“基于极值检测的中值滤波算法”论文第四章研究报告

ZYH,YQF

28th June 2024

1 论文目的

在数字图像处理领域，图像去噪一直是一个核心问题，其目的在于从受噪声污染的图像中恢复出尽可能清晰的原始图像。中值滤波作为一种常见的非线性滤波技术，因其简单高效而被广泛应用于图像去噪中。然而，传统中值滤波算法在处理特定类型的噪声（如椒盐噪声）时，虽能取得较好的去噪效果，但往往会牺牲图像的细节信息，尤其是在图像边缘区域。第四章“基于极值检测的中值滤波算法”的研究，正是基于此问题背景，旨在通过算法改进，实现去噪与保护图像细节的双重目标。

2 研究方法

1. 基于极值的噪声检测

该章节首先介绍了基于极值检测的噪声识别方法，包括基本极值检测法、显著极值检测法、两极门限检测法及多窗口组合噪声检测等多种策略。这些方法通过分析图像中像素值的极值特性，有效区分噪声与非噪声像素，为后续的噪声过滤提供了准确的判断依据。

2. 误检像素的修正处理

针对噪声检测过程中可能出现的误检情况，本章提出了相应的误检像素修正策略。通过对比误检像素与其邻域像素的相关性，采取适当的修正措施，旨在减少误检带来的图像质量损失，特别是对图像细节与边缘的保护。

3. 改进的中值滤波算法

在噪声检测与误检修正的基础上，本章进一步提出了改进的中值滤波算法。该算法通过综合考虑图像的局部特性及噪声分布情况，动态调整滤波策略，不仅在去除噪声方面表现优异，更重要的是在保护图像细节和边缘信息方面做到了显著改进。

3 具体实现方法

1. 算法设计原理

该改进的中值滤波算法，首先依据前文所述的极值检测机制精确识别出图像中的噪声像素。随后，通过一个针对误检进行优化的修正过程，确保图像中的非噪声像素得以保留，从而减少对图像细节的不必要损害。在此基础上，算法进一步通过分析图像的局部特性（如纹理复杂度、边缘强度等）和噪声的分布模式（包括噪声类型、密度等），动态调整滤波窗口的大小和形状，以及滤波操作的具体实施策略。

2. 动态调整滤波策略

具体而言，改进算法在处理图像的不同区域时，会根据该区域内的特性选择合适的滤波参数。例如，在图像的平滑区域，噪声像素较易于被识别，此时算法可能采用较大的滤波窗口以提高去噪效率；而在细节或边缘区域，为了避免过度平滑导致细节丢失，算法则优先选择较小的滤波窗口，并可能结合图像的梯度信息来调整滤波策略，以更好地保留这些重要信息。

3. 保护图像细节和边缘信息

在细节和边缘保护方面，改进的中值滤波算法采用了局部自适应的处理方式，能够在去除噪声的同时，尽量保持图像的局部特征不被破坏。这一点是通过在滤波过程中对图像的局部纹理和边缘特征进行分析，并据此优化滤波操作来实现的。例如，算法在检测到边缘或细节特征较为显著的区域时，会减小滤波强度或调整滤波方向，以避免边缘被过度模糊。

综上所述，该章节提出的改进中值滤波算法，通过对图像局部特性的细致分析和滤波策略的动态调整，成功地实现了去噪与保护图像细节的双重目标，为中值滤波技术在复杂图像处理应用中的进一步优化和应用提供了有力的理论支持和实践指导。

4 图像质量评价方法

在该章节中，图像质量评价主要采用的方法是峰值信噪比（PSNR）和均方误差（MSE）。PSNR 是一种评估图像压缩质量的常用标准，它是通过比较原始图像和处理后的图像来衡量的，通常表示为对数形式的分贝单位。PSNR 越高，说明图像质量越好。MSE 则是原始图像与处理后的图像之间像素强度差异的平方和的平均值，用于衡量图像质量的退化程度。MSE 值越小，表示图像质量损失越少。

为了分析滤波算法的性能，论文中提出了一个误检噪声像素灰度值的修正恢复过程，以此来提高滤波图像的质量。这个过程包括利用误检率问题，恢复被误检像素的灰度值，以提高滤波后图像的质量，并在仿真实验中

验证了这些改进算法的效果。论文中也提到，人眼的视觉感知与图像质量的好坏密切相关，尤其是人眼对亮度的感知主要是通过杆状细胞来完成的。因此，中值滤波技术的应用与人眼的感光特性有一定的相关性。

总结来说，该章节的图像质量评价方法主要侧重于 PSNR 和 MSE 指标，以及对这些指标的改善方法和与人眼视觉特性的关联性研究。通过实验结果的分析，作者能够说明提出的滤波算法在去噪和保持图像细节方面相比于传统方法的优势。

5 论文算法复现

我们采用 Python 编程语言结合 OpenCV 库来实现了改进的中值滤波算法。算法的核心是对图像中的每一个像素进行检查，判断其是否受到椒盐噪声的影响。如果判断为噪声像素，我们则通过中值滤波的方式对该像素进行处理，从而达到去噪的目的。

Figure 1: 处理前的图像

Figure 2: 处理后的图像

为了评估改进的中值滤波算法的性能，我们对比了处理前后图像的峰值信噪比（PSNR）和均方误差（MSE）指标。实验结果表明，相比于传统的中值滤波算法，我们的改进算法在去除椒盐噪声的同时，更好地保留了图像的细节信息，展示了出色的性能和实用价值。

通过这一系列的实验和分析，我们验证了改进中值滤波算法在图像去噪中的有效性和优越性。未来的工作将探索更多的图像处理技术，以进一步提高图像质量，并扩展到更多的应用场景中。

6 所得经验

首先，通过对第四章“基于极值检测的中值滤波算法”内容的仔细阅读与理解，我对中值滤波处理椒盐噪声的有效机制有了更深刻的认识。特别是，该算法通过将像素值替换为其邻域内像素值的中位数来实现去噪，这在针对随机分布的亮点或暗点噪声时显示出了特别的有效性。这篇论文不仅加深了我对中值滤波基础概念的理解，而且还启发了我如何根据噪声的具体特征来选择适当的滤波策略。

其次，改进算法体现了噪声识别在图像预处理中的核心作用。准确地识别图像中的噪声像素对于提升滤波效率及保留图像细节极为关键。这一发

现对我来说颇具启发性，它让我意识到在执行图像滤波处理前，进行细致的噪声分析和识别是一个不可或缺的步骤。

进一步地，从论文中我还学习到自适应滤波技术的重要性。自适应技术能够根据图像内容的变化自动调整滤波策略，既能有效去除噪声，又能在更大程度上保留图像细节。这种方法对我来说极具吸引力，因为它展示了图像处理技术的灵活性和进阶应用的可能性。

最后，通过这篇论文，我学会了使用峰值信噪比（PSNR）和均方误差（MSE）这两种评价指标来衡量图像处理算法的性能。这提供了一种量化的方法来分析图像处理效果，帮助我深入理解不同算法或参数设置对最终结果的影响，是我学习图像处理过程中的一项重要技能。

附件

1. 代码展示

以下是本研究中使用的 Python 代码片段：

```python
import numpy as np
import cv2


def is_salt_pepper_noise(image, i, j, H):
    pixel_value = image[i, j]
    return pixel_value >= 255 - H or pixel_value <= 0 + H


def pixel_median_filter(image, i, j, H):
    kernel_size = 3  # 使用3x3的滤波核
    half_kernel = kernel_size // 2
    rows, cols = image.shape
    pixel_values = []

    for k in range(-half_kernel, half_kernel + 1):
        for l in range(-half_kernel, half_kernel + 1):
            if 0 <= i + k < rows and 0 <= j + l < cols and not is_salt_pepper_noise(image, i + k, j + l, H):
                pixel_values.append(image[i + k, j + l])

    if len(pixel_values) == 0:
        return image[i, j]
    else:
        return np.median(pixel_values)


def image_filter(image, H):
```

```
    filtered_image = image.copy()
    rows, cols = image.shape

    for i in range(rows):
        for j in range(cols):
            if is_salt_pepper_noise(image, i, j, H):
                filtered_image[i, j] =
                    pixel_median_filter(image, i, j, H)

    return filtered_image


image = cv2.imread('original_grey_image.png',
    cv2.IMREAD_GRAYSCALE)
H = 20
filtered_image = image_filter(image, H)

cv2.imshow('Original Image', image)
cv2.imshow('Filtered Image', filtered_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```