EE530 - FINAL PROJECT
WAVELET-BASED NOISE REMOVAL

Student: Rafael Lima
ID: 20155403

Reference Paper

IMACS Multiconference on "Computational Engineering in Systems Applications"(CESA), October 4-6, 2006, Beijing, China.

Wavelet Image Threshold Denoising Based on Edge Detection

Wei Liu [a], Zhengming Ma [b]
[a] Computer School, South China Normal University, Guangzhou, 510631, China
Phone: 862031772014, E-mail: buzzingbee@126.com
[b]Information Processing Lab., Electronics Dept., Zhongshan Univ., Guangzhou,510275, China

Abstract - Most commonly used denoising methods use low pass filters to get rid of noise. However, both edge and noise information is high-frequency information, so the loss of edge information is evident and inevitable in the denoising process. Edge

focused on solving this problem.

In the wavelet domain, the denoising algorithm based on the threshold filter [1] is widely used, because it's comparatively efficient and easy to realize. Wavelet

Background

Image Denoising:

- Removing noise from image (for both aesthetical and technical reasons);

- Basic idea is to reduce high frequencies.

Problem Statement

Both noise and edges are characterized by **high** frequencies.

Thus, it is difficult to remove noise from images without blurring the edges.

Review of Existing Methods

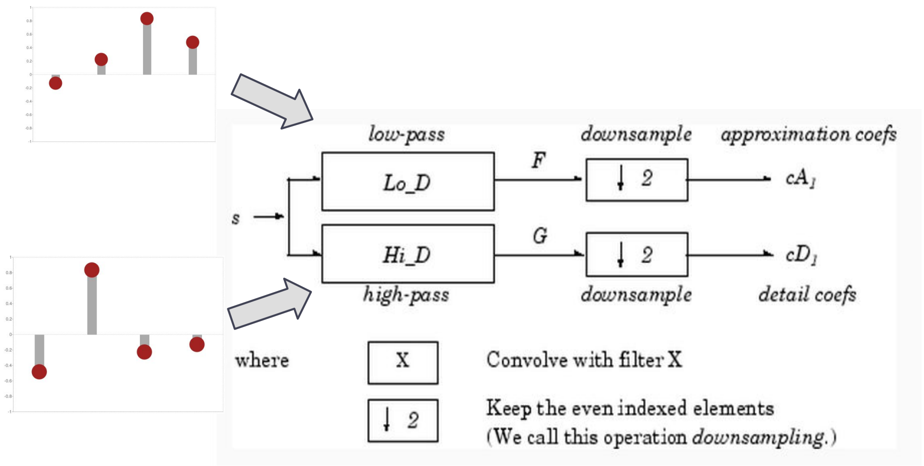
- In the wavelet domain, the denoising algorithm based on the threshold filter is widely used -> comparatively efficient and easy
- Wavelet decomposition transforms a signal from the time domain to the time-scale domain -> “shifting and scaling”
- Select a threshold according to the characteristics of the image, modifying the discrete detail coefficients so as to reduce the noise.
- Higher Threshold = Better Denoising + Blurrier Edges **(trade-off)**

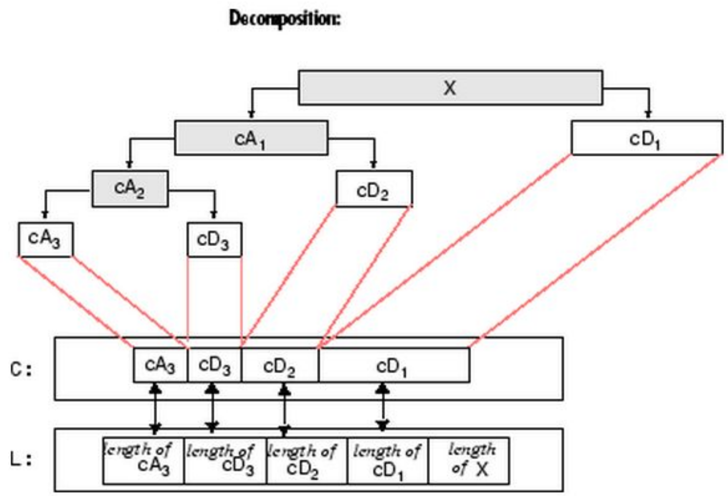
Method Proposed in the Paper

- Also a wavelet threshold-based method;
- But, before denoising, wavelet coefficients corresponding to an image's edges are first detected (through Lipschitz Exponents);
- Softer threshold is applied to these edges' coefficients and harder threshold is applied to the remaining coefficients.

Method Proposed in the Paper

Basics of Discrete Wavelet Decomposition:





(Source: www.mathworks.com)

Method Proposed in the Paper

- **Basic Algorithm:**

1. Detect the wavelet corresponding to the image's edges by the method of Lipschitz Exponent (α):

$$\alpha = \frac{\log_2 \left| \frac{Wf(s_{(i+1)},x)}{Wf(s_i,x)} \right|}{\log_2 \left| \frac{s_{(i+1)}}{s_i} \right|}$$

(α>0: Edge; α<0: Noise)

Wf(s_i,x): Value of the Wavelet component at coordinate "x" and decomposition level "i"

$s_i = 2^i$

(source: "Measurement of Lipschitz Exponent (LE) using Wavelet Transform Modulus Maxima (WTMM)", by Venkatakrishnan et al. (IJSER - June/2012).

Method Proposed in the Paper

- **Basic Algorithm:**

2. Preserve the coefficients corresponding to the edges;

3. Apply wavelet transform to the original noise-corrupted image;

4. Apply soft-threshold ($T = \beta^{*}\sigma^{*}sqrt(lnN)$) in the edge coefficients and hard-threshold ($T = \sigma^{*}sqrt(lnN)$) in the remaining ones;

5. Recompose the image from the thresholded components.

Parameters:

- $0.2 < \beta < 0.3$
- σ: Strength of the noise (related to the variance)
- N: Number of elements in the decomposition level

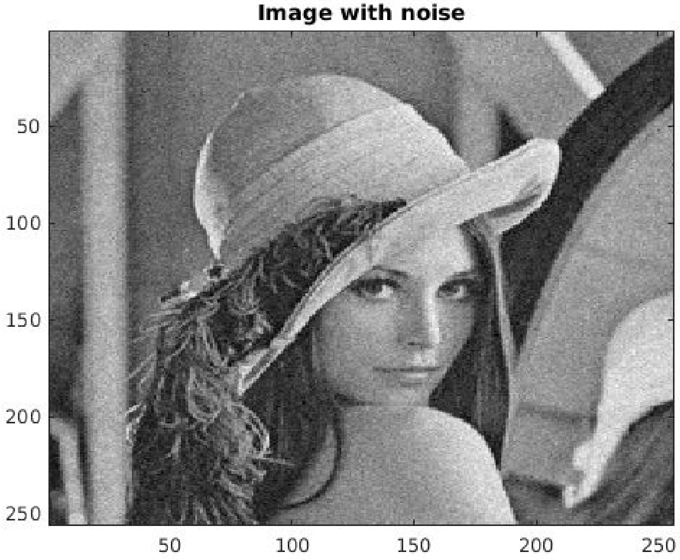
Why is this method better than others?

- If the edge-related coefficients are precisely detected, the remaining of the image can be subject to hard-thresholding without an excessive blurring of the edges (trade-off “higher denoising + blurrier edges” can be circumvented).

Result with examples

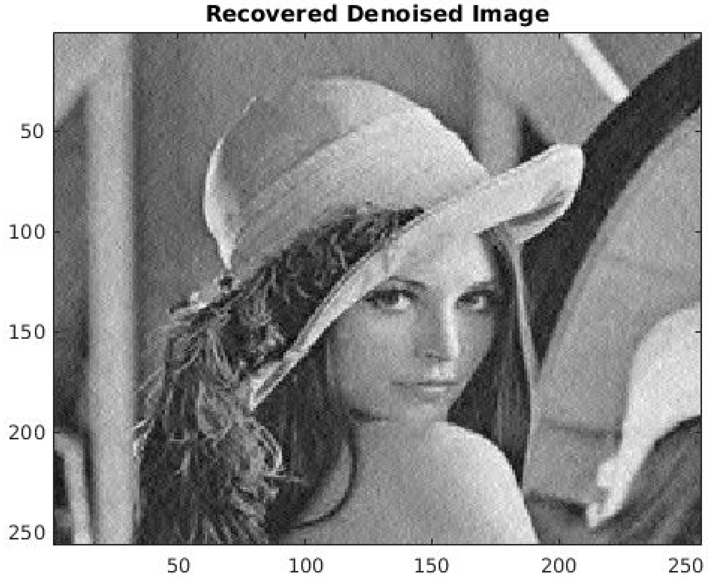
- The algorithm was implemented with a “Lena” 256x256 image;
- A code from the Mathworks website was used for comparison with both the hard and soft-threshold Visushrink denoising algorithms;
- For the comparisons: $\beta = 0.2$ and $\sigma = 10$.

Result with examples



SNR = 22.55 (dB)

Result with examples



SNR = 23.07 (dB)

Result with examples

SNR = 22.96 (dB)

Result with examples

SNR = 20.19 (dB)

Conclusion

Denoising Method	None	Implemented Algorithm	Visushrink Hard-threshold	Visushrink Soft-threshold
SNR (dB)	22.55	**23.07**	22.96	20.19

Proposal for possible improvement

- Use a more **efficient** detector for extracting the edges (Average Local Distance, Sobel Mask), since computing Lipschitz exponents is extremely inneficient in both memory usage and runtime.

- “Sharpen” and denoise the edge area through, perhaps, a median filter.

Thank You