Wavelet Based Image Denoising

Tejash More, Dwaipayan Haldar
M Tech, Signal Processing
Indian Institute of Science

November 16, 2025

Introduction

- Spatial denoising smooths pixels directly, so it often blurs edges and fine textures, especially at higher noise levels. It cannot distinguish noise from real high-frequency details.
- While wavelet based denoising, noise is spread mostly as low coefficients values in the wavelet domain. Applying the threshold removes the low coefficient noise while preserving the high coefficient edges.
- The basic workflow of our denoising algorithm works as follows:

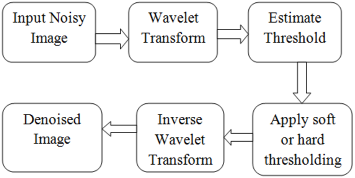


Figure 1: Flowchart of the Overall Algorithm

The Wavelet Transform

- The Discrete Wavelet Transform (DWT) decomposes an image into multiple frequency subbands (LL, HL, LH, HH), capturing coarse structure and fine details at different scales.
- It provides a sparse representation where important features concentrate in a few large coefficients.
- Noise mostly appears in the high-frequency bands, making it easy to remove using thresholding.

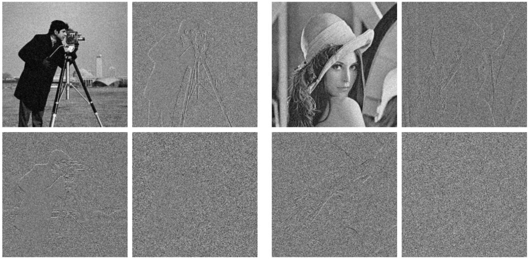


Figure 2: LL,HL,LH,HH representation for cameraman and lena with Haar Wavelet (1-Level)

Thresholding / Shrinking Algorithms

VisuShrink

- Simplest of all. Oversmooths. Uses universal threshold: $T = \sigma\sqrt{2\log n}$, where σ is estimated by the robust median estimator $\hat{\sigma} = \frac{Median(|Y_{ij}|)}{0.6745}$, $Y_{ij} \in$ subband HH1

SUREShrink

- Uses Stein's Unbiased Risk Estimate(SURE) for estimating bayesian risk. Picks the threshold that minimizes $SURE(t; \mathbf{x})$. It is given by:

$$T_S = argmin_{t\geq 0} SURE(t; \mathbf{x}) = d - 2\cdot\sum_{i=1}^{d} \mathbb{1}_{|x_i|\leq t} + \sum_{i=1}^{d} min(x_i^2, y^2)$$

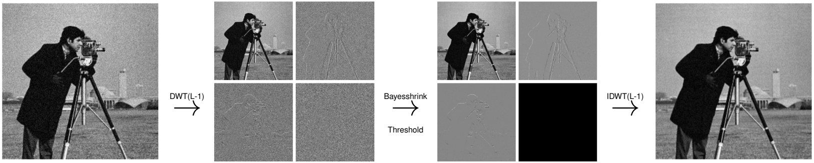
BayesShrink

- Minimizes the Bayesian risk using a GGD prior.

$$T_B = \frac{\sigma^2}{\sigma_X}; \hat{\sigma}_X = \sqrt{max(\hat{\sigma}_Y^2 - \hat{\sigma}^2, 0)}; \hat{\sigma}_Y^2 = \frac{1}{n^2}\sum_{i,j=1}^{n} Y_{ij}^2$$

Cameraman: Complete Workflow

The complete workflow is given cameraman image. Level-1 wavelet decompostion is shown for easy visualization. Experiments are carried on Level-4. Here Haar filter is used. Denoising is done using BayesShrink.



(a) (b) (c) (d)

Figure 3: Denoising Using wavelet Transform Flowchart: (a) Denoised Image, (b) DWT with Haar Wavelet Level-1, (c) Thresholded Wavelet coefficients using BayesShrink threshold, (d) Denoised Image

Cameraman: Comparison with Haar Wavelet

(a) SSIM: 27.99, PSNR: 20.57

(b) SSIM: 70.96, PSNR: 23.97

(c) SSIM: 71.26, PSNR: 27.97

(d) SSIM: 68.62, PSNR: 27.75

(e) SSIM: 43.78, PSNR: 25.02

(f) SSIM: 75.87, PSNR: 25.72

(g) SSIM: 84.09, PSNR: 30.98

(h) SSIM: 79.21, PSNR: 30.69

Figure 4: Image Comparison: (a) Gaussian Noisy Image, (b) Denoised Image (Gaussian, VisuShrink), (c) Denoised Image (Gaussian, SureShrink), (d) Denoised Image (Gaussian, BayesShrink), (e) Uniform Noisy Image, (f) Denoised Image (Uniform, VisuShrink), (g) Denoised Image (Uniform, SureShrink), (h) Denoised Image (Uniform, BayesShrink)

Cameraman: Comparison with Db4 Wavelet

(a) SSIM: 27.99, PSNR: 20.57

(b) SSIM: 71.88, PSNR: 24.91

(c) SSIM: 73.40, PSNR: 29.17

(d) SSIM: 73.00, PSNR: 29.28

(e) SSIM: 43.78, PSNR: 25.02

(f) SSIM: 76.87, PSNR: 26.89

(g) SSIM: 85.39, PSNR: 32.41

(h) SSIM: 81.69, PSNR: 32.09

Figure 5: Image Comparison: (a) Gaussian Noisy Image, (b) Denoised Image (Gaussian, VisuShrink), (c) Denoised Image (Gaussian, SureShrink), (d) Denoised Image (Gaussian, BayesShrink), (e) Uniform Noisy Image, (f) Denoised Image (Uniform, VisuShrink), (g) Denoised Image (Uniform, SureShrink), (h) Denoised Image (Uniform, BayesShrink)

Cameraman: Comparison with Spatial Filter Image

(a) SSIM: 71.88, PSNR: 24.91

(b) SSIM: 73.40, PSNR: 29.17

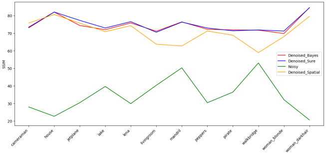
(c) SSIM: 73.00, PSNR: 29.28

(d) SSIM: 75.78, PSNR: 27.54

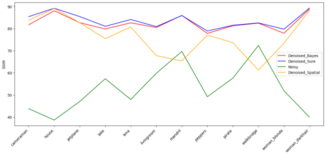
Figure 6: Image Comparison: (a) Denoised Image (Gaussian, VisuShrink), (b) Denoised Image (Gaussian, SureShrink), (c) Denoised Image (Gaussian, BayesShrink), (d) Denoised Image (Gaussian Spatial Filter)

Quantitative Comparison (Standard Image Dataset)

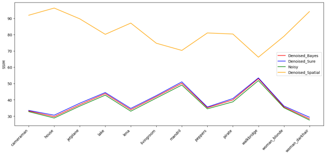
Comparison of SSIM plots of 12 standard test images with 2 Wavelet based thresholding methods(Db4) and spatial domain filtering(Gaussian Filter, Median Filter) is calculated for different Gaussian Noise, Salt and Pepper Noise and Uniform Noise.



(a)



(b)



(c)

Figure 7: (a) Gaussian, (b)Uniform, (c) Salt and Pepper Noise

Discussion

- **Wavelet:** One observation while operating with different mother wavelets is that *Db4* performs better than *Haar* in almost all the cases. That is expected since *Db4* is smoother and more correlated to natural images than *Haar* wavelet.
- **Noise:** While wavelet domain denoising performs pretty well for Gaussian and Uniform noise, it fails for Salt and Pepper Noise and lags much behind Median filters. This happens due to the prior assumption of Gaussian Noise in all the thresholding methods.
- **Thresholding:** SUREShrink and BayesShrink almost goes parallel to each other. That is also evident from the [5] where the author compares BayesShrink and SUREShrink, that comes out to be within 8% of each other. VisuShrink is universal thresholding it oversmooths images in most cases, giving below average results.

References

- S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA, USA: Academic Press, 1999.
- D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.
- D. L. Donoho and I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data," in *Proc. 16th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 1, 1994, pp. A24–A25.
- S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, 2000.

Thank You!