。窗口大小可以增加到预定的最大窗口大小。否则，进入下一个步骤。
4. 检查被处理的像素的值是否被破坏。如果被破坏，则用之前计算的中位数替换该像素，否则该像素保持不变。

AMF 在低度和中度噪声密度下工作良好。它在图像中引入了模糊效应，因为它使用的可变窗口大小有时很大，导致重建的图像中出现模糊效应。

3.5.2 自适应里兹平均滤波器

作者提出了自适应里兹平均滤波 (ARMF[8])。使用的主要概念是像素相似性，用于像素去噪。它在将图像转换为双重形式后对其进行操作。它使用基于相似性的 Riesz 平均值。它比其他不同的相似性或距离函数 (如 p-norm、Euclid 和 Hamming) 表现得更好。它也被优化来完成所有这些操作。通过使用噪声检测掩码或另一个平均值并使用相同的像素相似度概念，也有改进的余地。

3.5.3 NRSMMF

它 [25] 是一种独特的技术，根据图像的噪声密度选择要执行的操作。基本操作是。平均数、中位数、预处理值。在恢复被极高密度噪声破坏的图像方面非常有效。它使用三个噪声密度范围。对于噪声密度小于等于 25%，它检查一系列的 4 个条件来恢复像素值。如果噪声密度小于 85% 且大于 25%，它计算 5×5 窗口中先前处理的像素的平均值。否则，如果噪声密度大于 85%，则计算 5×5 窗口中先前处理的像素的平均值。这个过滤器根据特定窗口的信息做出准确的决定。

3.5.4 AWM3F

开发这个 [23] 算法的动机是为了过滤密度非常高的 SAP 噪声 (即大于 85%)。该算法使用对称填充和自适应窗口大小的概念。在第一步中，该算法试图选择有足够的非噪声像素的窗口大小进行去噪。基本操作是找到：窗口中的最小、最大和中间像素值，并找出它们的权重。在这里，使用所选窗口的最小值和最大值来计算两组高度相关的无噪声像素。如果没有找到，其大小将增加一个。最大尺寸可以是 7×7 。

3.5.5 AFMF

这个 [9] 滤波器可以被认为是自适应中值滤波器 (AMF) 的一个发展。它计算频率中位数而不是振幅中位数来恢复噪声像素的像素值。这种修改产生了比 AMF 更好的结果，因为恢复的像素值非常接近于原始像素值。其原因可能是在频域中进行的操作。这将在评估恢复的像素值时排除嘈杂的像素，并将重点放在恢复值的唯一性上。

3.6 组合滤波器

这些滤波器利用线性和非线性相结合的方法来恢复噪音像素。根据噪声密度或处理窗口中无噪声像素的可用性，这些滤波器要么使用线性方法，如均值、插值等，要么使用非线性技术，如中值进行去噪。以下各小节讨论了最近提出的各种组合滤波器。

3.6.1 基于模态决策的非对称修剪中值 (MDBUTM) 滤波器

在这个滤波器中，计算处理窗口中非噪声像素的中值，用计算出的中值替换损坏的处理像素 [11]。此外，如果一个窗口中的所有像素值都被破坏了，那么就计算处理窗口的平均值来替换中央被破坏的像素。然而，这种方法会导致过滤器的性能变差，因为这相当于给损坏的像素分配任何随机值，而不是'0'或'255'值。

3.6.2 基于决策的耦合窗口中值滤波器 (DBCWMF)

这个过滤器根据一些指定的条件选择窗口的大小。用来寻找损坏像素值的策略是中位数。如果在 3×3 的处理窗口中存在任何未被破坏的像素，那么破坏的像素就会被给定处理窗口中存在的所有非噪声像素的中值所取代。如果没有可用的无噪声像素，那么窗口大小将逐渐 (逐步) 增加到最大的 9×9 ，以便计算中值。但是，如果在 9×9 的窗口中仍然没有未被破坏的像素，那么就计算 9×9 窗口中所有像素的平均值来替代噪声像素。这个滤波器性能更好的主要原因是它的可变窗口大小。通过这种方式，它提供了比 MDBUTMF[11] 更好的质量指标。然而，大窗口尺寸也是造成模糊效果的原因。

3.6.3 对噪声密度敏感的平均介质过滤器

噪声密度范围敏感 (NDRS) 过滤算法在 [25] 中提出，噪声像素被替换成基于噪声密度值的平均值、中值或预处理值。NDRS 过滤器提供了一种独特的方法，即使在非常高的噪声密度下也能恢复被破坏的像素。在 [15] 中提出了一种四阶段中值过滤算法 (FSMA)。FSMA 算法在第一、第二、第三和第四阶段分别以较小的窗口大小、大窗口大小、运行平均和替换以前处理过的像素进行中值过滤。只有当当前阶段无法对候选的噪声像素进行去噪时，该算法才会进入下一阶段。FSMA 算法可以通过考虑前两个阶段的低噪声密度来有效地消除图像的噪声，而它可能需要所有阶段的高噪声密度。这个滤波器在消除 SAP 噪声的同时，还能更好地保护边缘，因为它在估计噪声像素的值时考虑了无噪声的像素。

3.6.4 基于平均数的快速切换 (FSBMM)

这个过滤器是另一个最先进的过滤器，它以组合的方式使用两种策略—平均数和中位数 [29]。未处理的像素被之前处理过的像素的平均值所取代。这是一个非常好的方法，可以提高过滤器的性能，但是这个过滤器有一个问题，就是当边界像素没有被中位数策略处理时，其值会被各自行/列中先前处理过的像素的值所取代。因此，在高噪声密度下，会出现像素的复制，导致图像质量下降。因此，这个滤波器在高噪声密度条件下的性能很低。

3.6.5 非对称修剪中点滤波器 (UTMP)

这个过滤器是 DMF 过滤器的延伸，是针对噪声密度范围的过滤器。在这个滤波器中，中位数策略被用来评估去噪像素的值，这在高噪声密度条件下表现良好。

3.6.6 非对称修剪中值滤波器 (UTMF)

它也是 DMF 滤波器的一个扩展，以实现噪声密度范围的特定过滤。这个滤波器使用平均策略评估去噪像素的值，在较低的噪声密度下有更好的去噪性能。但这个滤波器在每个噪声密度范围都不能适应。

3.6.7 自适应修剪滤波器

在 [13] 中提出了一个自适应修剪中值 (ATM) 滤波器，可以有效地去除被高噪声密度破坏的图像的噪声。该滤波器通过计算自适应大小窗口的无噪声像素的中值，以及在低噪声密度和高噪声密度下分别通过基于插值的程序计算来恢复噪声像素。此外，该算法使用最近处理的像素程序对候选像素进行去噪，特别是在使用内插法去噪效果不好的边界处。在 [22] 中提出了一个基于多程序最小-最大平均池的滤波器 (MMAPF)。在这个滤波器中，经过预处理后，噪声图像被分成两部分，并通过多层的最大和最小池化。最后，该算法将前面程序处理过的图像结合起来，并进行平均池化，以去除所有的残余噪声。[14] 中提出了一个基于自适应最小-最大值的加权中值 (AMMWM) 滤波器。该过滤器首先使用当前窗口的最小值和最大值计算两组高度相关的无噪声像素，然后确定这些组的加权中值来恢复噪声像素。如果当前窗口不能提供任何无噪声像素，该过滤器会增加窗口大小。然而，该过滤器可以在高噪声密度下恢复噪声像素，但它表现出较高的计算复杂性。

3.6.8 递归样条插值滤波器 (RSIF)

RSIF 滤波器 [28] 与 MDBUTMF[11] 相似，但该滤波器使用三次样条插值技术取未损坏像素的中值。由于三次样条插值是一种卓越的数学技术，它在低、中噪声密度范围内产生更好的结果。然而，由于缺乏估计去噪值的信息，这种滤波器在高噪声密度条件下不能产生良好的效果。还有一个需要满足的约束条件是，在窗口中至少要有两个未被破坏的像素来应用插值，而在高噪声密度的情况下很少有这种情况。这个滤波器的另一个主要缺点是它需要更多的执行时间。

3.6.9 自适应切换加权中值滤波器

964 / 5,000 翻译结果翻译结果该过滤器主要使用两个条件来检测噪声像素，即窗口中的像素应该是“255”或“0”，其窗口均值分别不等于“255”和“0” [12]。它还使用可变窗口大小的概念。如果大小为 3×3 的处理窗口中无噪声可用像素的数量是奇数，则损坏的像素将被修剪窗口的中值替换。否则，重复窗口中具有最大出现次数的非噪声像素或最近的非噪声像素（如果没有具有最大填充可用的单个值），并且损坏的像素被结果窗口的中值替换。如果整个 3×3 的窗口都被损坏，则使用 5×5 大小的处理窗口执行上

述过程。该算法存在一个主要问题，即在高噪声密度下会出现条纹效应，因为最近的未损坏像素用于中值计算，主要导致直接替换。

3.6.10 三值加权滤波器 (TVWA)

TVWA 滤波器 [17] 是另一个加权滤波器。这个滤波器最初采用一个 3×3 的窗口，根据它们与中间、最大和最小像素值的相关性，将所有的噪声像素分成三组。这里使用的权重系数是根据所选窗口而变化的，并由这三组的长度决定。之后，被破坏的像素被窗口的加权平均值所取代。由于，这个算法使用了可变的窗口大小，所以如果 3×3 的窗口完全被破坏，那么窗口大小就会增加到 5×5 ，以处理后的破坏像素作为中心像素，并重复上述过程。最大的窗口尺寸是 7×7 。这个滤波器的主要问题是由于大窗口尺寸造成的模糊效应。

3.6.11 迭代非局部中值滤波器 (INLM)

这个过滤器使用非局部加权平均法与迭代法相结合来寻找被破坏的像素的值 [30]。该算法分三步工作。首先，创建一个具有噪声像素位置的矩阵。这个矩阵中的值 '0' 或 '1' 分别表示噪声像素和无噪声像素。噪声像素通过这个称为 NMap 的矩阵进行过滤。用于过滤的策略是基于开关的中值过滤器。这里，在 NMap 的帮助下，对噪声像素进行修补。在早期的基础上，具有相似性的噪声像素被修补在一起进行 INLM 处理 [30]。

3.6.12 基于决策的耦合窗口中值滤波器

这种滤波器使用可变的窗口大小来去除图像中的噪声 [4]。它是一种独特的去除高密度饱和脉冲噪声的方法。这个过滤器使用以下步骤来去除噪声。首先，它检查该像素是否被破坏。然后，取一个 3×3 的窗口，用窗口的中位数替换噪声像素，排除值为 '0' 或 '255' 的损坏像素。如果处理窗口没有无噪声像素，则使用 $2n+1$ where n 从 '1' 到 '4' 递增的方法增加窗口大小。根据窗口中存在的噪声类型，被破坏的像素的值被窗口的平均数或中位数所取代。由于它使用了可变窗口大小的概念，在高噪声密度范围内，其性能比 DBUTMF [11] 更好。由于可变窗口大小，在高噪声密度下，图像的质量更好。这个滤波器有一些局限性。增加处理窗口的大小，在最终的图像中会出现模糊或弄脏的效果。但由于动态地增加处理窗口的大小来处理被破坏的像素，所以比同类的其他过滤器表现得更好。

3.6.13 基于决策的非对称修剪修正的 winsorized 均值滤波器

这个滤波器通过使用修剪后的平均值和排序后的一维阵列来去除盐和胡椒的噪声 [27]。首先，选择一个以噪声像素为中心的 3×3 窗口，将其转换为一个排序的一维阵列。最后，从后一个阵列中去除噪声。这就产生了两种情况，在第一种情况下，结果数组中的一些像素被留下（通过替换修剪过的窗口中最小和最大的值进行不对称的赢余化操作），处理过的噪声像素被不对称赢余化操作后得到的数组的平均值替换，从而消除了图像中的噪声。在第二种情况下，剩余的阵列点导致空值，即初始窗口没有无噪声像素，为了适应这种情况，初始窗口的平均值被取走，并被放置在所选处理窗口的中心像素（处理像素）。在高

噪声密度 95% 的情况下，过滤器的性能会降低，因为更多的被选窗口将是完全没有噪声的，因此其平均值将只是一个随机值。图 2 说明了各种滤波器的去噪过程。

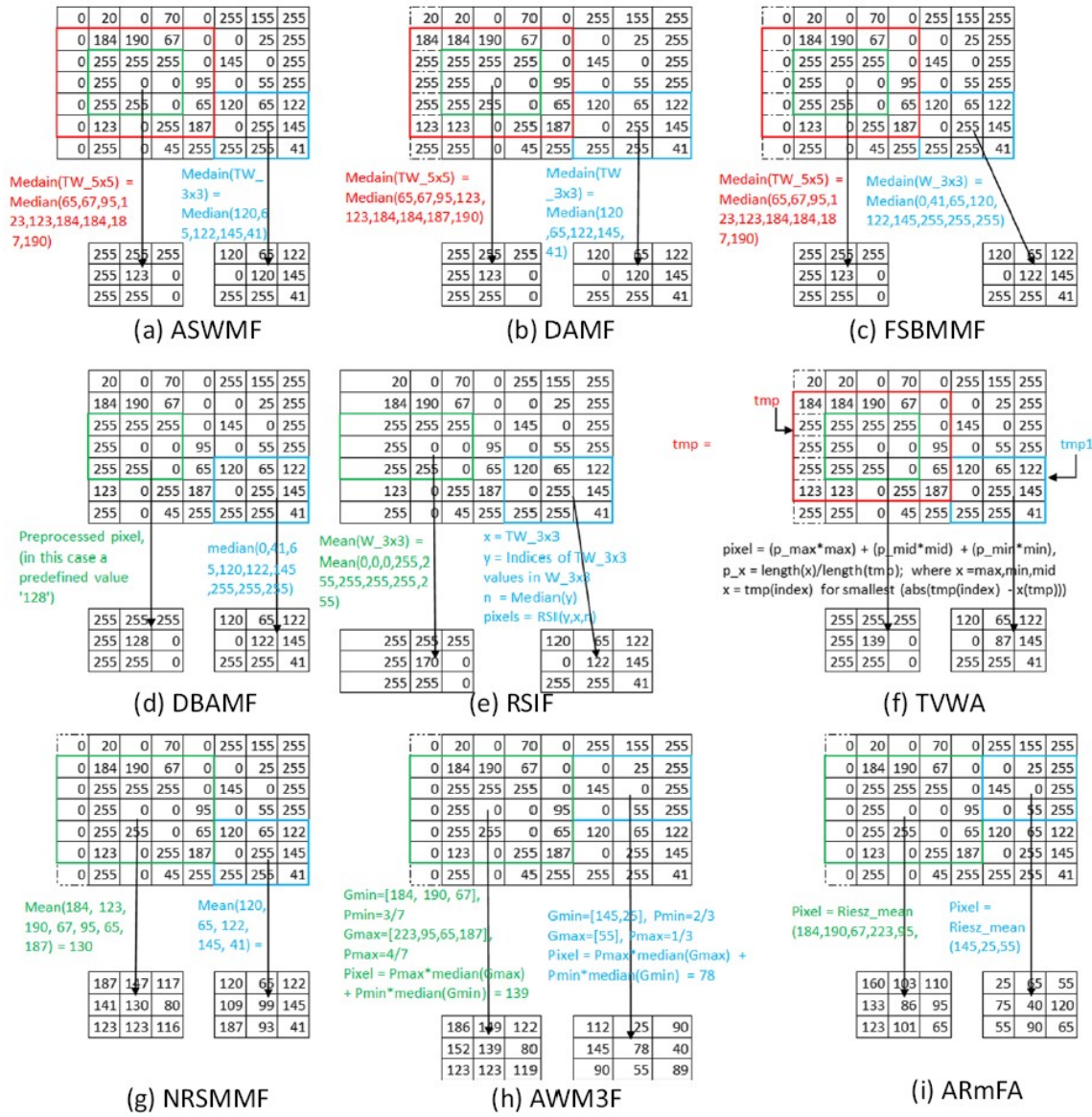


图 2: 使用 (a)ASWMF,(b)DAMF,(c)FSBMMF,(d)DBAMF,(e)RSIF,(f)TVWA,(g)NRSMMF,(h)AWM3F (i)ARmF 滤波器的去噪图示。黑色操作适用于两个箭头指示器，值已四舍五入到最接近的整数

3.7 过滤器选择

表 1 总结了上述的滤波技术及其数学公式。这些过滤器是在详细分析了该领域的各种研究后选择的。选择的基本标准如下。

- 方法的复杂性 (基础数学到高级数学)。
- 时间的复杂性。
- 基于不同性能指标的仿真结果。

表 1: 用于去除盐和胡椒噪声的各种滤波器的数学公式

Fileters	Equation	Explanation
FSBMMF	$\begin{cases} S_{med}, & \text{if } X_{\min} < S_{med} < X_{\max} \\ X_{i,j-1}; & \text{if } i = 1 \text{ or } i = N \\ X_{i-1,j}; & \text{if } j = 1 \text{ or } j = N \\ \text{mean (four PPP in } S_{i,j} \text{)} \end{cases}$	<p>S_{med} 是 $S_{i,j}$ 的中位数, 其中 $S_{i,j}$ 是 3x3 的窗口 (中位数 (W^{nf}))</p> <p>$X_{i,j}$ 是 $S_{i,j}$ 的中心像素, W^{nf} 是无噪声处理窗口</p> <p>PPP:-以前处理过的像素</p>
DBUTMWMF	$\begin{cases} W_{\text{mean}} (TW_{i,j}); & \text{if } W_{i,j} \text{ has NNP} \\ \text{mean} (W_{i,j}); & \text{Otherwise} \end{cases}$	<p>W_{mean} 是赢化均值, $TW_{i,j}$ 是修剪后的处理窗口</p> <p>$W_{i,j}$ 是 3x3 的处理窗口</p> <p>NNP - 无噪声像素</p>
MDBUTMF	$\begin{cases} \text{median} (TW_{i,j}); & \text{if } W_{i,j} \text{ has NNP} \\ \text{mean} (W_{i,j}); & \text{Otherwise} \end{cases}$	<p>$W_{i,j}$ 是尺寸为 3x3 的处理窗口</p> <p>$TW_{i,j}$ 是尺寸为 3x3 的修剪窗口</p> <p>NNP:-非噪声像素</p>
UTMF	$\begin{cases} \text{mean} (TW_{i,j}); & \text{if } W_{i,j} \text{ has NNP} \\ \text{PPP}; & \text{Otherwise} \end{cases}$	<p>$TW_{i,j}$ 是尺寸为 3x3 的修剪窗口</p> <p>PPP:-以前处理过的像素</p>
UTMP	$\begin{cases} \text{median} (TW_{i,j}); & \text{if } W_{i,j} \text{ has NNP} \\ \text{PPP}; & \text{Otherwise} \end{cases}$	<p>PPP:-以前处理过的像素</p> <p>Z: 初始尺寸为 3x3 的修剪窗口中的奇数向量, 水平和垂直方向的数值重复两次</p> <p>Y: Z 中重复数值最多的向量 (如果有的话), 否则为最接近的 NNP。</p>
ASWMF	$\begin{cases} \text{median}(Z) & \text{if } W_{3 \times 3} \text{ has NNP} \\ \text{median}(Y) & \text{if size of } Z \text{ even} \end{cases}$	<p>如果 $W_{3 \times 3}$ 没有 NNP, 那么尺寸就增加到 5x5, 在最近的 + 方向的像素重复 4 次, 在 W 方向重复 3 次, 其他方向重复 2 次, 并转换为矢量 Z。</p> <p>$P_x = \text{length}(x)/\text{length}(\text{tmp})$。其中 $x = \text{max}, \text{min}, \text{mid}$</p> <p>$x = \text{tmp}(\text{index})$ for smallest ($\text{abs}(\text{tmp}(\text{index}) - x(\text{tmp}))$)</p> <p>max, mid 和 min 是 tmp 中的最大值、最小值和中间值</p>
TVWA	$(p_{\max} \times \max) + (p_{\text{mid}} \times \text{mid}) + (p_{\min} \times \min)$	<p>其中 tmp 是处理窗口的修剪后的一维矢量。</p> <p>$w_{i,j,k,l}^{(t)} = \exp \frac{\ P(X^{(t-1)}) - P(Y_{k,l})\ _2^2}{h^2}$</p> <p>$X_{i,j}$ 是使用预先过滤的结果初始化的输入图像。</p>
TVWA	$\frac{\sum_{k,l \in \Omega_{i,j}} w_{i,j,k,l}^{(t)} Y_{k,l}}{\sum_{k,l \in \Omega_{i,j}} w_{i,j,k,l}^{(t)}}$	<p>max, mid 和 min 是 tmp 中的最大值、最小值和中间值, 其中 tmp 是处理窗口的修剪后的一维矢量。</p> <p>$w_{i,j,k,l}^{(t)} = \exp \frac{\ P(X^{(t-1)}) - P(Y_{k,l})\ _2^2}{h^2}$</p> <p>$X_{i,j}$ 是使用预先过滤的结果初始化的输入图像。</p>

表 2: (continue)

Fileters	Equation	Explanation
DBCWMF	$\begin{cases} \text{median}(TW_n) & \text{if } W_n \text{ has NNP } \forall n < 5 \\ \text{mean}(W_1); & \text{Otherwise} \end{cases}$	<p>TW_n 是修剪过的处理窗口, 尺寸为 $(2n+1)X(2n+1)$</p> <p>W_n 是大小为 $(2n+1)X(2n+1)$ 的处理窗口</p> <p>NNP:-非噪声像素</p> <p>DWM: 定向加权中值操作</p>
MDWMF	$DWM(TW_{7X7})$	<p>以中心像素为中心, 可能的方向总数为 12 个</p> <p>TW_{7X7} 是修剪过的 7x7 尺寸的窗口。</p>
Mean Filter	$\frac{1}{mn} \sum_{(s,t) \in S} g(s, t)$	<p>$g(s,t)$ 是原始图像</p>
Weighted Mean Filter	$\frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s,t) * g(s+x,y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s,t)}$	<p>$g(s,t)$ 是原始图像</p> <p>$W_{i,j}$ 是用户定义大小的处理窗口 (最可能是 3X3)。</p>
Adaptive Median Filter	$median(w_{i,j})$	<p>NNP:-非噪声像素, $RSCI$:-递归立方曲线插值法。</p>
RSIF	$P_{i,j} = \begin{cases} RCSI(y, x, n) \\ \text{mean}(W_{3 \times 3}), & \text{if } W_n \text{ has no NNP} \end{cases}$	<p>$y=x$ 中像素的指数, $x=W_{3X3}$ 中的非噪声像素</p> <p>W_{3X3} 是尺寸为 3X3 的处理窗口。</p>

本研究使用的特定过滤器在不同的噪声密度上显示了显著的模拟结果。

下一节介绍了对这些最先进的盐和胡椒噪声去除技术的比较分析。

3.8 仿真环境和比较分析

下面的小节给出了一些最先进的盐和胡椒噪声去除技术在不同噪声密度的基准图像上的定性和定量分析。所有现有的过滤技术都在 MATLAB 中实现, 并以基准输入进行模拟。最初, 在这些基准图像中加入不同噪声密度的 SAP 噪声 (宽范围: 10% 到 90%, 非常高的噪声密度范围: 91% 到 99%), 然后通过这些过滤技术对产生的噪声图像进行过滤, 并提取质量指标。最后, 根据提取的质量指标和重建的图像, 分别进行定量和定性分析。图 3 是用于分析的一些基准图像, 以及柯达 (24 张图像) 图像。对以下图像的模拟结果进行了讨论: