IMAGE DENOISING USING NEIGHBOURING WAVELET COEFFICIENTS

G. Y. Chen, T. D. Bui and A. Krzyzak

Department of Computer Science,
Concordia University,
1455 De Maisonneuve West,
Montreal, Quebec, Canada H3G 1M8.
Email: $\{guang\_c, bui, krzyzak\}$@$cs.concordia.ca$.

ABSTRACT

The denoising of a natural image corrupted by Gaussian noise is a classical problem in signal or image processing. Donoho and his coworkers at Stanford pioneered a wavelet denoising scheme by thresholding the wavelet coefficients arising from the standard discrete wavelet transform. This work has been widely used in science and engineering applications. However, this denoising scheme tends to kill too many wavelet coefficients that might contain useful image information. In this paper, we propose one wavelet image thresholding scheme by incorporating neighbouring coefficients, namely *NeighShrink*. This approach is valid because a large wavelet coefficient will probably have large wavelet coefficients as its neighbours. Experimental results show that *NeighShrink* is better than the *Wiener* filter and the conventional wavelet denoising approaches: *VisuShrink* and *SUREShrink*. We also investigate different neighbourhood sizes and find that a size of 3×3 is the best among all window sizes.

1. INTRODUCTION

Wavelet transforms have been successfully used in many scientific fields such as image compression, image denoising, signal processing, computer graphics, and pattern recognition, to name only a few. Donoho and his coworkers pioneered a wavelet denoising scheme by using soft thresholding and hard thresholding. This can be summarized as follows. Let $A(i,j)$ be the noise-free image and $B(i,j)$ the image corrupted with white noise $Z(i,j)$, i.e., $B(i,j) = A(i,j) + \sigma Z(i,j)$, where $Z(i,j)$ has normal distribution $N(0,1)$. The Donoho's wavelet denoising scheme can be summarized as follows:

1. Transform the noisy image $B(i,j)$ into an orthogonal domain by 2D discrete wavelet transform.
2. Apply soft or hard thresholding to the resulting wavelet coefficients by using the threshold $\lambda = \sigma\sqrt{2\log n^2}$.
3. Perform inverse 2D discrete wavelet transform to obtain the denoised image.

This method performs well under a number of applications because wavelet transform has the compaction property of having only a small number of large coefficients. All the rest wavelet coefficients are very small. The denoising is done only on the detail coefficients of the wavelet transform. It has been shown that this algorithm offers the advantages of smoothness and adaptation. However, as Coifman and Donoho [1] pointed out, this algorithm exhibits visual artifacts: Gibbs phenomena in the neighbourhood of discontinuities. Therefore, they propose in [1] a translation invariant (TI) denoising scheme to suppress such artifacts by averaging over the denoised signals of all circular shifts. The experimental results in [1] confirm that single TI wavelet denoising performs better than the traditional single wavelet denoising. Bui and Chen [2] also proposed a TI multiwavelet denoising scheme that gave better results than the TI single wavelet denoising. Recently, several important approaches are proposed by considering the influence of other wavelet coefficients on the current wavelet coefficient to be thresholded. The motivation of this idea is that a large wavelet coefficient will probably have large wavelet coefficients at its neighbours. This is because wavelet transform produces correlated wavelet coefficients. Cai and Silverman [3] proposed a thresholding scheme by taking the immediate neighbour coefficients into account. Their experimental results showed apparent advantages over the traditional term-by-term wavelet denoising. Chen and Bui [4] extended this neighbouring wavelet thresholding idea to the multiwavelet case. They claimed that neighbour multiwavelet denoising outperforms neighbour single wavelet denoising for some standard testing signals and real-life images. Shengqian et al. [5] proposed an adaptive shrinkage denoising scheme by using neighbourhood characteristics. They claimed that their new scheme produced better results than Donoho's methods. Sendur and Selesnick [6] [7] proposed bivariate shrinkage functions for denoising. It is indicated that the estimated wavelet coefficients depend on the parent coefficients. The smaller the parent coefficients, the greater the shrinkage. Crouse et al. [8] developed a framework for statistical signal processing based on wavelet-domain hidden markov models (HMM). The framework enables us to concisely model the non-Gaussian statistics of individual wavelet coefficients and capture statistical dependencies between coefficients.

In this paper, we extend Cai and Silverman's idea to the image case. For images, we need to consider a neighbourhood window around the wavelet coefficients to be thresholded. We propose one way to thresholding the wavelet coefficients, namely $NeighShrink$. $NeighShrink$ thresholds the wavelet coefficients according to the magnitude of the square sum of all the wavelet coefficients within the neighbourhood window. Experimental results show that by using neighbouring coefficients $NeighShrink$ gets higher Peak Signal to Noise Ratio (PSNR) for all the denoised images. Also, we find that neighbour wavelet image denoising algorithm $NeighShrink$ outperforms *VisuShrink*, *SUREShrink* and *Wiener* filter.

The organization of this paper is as follows. We explain how to incorporate neighbouring wavelet coefficients into image denoising in Section 2. Experimental results are shown in Section 3. And finally we give the conclusion and future work to be done in section 4.

2. INCORPORATING NEIGHBOURING WAVELET COEFFICIENTS IN IMAGE DENOISING

The wavelet transform can be accomplished by applying the low-pass and high-pass filters on the same set of low frequency coefficients recursively. That means wavelet coefficients are correlated in a small neighbourhood. A large wavelet coefficient will probably have large coefficients at its neighbours. Therefore, Cai et al. [3] proposed the following wavelet denoising scheme for 1D signal by incorporating neighbouring coefficients in the thresholding process. Suppose $d_{j,k}$ is the set of wavelet coefficients of the noisy 1D signal. If $S_{j,k}^2 = d_{j,k-1}^2 + d_{j,k}^2 + d_{j,k+1}^2$ is less than or equal to λ^2, then we set the wavelet coefficient $d_{j,k}$ to zero. Otherwise, we shrink it according to

$$d_{j,k} = d_{j,k}(1 - \lambda^2 / S_{j,k}^2)$$

where $\lambda = \sqrt{2\sigma^2 \log n}$ and n is the length of the signal. Note that we should omit the first (last) term in $S_{j,k}^2$ if $d_{j,k}$ is at the left (right) boundary of level j wavelet coefficients.

For image denoising, we have to do a 2D wavelet transform. At every decomposition level, we get four frequency subbands, namely, LL, LH, HL, and HH. The next level should be applied to the low frequency subband LL only. This process is continued until a prespecified level is reached.

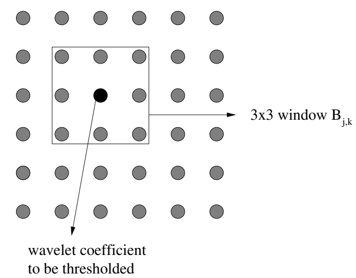

Fig. 1. An illustration of the neighbourhood window centered at the wavelet coefficient to be thresholded.

Since the Gaussian noise will be nearly averaged out in the low frequency wavelet coefficients and we want to keep small coefficients in these frequencies, only wavelet coefficients in the high frequency levels need to be thresholded. That means we need to threshold all LH, HL, and HH within these high frequency subbands. For every wavelet coefficient $d_{j,k}$ of our interest, we need to consider a neighbourhood window $B_{j,k}$ around it. We choose the window by having the same number of pixels above, below, on the left or right of the pixel to be thresholded. That means the neighbourhood window size should be 3×3, 5×5, 7×7, 9×9, etc. Figure 1 illustrates a 3×3 neighbourhood window centered at the wavelet coefficient to be thresholded. We threshold different wavelet coefficient subbands independently.

Let

$$S_{j,k}^2 = \sum_{(i,l)\in B_{j,k}} d_{i,l}^2$$

when the above summation has pixel indices out of the wavelet subband range, we omit the corresponding terms in the summation. For the wavelet coefficient to be thresholded, we shrinkage it according to the following formula:

$$d_{j,k} = d_{j,k}\beta_{j,k}$$

where the shrinkage factor can be defined as:

$$\beta_{j,k} = (1 - \lambda^2 / S_{j,k}^2)_+$$

here, the $+$ sign at end of the formula means to keep the positive value while set it to zero when it is negative, and $\lambda = \sqrt{2\sigma^2 \log n^2}$. Note that this thresholding formula is a modification to the classical soft thresholding scheme developed by Donoho and his coworkers. The neighbourhood window size around the wavelet coefficient to be thresholded has influence on the denoising ability of our proposed algorithm. The larger the window, the relatively smaller the threshold is. If the size of the window around the pixel is too large, a lot of noise will be kept, so an intermediate window size of 3×3 or 5×5 should be used.

The neighbour wavelet image denoising algorithm can be described as follows:

1. Perform forward 2D wavelet decomposition on the noisy image.
2. Apply the proposed shrinkage scheme to threshold the wavelet coefficients using a neighbourhood window $B_{j,k}$ and the universal threshold $\sqrt{2\sigma^2 \log n^2}$.
3. Perform inverse 2D wavelet transform on the thresholded wavelet coefficients.

We call this algorithm $NeighShrink$. As we know *VisuShrink* kills too many small wavelet coefficients, our shrinkage schemes should perform better. From our experiments we find that *NeighShrink* performs the best. It outperforms $VisuShrink$, *SUREShrink*, and *Wiener* filter for all experiments.

We also consider the shrinkage scheme that calculates the average value in the neighbourhood window and then threshold the current wavelet coefficient according to this average value. This thesholding formula can be given as:

$$\beta_{j,k} = (1 - N^2\lambda^2/S_{j,k}^2)_+$$

where N is the neighbourhood window size in one dimension. However, using the average gets worse denoising results than $VisuShrink$. We give this thresholding formula because taking the average is a natural choice. Unfortunately, it does not give us better denoising results.

This algorithm has higher computational demands. We give the algorithm complexity of the algorithm here. The forward 2D wavelet transform needs $2Ln^2$ flops of computation, where L is the wavelet filter length and n is the image size in one dimension. The thresholding process using neighbour information requires N^2n^2 flops of calculation, where N is the neighbourhood window size in one dimension. The inverse 2D wavelet transform also needs $2Ln^2$ flops of computation, just like the forward 2D wavelet transform. In total, the algorithm *NeighShrink* takes $(4L + N^2)n^2$ flops of computation. On the other hand, $VisuShrink$ only needs $4Ln^2$ flops of computation. We get better quality denoised images by sacrificing some amount of computation time.

3. EXPERIMENTAL RESULTS

We perform our experiments on the well-known images *Lena*. We get this image from the free software package $WaveLab$ developed by Donoho et al. at Stanford University. For comparison, we implement *VisuShrink*, *NeighShrink*, *SUREShrink*, and *Wiener* filter. *VisuShrink* is the universal soft-thresholding denoising technique [9] and *SUREShrink* is a SURE risk-based scale dependent denoising technique. Our program is written in Matlab by calling WaveLab functions. We use a 5×5 neighbourhood of each pixel in the image for the *Wiener* filter. The Daubechies wavelet with 8 vanishing moments is used for the wavelet decomposition. All detailed scales except the five coarsest scales are thresholded using the universal threshold $\sqrt{2\sigma^2 logn^2}$. Note that this threshold is the same as Donoho's threshold for 1D signal except we replace n in 1D signal with n^2 in 2D image. For different Gaussian white noise levels, the experimental results in Peak Signal to Noise Ratio ($PSNR$) are shown in Table 1 for denoising images *Lena*. The $PSNR$ is defined as

$$PSNR = -10\log_{10}\frac{\sum_{i,j}(B(i,j) - A(i,j))^2}{n^2 \max_{i,j} A(i,j)^2}.$$

where B is the denoised image and A is the noise-free image. The first column in this table is the $PSNR$ of the original noisy images, while other columns are the PSNR of the denoised images by using different denoising methods. From Table 1 we can see that *NeighShrink* outperforms *VisuShrink*, *SUREShrink*, and *Wiener* filter for all cases. *VisuShrink* does not have any denoising power when the noise level is low. Under such a condition, *VisuShrink* produces even worse results than the original noisy images. However, *NeighShrink* performs very well in this case. When the noise level is low, the improvement of $NeighShrink$ over $VisuShrink$ is large. When the noise level is high, the improvement is low even though $NeighShrink$ is still better than $VisuShrink$. Figure 2 shows the noise-free images, the same image with noise added, the denoised image with *VisuShrink*, the denoised image with *NeighShrink*, the denoised image with *SUREShrink*, and the denoised image with *Wiener* filter for images *Lena*. By studying the denoised images in Figure 2, we see that *NeighShrink* produces smoother and clearer denoised images. We also threshold the wavelet coefficients by looking at the average value in the neighbourhood window, and we find that it does not perform as well as *VisuShrink* for all denoising experiments. We conduct this experiment in this paper because taking the average of the wavelet coefficients in the neighbourhood window is a natural choice. Unfortunately, it does not provide better performance.

In order to investigate the influence of neighbourhood window size to the denoising ability, we list the experimental results for different window sizes in Table 2. These experiments are done using *NeighShrink*. We can see that the window sizes of 3×3 and 5×5 are the best. When the window size is getting larger, the denoising ability is getting worse. However, when the window size is extremely small, just like the term-by-term thresholding, the denoising ability is not very high. We have found that the intermediate neighbourhood window sizes of 3×3 and 5×5 are good choices for our proposed algorithm $NeighShrink$.

Noisy Image	*VisuShrink*	*NeighShrink*	*Wiener*
27.28	25.49	**31.50**	30.75
21.26	23.03	**27.43**	26.38
17.74	22.01	**25.24**	23.89
15.23	21.46	**23.83**	22.07
13.30	21.04	**22.85**	20.58
11.72	20.68	**22.10**	19.29
10.38	20.37	**21.50**	18.14

Table 1. The PSNR (dB) of the noisy images of Lena and the denoised images with different denoising methods.

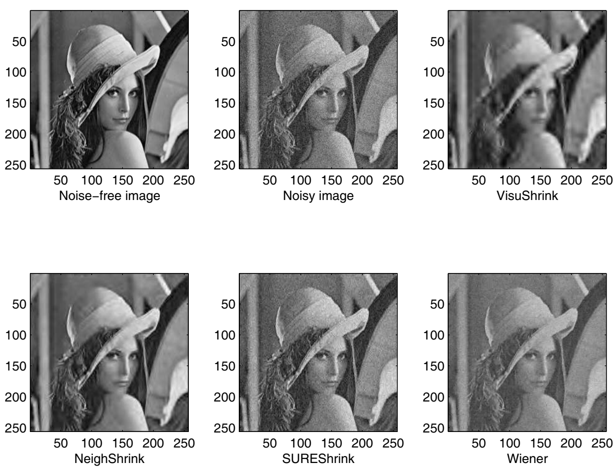

Fig. 2. Image denoising by using different methods on a noisy image with PSNR = 21dB.

	Noisy	*Window Size*			
	Image	1x1	3x3	5x5	7x7
Lena	17.74	22.01	**25.25**	25.05	21.74
MRIScan	18.56	22.70	**26.76**	26.32	22.71
Fingerprint	16.51	20.83	23.68	**24.57**	21.17
Phone	18.56	23.04	**24.90**	24.25	21.78
Daubechies	17.28	28.43	**29.59**	27.27	22.09
Coifman	15.34	24.87	**26.77**	24.91	20.05

Table 2. The PSNR (dB) of the denoised images with different neighbourhood window sizes.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we study image denoising by incorporating neighbouring wavelet coefficients. Experimental results show that *NeighShrink* gives better results than *VisuShrink*, *SUREShrink* and *Wiener* filter under all experiments. It should be mentioned that in this paper we investigate only how the classical soft thresholding approach should be modified to take into account neighbour wavelet coefficients. We conclude that *NeighShrink* can be used for practical image denoising applications. Future work may be done by considering the technique of incorporating neighbour multiwavelet coefficients in the thresholding process for multiwavelet image denoising.

5. REFERENCES

[1] R. R. Coifman and D. L. Donoho, "Translation Invariant Denoising," In *Wavelets and Statistics, Springer Lecture Notes in Statistics 103*, pp. 125-150, New York:Springer-Verlag.

[2] T. D. Bui and G. Y. Chen,"Translation invariant denoising using multiwavelets," *IEEE Transactions on Signal Processing*, vol.46, no.12, pp.3414-3420, 1998.

[3] T. T. Cai and B. W. Silverman, "Incorporating information on neighbouring coefficients into wavelet estimation," *Sankhya: The Indian Journal of Statistics*, Vol. 63, Series B, Pt. 2, pp. 127-148, 2001.

[4] G. Y. Chen and T. D. Bui, "Multiwavelet Denoising using Neighbouring Coefficients," *IEEE Signal Processing Letters*, vol.10, no.7, pp.211-214, 2003.

[5] W. Shengqian, Z. Yuanhua and Z. Daowen, "Adaptive shrinkage denoising using neighbourhood characteristic," *Electronics Letters*, vol. 38, no. 11, pp. 502-503, 2002.

[6] L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2744-2756, 2002.

[7] L. Sendur and I. W. Selesnick, "Bivariate Shrinkage with Local Variance Estimation," *IEEE Signal Processing Letters*, Vol. 9, No. 12, pp. 438-441, 2002.

[8] M. S. Crouse, R. D Nowak and R. G. Baraniuk, "Wavelet-based signal processing using hidden markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886-902, 1998.

[9] D. L. Donoho, "Denoising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 613-627, 1995.