Evaluation of Salt and Pepper Noise Removal Techniques: A Comparative Study of Standard Median Filtering vs. Switch-Based Median Filtering

Anirudh Narasimha Bharadwaj

Abstract

This report compares two denoising methods - 1.Matlab's standard median filter (`medfilt2`) and 2.a custom switch-based median filter (`switchmedfilt2`) for removing salt and pepper noise from digital images. Two test images, **cameraman** and **einstein**, were used with noise densities varying from 10% to 40%. In addition to comparing static filtering results, the effect of recursively applying `switchmedfilt2` at high noise levels (`40%`) is seen by plotting the SNR over iterations. The comparision and analysis show that the switch-based filter is more robust at higher noise levels while preserving details better than the standard median filter.

Contents

1 Introduction

Salt and pepper noise, which introduces black and white pixels randomly into an image is a major challenge in digital image processing. Though the standart median filter (`medfilt2` in Matlab) is widely used for noise cancelling, it may blur fine details in the image. In comparison, the switch-based median filter (`switchmedfilt2`) applies filtering only to pixels detected as noisy maintaining the unaffected areas.

The goal of this study is to compare the two filtering methods under different noise densities and to assess the benefits of recursively applying the switch-based filter. Two images, **cameraman** and **einstein**, were used for the experiments.

2 Methodology

2.1 Experimental Setup

Two test images (**Cameraman** and **Einstein**) were introduced with salt and pepper noise at four different densities: 10% (0.1), 20% (0.2), 30% (0.3), and 40% (0.4). For each noise :

- A noisy image was generated and saved as `noisy.png`.
- Both filters, `medfilt2` and `switchmedfilt2`, were applied. Their outputs were saved as `denoised_medflit.png` and `denoised_switch.png` respectively.
- The images noisy.png, `denoised_medflit.png` and `denoised_switch.png` were later stitched together.
- SNR values were computed using a script (`mysnr.m`), ignoring border pixels to suppress edge effects.

2.2 Recursive Filtering

At high noise level (40%), the `switchmedfilt2` filter was applied recursively (for 10 iterations) and the SNR was computed after each iteration. These SNR values were plotted against the iterations to analyze the performance.

3 Experimental Results

3.1 cameraman Results

The following figures illustrate the results for the **cameraman** image.

Noise Level: 10% (0.1)

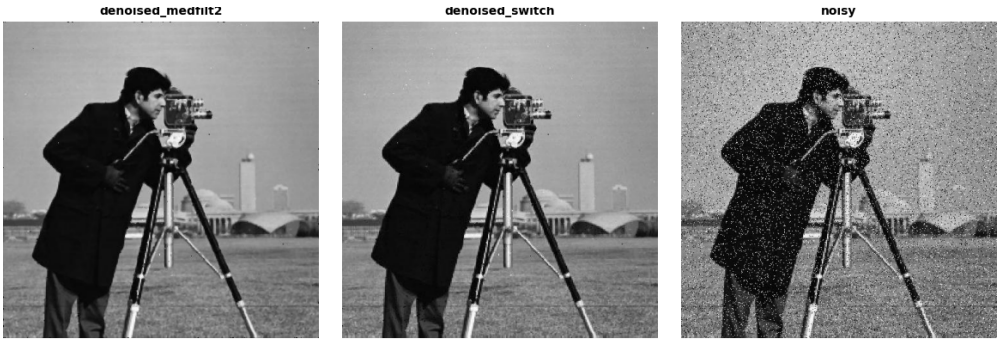

Figure 1: cameraman - noise=0.1.

Noise Level: 20% (0.2)

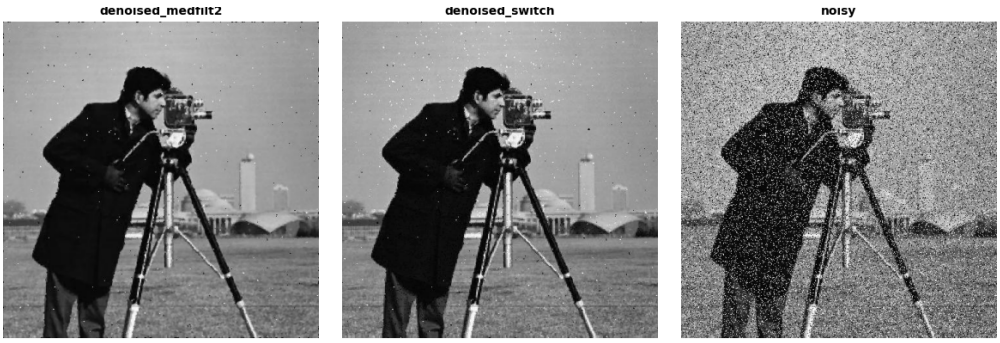

Figure 2: cameraman - noise=0.2.

Noise Level: 30% (0.3)

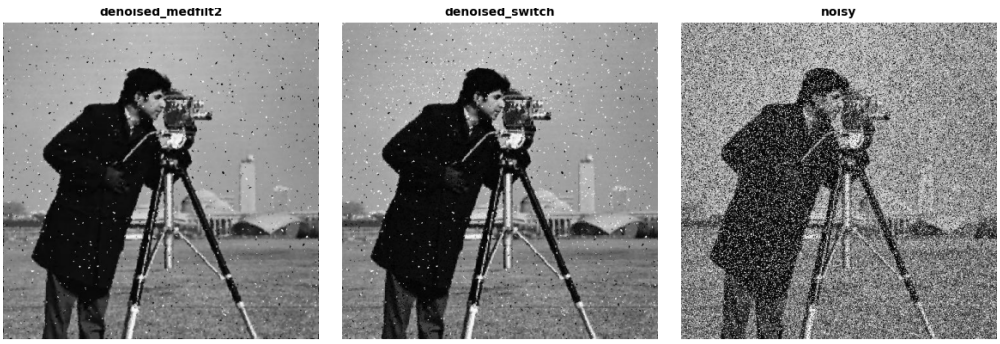

Figure 3: cameraman - noise=0.3.

Noise Level: 40% (0.4)

Figure 4: cameraman - noise=0.4.

3.2 einstein Results

The results for the **einstein** image are shown below.

Noise Level: 10% (0.1)

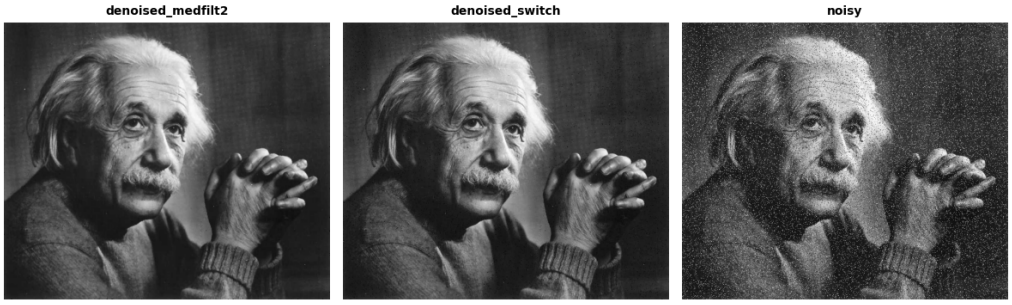

Figure 5: einstein - noise=0.1.

Noise Level: 20% (0.2)

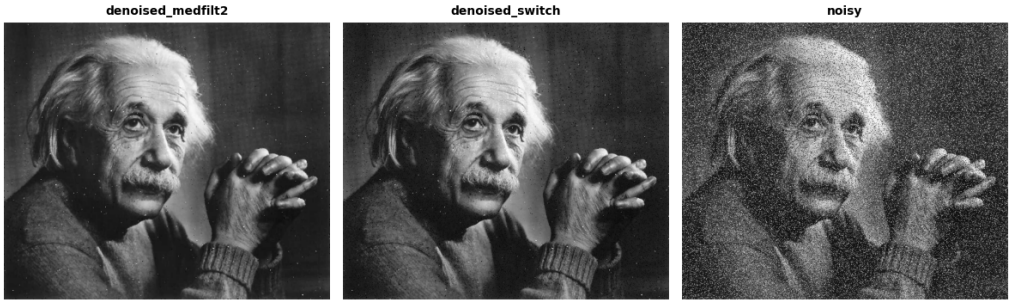

Figure 6: einstein - noise=0.2.

Noise Level: 30% (0.3)

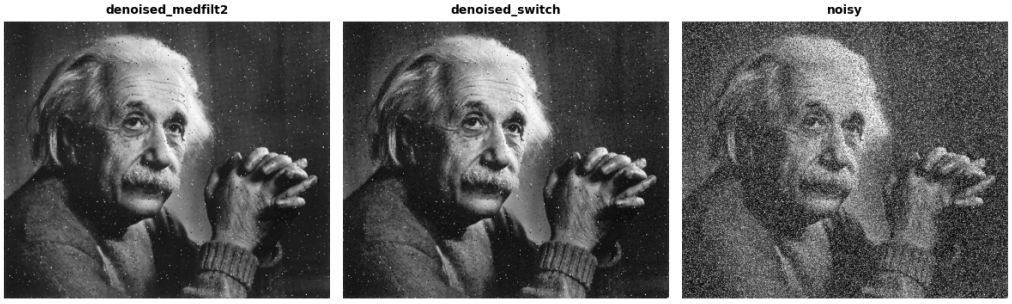

Figure 7: einstein - noise=0.3.

Noise Level: 40% (0.4)

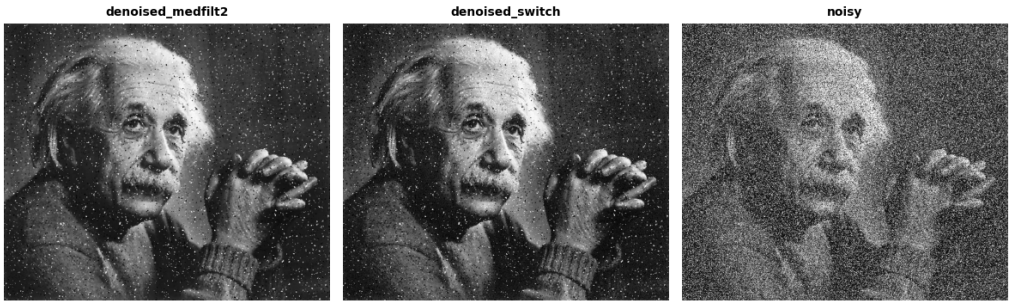

Figure 8: einstein - noise=0.4.

3.3 Analysis of Signal to Noise Ratio(SNR)

For each image, the following figures gives the SNR performance.

Cameraman

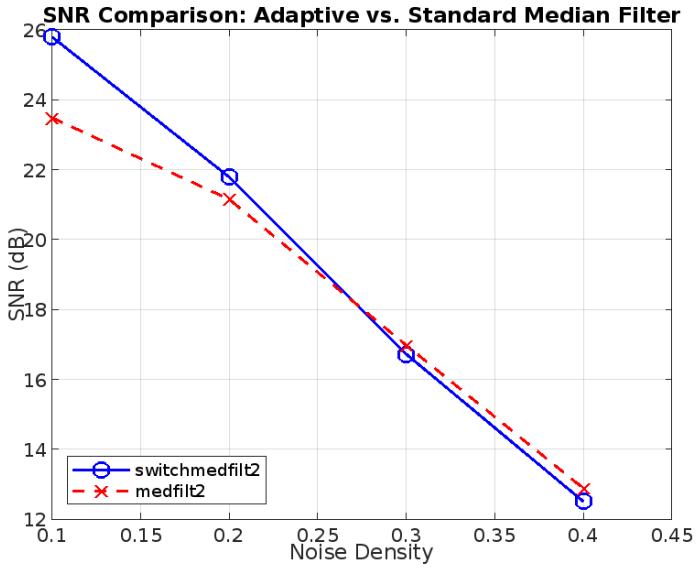

Figure 9: cameraman - SNR vs Noise Density.

	Noise Density	switchmedfilt2 (dB)	medfilt2 (dB)
1	0.1000	25.7940	23.4815
2	0.2000	21.7848	21.1498
3	0.3000	16.7207	16.9737
4	0.4000	12.5035	12.8798

Figure 10: cameraman - SNR Comparison Table for `medfilt2` and `switchmedfilt2`.

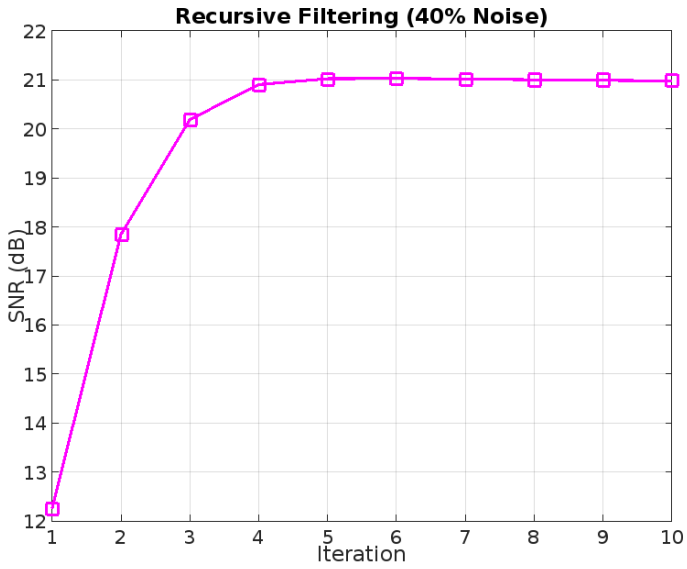

Figure 11: cameraman - SNR vs Iteration of `switchmedfilt2`.

Einstein

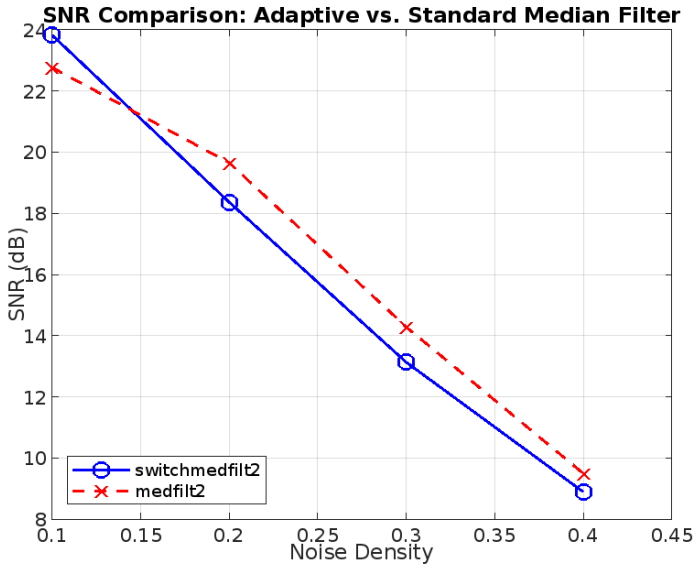

Figure 12: einstein - SNR vs Noise Density.

	Noise Density	switchmedfilt2 (dB)	medfilt2 (dB)
1	0.1000	23.8248	22.7655
2	0.2000	18.3616	19.6520
3	0.3000	13.1284	14.2806
4	0.4000	8.8883	9.4908

Figure 13: einstein - SNR Comparison Table for `medfilt2` and `switchmedfilt2`.

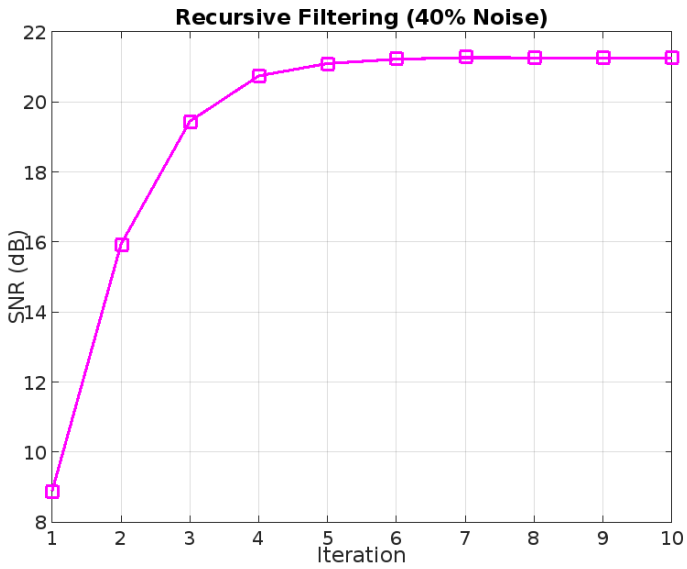

Figure 14: einstein - SNR vs Iteration of `switchmedfilt2`.

4 Analysis and Discussion

The performance charts and tables provide a clear picture of how both filters behave under noise ranging from 0.1 to 0.4.

4.1 SNR vs. Noise Density

Figures 9 and 12 show that at low noise levels (10% and 20%), both `medfilt2` and `switchmedfilt2` produce similar SNR values. As the noise increases to 30% and 40%, the SNR for `switchmedfilt2` remains higher, indicating better noise removal while preserving details.

4.2 SNR Comparison Tables

The SNR values in Figures 10 and 13 confirm the trends seen in the SNR vs. Noise Density graphs. The switch-based filter consistently achieves higher SNR, especially at higher noise levels, highlighting its performance compared to the standard median filter.

4.3 Recursive Filtering Analysis

The recursive SNR plots (Figures 11 and 14) illustrate that applying `switchmedfilt2` iteratively results in an initial increase in SNR. However, roughly after 4 to 5 iterations the improvements plateau, suggesting that further recursion yields diminishing returns and may risk over-smoothing of image. This behavior indicates the importance of determining an optimal number of iterations.

4.4 Overall Observations

- **Edge Preservation:** The selective nature of `switchmedfilt2` allows for better preservation of edges and fine details.

- **Robustness at High Noise Levels:** While both filters perform well at low noise levels, `switchmedfilt2` outperforms `medfilt2` at higher noise densities.
- **Recursive Improvement:** Recursive application of the switch-based filter improves SNR up to a point, after which further iterations provide negligible benefits.

5 Conclusion

This study compared Matlab's standard median filter and a custom switch-based median filter for salt and pepper noise removal. The experimental results demonstrate that while both methods perform similarly at low noise levels, the switch-based approach is more effective at higher noise levels by maintaining higher SNR values and preserving details. Recursive application of `switchmedfilt2` further enhances noise removal, although the benefits reduce off after about 4 to 5 iterations.