A Spatially Selective Filter Based on the Undecimated Wavelet Transform That Is Robust to Noise Estimation Error

Dayong Zhou, Victor DeBrunner and Joseph Havlicek
School of Electrical and Computer Engineering
The University of Oklahoma
202 West Boyd Street, Room 219
Norman, OK 73019
{dayong,vdebrunn,joebob}@ou.edu

Abstract

Xu et al, *[1] proposed an effective wavelet based spatial selective denoising algorithm. The performance of the algorithm depends on the noise power estimation. Pan* et al, *[2] tried to improve the performance via a small modification. However, our simulation shows that both of these methods are sensitive to the noise estimation. Here we analyze the sensitivity of these two methods and introduce a new spatially selective noise filter based on the UDWT (Undecimated wavelet transform) that uses spatial correlation thresholding. Theoretic analysis and simulations show our algorithm improves the denoising effect. They also show that our proposed method is robust to errors in the noise power estimate. Because our approach is robust, we can relax the requirements for the estimation of the threshold without sacrificing performance, and so our method is more computationally efficient. We also put some perspective on the impact of employing non-orthogonal representations. Simulation results show the effectiveness of our proposed algorithm.*

1. Introduction

Unlike the Fourier transform, the wavelet transform gives a multi-resolution analysis of a signal. It is used widely in signal processing applications such as denoising and coding. The wavelet shrinkage (denoising) method introduced by Donoho and Johnstone [3] is a popular method for image denoising. In this approach, large transform coefficients are assumed to be associated with the signal while small transform coefficients are assumed to be associated with the noise. However, this approach exhibits spurious oscillations and other visual artifacts. Lang, et al., proposed a denoising algorithm using an undecimated wavelet transform (UDWT) [4]. The shift invariance of the UDWT appears to improve the denoising performance in both vision and mean square error sense [4]. Many researchers have pointed out that the key factor in the performance of the wavelet shrinkage methods is the thresholding, which is related to the noise power estimation. Most methods do not incorporate the correlation information existing between scales. With this in mind, Xu, et al., proposed an effective spatially-selective nonlinear filter [1]. Pan, et al., [2] modified the Xu algorithm in its method of choosing *the stopping point* of the nonlinear filtering (iteration), yielding a slightly improved performance.

Our simulations and analysis show that both the Xu method and the improvement offered by Pan are sensitive to the noise power estimate used in defining the nonlinear denoising filter. In this paper, we introduce a new spatially selective noise filter based on the UDWT that incorporates ideas from these three papers [1][2][4]. In Section 2, we analyze the Xu and Pan spatial selective methods and point out the cause of the sensitivity of these two methods. Then the undecimated wavelet transform and its properties are discussed in Section 3. The new spatial selective nonlinear filter based on UDWT is introduced in Section 4. Simulation results are given in Section 5, while conclusions and discussion appear in Section 6.

2. Previous work – some details

The Xu [1] algorithm is a spatially-selective noise filtration technique that uses the nonorthogonal wavelet introduced by Mallat and Zhong [5]. The method extracts the signal at each scale through the direct correlation of the coefficients at several adjacent scales. We have the inter-scale correlation

$$C_L(m,n) = \prod_{i=0}^{L-1} W(m+i,n), \quad n = 1,2,\ldots,N, \qquad (1)$$

0-7803-8387-7/04/$20.00 © 2004 IEEE

where $W(m,n)$ are the wavelet transform detail coefficients of the signal x at level m and point n. The first step in the algorithm is to normalize $C_2(m,n)$

$$\hat{C}_2(m,n) = C_2(m,n)\sqrt{P_W(m)/P_C(m)}, \quad (2)$$

where

$$P_W(m) = \sum_n W^2(m,n)$$
$$P_C(m) = \sum_n C_2^2(m,n) \quad (3)$$

The normalized inter-scale correlation $\hat{C}_2(m,n)$ is used as a dynamic intra-scale threshold function. Transform components larger than the threshold are called "signal" and saved. $\hat{C}_2(m,n)$ is then recomputed based on the remaining components. This renormalization of the inter-scale correlation lowers the threshold function, and more components are classified as signal. The iterated in-scale classification is stopped according to the estimated noise power. The devil in the details, of course, is determining the *stopping point* because accurate estimation of the noise power can be quite difficult. Stopping too early will fail to save (extract) significant signal components, while stopping too late allows too much noise to be saved. Simulation shows that this algorithm will generate more visually pleasing results when some signal is lost.

Pan observed in [2] that performance of the Xu algorithm can be improved by modifying the *stopping point* according to

$$P_W(m) - \mathrm{th}(m)(N-K)\sigma_m^2 > 0.05P_W(m), \quad (4)$$

which stops the iteration at a slightly earlier point. In (4), N is the total number of samples, K is the number of samples extracted as signal, and $1.2 \le \mathrm{th}(m) \le 1.8$ is a scale-dependent value. Rewriting, we have

$$N\frac{P_W(m)}{N-K} > \frac{\mathrm{th}(m)\sigma_m^2 N}{0.95} = (1.05\times\mathrm{th}(m))\sigma_m^2 N. \quad (5)$$

The left side is the noise power estimated from the remaining wavelet coefficients. The right side is the noise power estimated from noise variance, which is $P_W(m) > c \times$ estimation noise power at scale m, $c > 1$.

Simulations indicate that this alteration in threshold generates better results when the noise power estimate is reliable. Simulation results based on the two algorithms are given in Fig. 1 using a typical Lena image and added white Gaussian noise with MSE=600 out of 256 gray levels (about 10dB SNR). The horizontal axis gives the ratio of the estimated noise power to the true noise power; the vertical axis gives the percentage of noise extracted. We see that both algorithms can remove the noise effectively. However, the performance of both algorithms is very sensitive to noise power estimation errors.

Analysis shows this sensitivity is caused by the extraction process. If the noise power estimate is too low, then the algorithm will continue extracting noise after the signal information has been extracted. On the other hand, if the noise power estimate is too high, the algorithm will stop when significant signal information remains in the unextracted wavelet coefficients. This is most pronounced in the fine scales, which are important for keeping sharp edges and are also frequently dominated by noise.

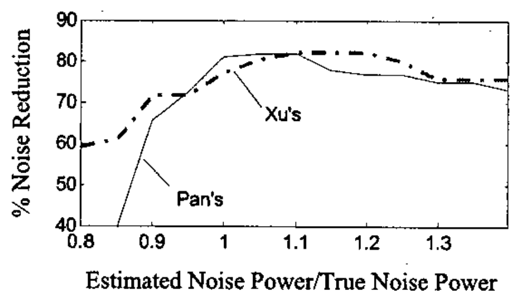


Fig. 1: Noise reduction vs. noise power estimation.

Since the Xu and Pan algorithms are sensitive to the noise power estimate, this paper considers the development and analysis of a robust, spatially-selective filter based on the UDWT.

3. Undecimated wavelet transform

The UDWT generates an equal number of coefficients at all resolution levels, and so is overcomplete. Its joint localization in the time-frequency plane allows us to denoise an image using the spatial information. We will give a brief introduction to UDWT next. In an orthogonal wavelet transform, there exist a scaling function $\phi(t)$ and a mother wavelet $\psi(t)$. The scaling function $\phi(t)$ can be built from a multiresolution analysis of $L^2(\mathbb{R})$. The set of functions $\{2^{m/2}\phi(2^{m/2}l-n)\}$ is an orthonormal basis of V_m. The set of functions $\{2^{m/2}\psi(2^{m/2}l-n)\}$ forms an orthonormal basis of W_m, where $V_{m+1} = V_m \oplus W_m = V_0 \oplus W_0 \oplus W_1 \oplus \cdots W_m$. Using these facts, we can decompose a signal $x(t) \in L^2(\mathbb{R})$ onto

$\{V_0, W_0, W_1 \cdots W_m\}$. An orthogonal wavelet decomposition of a continuous signal $x(t) \in L^2(\mathbb{R})$ with mother wavelet $\psi(t)$ results in

$$w_j^k(x) = \left\langle x(t), \frac{1}{2^{j/2}}\psi(\frac{t}{2^j} - k)\right\rangle$$
$$\triangleq \frac{1}{2^{j/2}}\int_{-\infty}^{+\infty} x(t)\psi^*(\frac{t}{2^j} - k)dt, \quad (k,j) \in \mathbb{Z}^2$$

An efficient algorithm for implementing the discrete orthogonal wavelet transform with ½-band filters was developed by Mallat [7]. In the UDWT, we do not down sample in the analysis (decomposition) side or up sample in the synthesis (construction) side of the filter bank, as we would in the discrete orthogonal wavelet transform case. Instead, we up and down sample the wavelet filter coefficients. The UDWT is illustrated in Fig. 2. In the figure, H and L are identically those used in the discrete orthogonal wavelet transform. $\uparrow H$ and $\uparrow L$ mean dyadic upsampling of H and L by 2. "$\uparrow\uparrow$" means two consecutive applications of dyadic upsampling, and so on [6].

The UDWT generates an equal number of coefficients at all resolution levels. As in any wavelet transform, $W(m,n)$ contain the detail information at scale m and translation n. It is this joint time frequency localization that we exploit to denoise the image [6].

In Fig. 2, x is Gaussian white noise distributed as $x \sim N(0, \sigma^2)$. The contents of the UDWT at each level are just x filtered by an FIR filter, so the noise variance at each scale can be directly computed as

$$\sigma_m^2 = \sigma^2 \left\| L*(\uparrow L/2)*(\uparrow\uparrow L/2)\cdots*(\uparrow\ldots\uparrow L/2)\right\|^2,$$

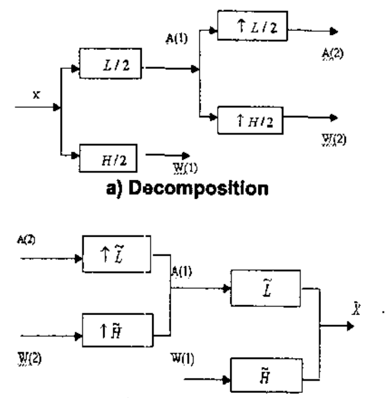
where '$*$' denotes linear convolution. The UDWT is shift invariant and redundant, and thus, in contrast to orthogonal wavelet transform denoising methods, may yield superior visual performance, e.g. no ringing. In [2] and [4], analysis and simulations show that using an appropriate threshold, the UDWT can generate a better denoising effect both visually and in the sense of mean square error. However, those methods do not incorporate the spatially correlated information between different scales exploited in the Xu algorithm.

4. A noise estimation error robust spatial selective filter

Now that we have seen what others have done, we examine our method for altering the threshold selections in the nonlinear filtering so that the resulting algorithm is robust to the estimated SNR.

4.1 Our proposed algorithm

We retain (1), in which $W(m,n)$ is the UDWT detail coefficients of a noisy signal x at scale m and sample n. Since we propose to use a different wavelet decomposition than Xu, we will distinguish between the two correlations. We use the notation $C_2'(m,n)$ to indicate our UDWT-based correlation.



b) Construction
Fig. 2: UDWT block diagrams.

In many respects, our proposed algorithm is similar to the Xu method, but we use a different spatial selection criterion to extract the signal components. We give a brief description of our choice:

1. Take the UDWT of the signal.
2. Calculate $C_2'(m,n)$ at each scale.
3. If $C_2'(m,n)$ is larger than the threshold value, select the component as a signal component.
4. Invert the UDWT of the signal components.

Our algorithm can also be regarded as a generalization of the Lang algorithm. Like the Lang algorithm, our method does not require iteration. In the algorithms developed by Xu and Pan, the threshold value is very critical to the performance of the algorithm. However, we are going to show in the following discussion that our algorithm is insensitive to the noise power estimate and therefore does not require prior or precise knowledge of the noise.

4.2 Noise power estimation insensitivity

$C_2'(m,n)$ is the product of UDWT coefficients at adjacent levels and may be modeled as a product of uncorrelated Gaussian processes ξ and γ. Thus ξ and γ are filtered sub-bands. Consequently, they are Gaussian, but not generally white. However, though the sub-bands are undecimated, each one is produced otherwise identically to one that would be orthogonal were the downsampling done; they are still approximately orthogonal to each other. For Gaussian noise, orthogonal implies uncorrelated, and so:

$$\mathrm{E}\{C_2'(m,n)\} = E\{\xi\}E\{\gamma\} = 0 \quad (6)$$

$$Var\{C_2'(m,n)\} = Var\{\xi\}Var\{\gamma\} \quad (7)$$

$$f_{\varsigma}\left(C_2'(m,n)\right) = \int_{-\infty}^{\infty} \frac{1}{|\xi|} f_{\xi\gamma}\left(\xi, \frac{C_2'(m,n)}{\xi}\right) d\xi \quad (8)$$

Calculating (8) can be difficult; however, Monte Carlo simulations indicate that the multiplication concentrates the distribution (see Fig. 3). As a result, if our threshold takes a relatively smaller value, most of the contents of $C_2'(m,n)$ arising from the noise can still be subtracted. On the other hand, edges are relatively unaffected because they appear at all scales. Here the joint localization of the UDWT helps us immensely. $C_2'(m,n)$ is the product of two scales. Through $C_2'(m,n)$, the noise information is suppressed while the signal information is amplified. Thus, we have that the difference of the parts of $C_2'(m,n)$ due to the noise and those parts due to the signal are much larger than the difference of the correlation to the wavelet coefficients themselves. Our algorithm is thus robust to errors in the threshold value caused by misestimating the noise power.

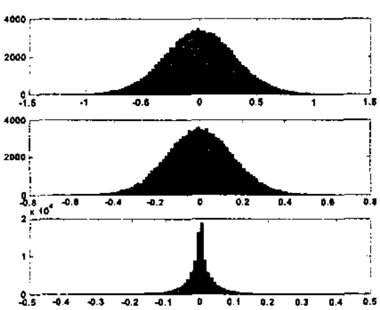


Fig. 3: Sample distributions (level 5 top, level 6 middle, and product of the two at bottom).

4.3 Threshold selection

There are several ways to calculate the threshold. Here we provide two methods. In the first method we use the fact that the UDWT retains all spatial relationships. Consequently, to characterize the noise, we can estimate $C_2'(m,n)$ based on a region of the image where there is only noise present, yielding

$$threshold = c * \max\left(C_2'(m,n)\right), 0.5 \le c \le 0.8 \quad (9)$$

The second approach for calculating the threshold is statistical – assume that the noise is AWGN, with distribution $\xi \sim N(0,\sigma^2)$. Then, we can use (6)-(8) to estimate the properties of $C_2'(m,n)$, yielding

$$threshold = c * \sigma_{\xi\gamma}^2, 2 \le c \le 3 \quad (10)$$

One significant difference from the Lang method [4] is that both of our thresholds (9) and (10) are not constant in the wavelet transform domain (though they are in the correlation domain).

5. Simulation

5.1 Sensitivity comparison

We simulate our algorithm and compare it to the other algorithms. We measure the reduced noise power with respect to the different noise power estimations. The simulation result is shown in Fig. 4. It is evident that our algorithm is very robust to the noise power estimation error. The "threshold" method shown is essentially the Lang method, or alternatively, the second variant given by Pan [2].

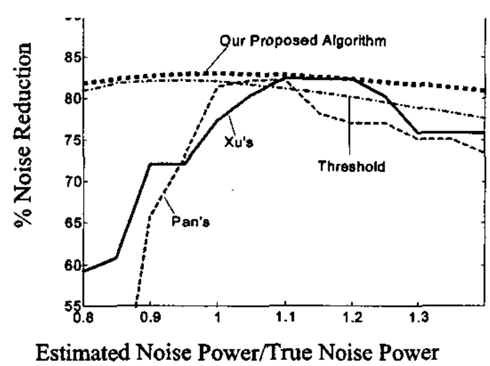


Fig. 4: Sensitivity analysis for Lena Image, σ^2 =600.

5.2 Visual comparisons

The above sensitivity simulation compares MSE performances that may not be indicative of visual quality. Accordingly, we compare the methods visually. Figure 5 shows the noisy image, the best result generated by Xu's algorithm (when the noise estimate is 1.2 times the true noise power), and our algorithm. In this case, the methods appear to be visually comparable. However, our result is significantly better than the typical results delivered by the Xu algorithm, as is expected by examining Fig. 4.

6. Discussion

The Xu and Pan spatial-selective denoising algorithms are sensitive to the noise power estimate. In this paper, we introduced a spatially-selective filter based on the UDWT that is robust to the noise power estimation error. Our algorithm also displays better visual effects while improving the MSE above the Lang algorithm, as shown in Fig. 4. The improvements are achieved by using the shift-invariant nature of the UDWT and the spatial correlation between scales. As a result, we obtain MSE and visual performance that is equivalent to the best performances attainable by those previous algorithms and do so over a much wider range of power estimation errors.

We note that Bao, *et al*, [8] recently proposed a novel algorithm based on spatial correlation thresholding simultaneously with our independently developed algorithm [9]. At first glance, their algorithm looks similar to ours. However there are several distinctions in the two algorithms. First, the Bao algorithm is based on a dyadic transform, while ours is based on the more general UDWT. More importantly, their algorithm requires a numerically complex SNR computation to set the threshold – our method does not require this added computational burden. In fact, the point of our algorithm was to develop one that is robust to these errors! Finally, Bao [8] provides no comparisons of their algorithm with either that of Xu or Pan. In this paper, we begin this comparison. In the future, we intend to extend our comparisons. Also, we note that further improvements are possible by using a self-orthogonalizing matching pursuits algorithm. We expect to do this next, as well.

7. References

[1] Y. Xu, J. B. Weaver, D. Healy, and J. Lu, "Wavelet transform domain filters: A spatially selective noise filtration technique," *IEEE T-IP*, vol. 3, no. 6, Nov 1994.

[2] Q. Pan, L. Zhang, and G. Dai "Two denoising methods by wavelet transform," IEEE T-SP, vol. 47, no. 12, Dec 1999.

[3] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, **81**, pp. 425-455, 1994.

[4] M. Lang, H. Guo, J. E. Odegaard, C. S. Burrus, and R. O. Wells, "Noise reduction using and undecimated discrete wavelet transform," *IEEE SP Lett.*, 3, pp. 10-12, 1996. S.

[5] Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE T-PAMI*, vol. 14, no. 7, Jul 1992.

[6] M. Jansen, "Noise reduction by wavelet thresholding," *Springer*, 2001.

[7] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE T-PAMI*, vol. 11, no. 7, pp. 674-693, 1989.

[8] P. Bao and L. Zhang, "Noise reduction for magnetic resonance images via adaptive multiscale products thresholding", *Medical Imaging, IEEE Transaction On*, Vol. 22, no. 9, pp. 1089-1099, Sep. 2003

[9] D. Zhou, "Wavelet based image denoising", *term project for ECE 5273*, OU, May 2002.

a) Picture before denoising

b) Denoised picture by our algorithm

c) Best denoised picture by Xu's algorithm

Fig. 5: Visual comparisons.