# Two Denoising Methods by Wavelet Transform

Quan Pan, Lei Zhang, Guanzhong Dai, and Hongcai Zhang

*Abstract*— Two wavelet-based noise reduction methods are discussed here. First, we improve the tradtional spatially selective noise filtration technique proposed by Xu *et al*. Second, we introduce a new threshold-based denoising algorithm based on undecimated discrete wavelet transform. Simulations and comparisons are given.

*Index Terms*— Denoising, spatial correlation, threshold, undecimated discrete wavelet transform.

## I. INTRODUCTION

Wavelet transforms can decompose a signal into several scales that represent different frequency bands, and at each scale, the position of signal's instantaneous structures can be determined approximately. Such a property can be used for denoising.

In [2], a spatially selective noise filtration technique was proposed. Based on the direct spatial correlation of wavelet transform at several adjacent scales, a high correlation is used to judge if there is a significant edge. The choice of noise power reference is very important in implementation, and it was not shown in [2]. In this correspondence, we give the noise power reference and an estimation of the standard deviation of original noise. In addition, we introduce some parameters and extract edges from coarse scales to fine scales to improve the filtering performance.

Another powerful approach to noise reduction was proposed by Donoho *et al*. [3]–[5]. In the case of orthogonal wavelet transform (OWT), Donoho made use of a threshold $t = \sigma\sqrt{2\log N}$ for all scales to obtain an ideal risk, but partly due to the lack of translation invariance of OWT, the results exhibit visual artifacts [9]. In this correspondence, a new threshold $t(m) = c\sigma_m$ in the case of undecimated discrete wavelet transform (UDWT) is presented, and why UDWT can suppress noise better than OWT is briefly illustrated.

## II. DYALTIC WAVELET TRANSFORM AND UNDECIMATED WAVELET TRANSFORM

The continuous wavelet transform can be defined as

$$(W_\psi f)(b, a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt, \quad f \in L^2(IR) \quad (1)$$

where $a, b \in L^2(IR)$, and $a \neq 0$. To allow fast numerical implementation, we impose that the scale parameter $a$ varies only along the dyadic sequence $(2^j)$, $j \in Z$. A wavelet $\psi \in L^2(IR)$ is a dyadic wavelet if and only if there exist two strictly positive constants $A$ and $B$ so that

$$\forall \omega \in IR, \quad A \leq \sum_{j=-\infty}^{\infty} |\hat{\psi}(2^j\omega)|^2 \leq B. \quad (2)$$

Equation (2) ensures that $f(t)$ can be recovered from its dyadic wavelet transform. The reconstructed wavelet $\chi(t)$ may be any
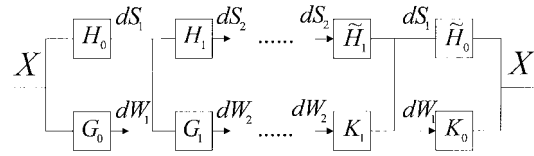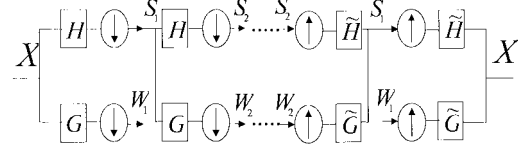
Fig. 1. Dyadic wavelet transform.



Fig. 2. Orthogonal wavelet transform.
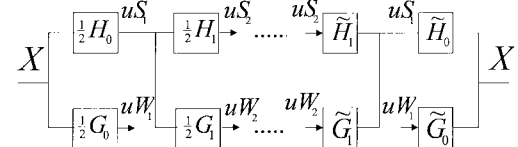


Fig. 3. Undecimated wavelet transform.

function that satisfies

$$\sum_{j=-\infty}^{\infty} \hat{\psi}(2^j\omega)\hat{\chi}(2^j\omega) = 1. \quad (3)$$

The dyadic wavelet transform is redundant. A fast discrete algorithm [1] is shown in Fig. 1. The filter $F_j$ is obtained by putting $(2^j - 1)$ zeros between each of the coefficients of the filter $F_0$; therefore, the bandwidth of $F_j$ is $1/2^j$ of the bandwidth of $F_0$.

If we also want to impose that the translation parameter $b$ varies along dyadic sequence $(2^j)$, then more constraints must be imposed on constructing the wavelet. See Daubechies [6]–[8] for more details about (bi-)orthogonal compactly supported wavelets. Mallat has given the fast pyramid algorithm of (bi-)orthogonal wavelet transform (OWT) (Fig. 2). Unfortunately, OWT is translation variant due to subsampling. If we rotate the input signal by one position, then the output signal at the first scale would be different. If the rotation were by two positions, then the output at the first scale would be the same except by one rotation, but the outputs at the second and higher scales would be different. Several methods can be used to overcome the dependence on the position of input signal. Here, we use the undecimated discrete wavelet transform (UDWT), which was shown in Fig. 3. It is noticeable that UDWT is almost the same as dyadic wavelet transform because the (bi-)orthogonal wavelet can also be called dyadic wavelet.

## III. THE STANDARD DEVIATION $\sigma_m$ OF GAUSSIAN WHITE NOISE AT EACH SCALE

From Fig. 1, suppose $X$ is Gaussian white noise $x \sim N(0, \sigma^2)$. It is easy to get the deviation of $W_1 = X * G_0$

$$\sigma_1^2 = D(W_1) = D(X * G_0) = \|g_n^0\|^2 \cdot D(X) = \sigma^2 \|g_n^0\|^2.$$

From the pyramid decomposing structure illustrated in Fig. 1, we can prove that

$$\sigma_m^2 = \sigma^2 \|h_n^0 * h_n^1 * \cdots * h_n^{m-2} * g_n^{m-1}\|^2 \quad (4)$$

where $\| \bullet \|$ denotes the norm of $S(n) \in l^2(n)$, and $*$ means convolution.

## IV. Spatially Selective Noise Filtration (SSNF)

Xu *et al.* developed a SSNF technique in [2]. They used the dyadic wavelet constructed by Mallat [1]. Based on the fact that sharp edges have large signals over many wavelet scales and noise will die out swiftly with scale, spatial correlation $Corr_l(m, n)$ is defined to sharpen and enhance edges and significant features while suppressing noise and small sharp features

$$Corr_l(m, n) = \prod_{i=0}^{l-1} W(m + i, n) \quad n = 1, 2, \ldots N \quad (5)$$

where $W(m, n)$ denotes the wavelet transform data, $m$ is the scale index, $n$ is the translation index, $l < M - m + 1$, and $M$ is the total number of scales. Usually, we select $l = 2$.

The algorithm [2] is described briefly as follows. The filtered data is referred to as $W_{\text{new}}(m, n)$:

1) Compute the correlation $Corr_2(m, n)$ for every wavelet scale $m$.

2) Rescale the power of $\{Corr_2(m, n)\}$ to that of $\{W(m, n)\}$ and get $\{New\,Corr_2(m, n)\}$.

$$New\,Corr_2(m, n) = Corr_2(m, n)\sqrt{PW(m)/P\,Corr(m)}$$

where

$$P\,Corr(m) = \sum_n Corr_2(m, n)^2$$

$$PW(m) = \sum_n W(m, n)^2.$$

3) If $|New\,Corr_2(m, n)| \geq |W(m, n)|$, we accept the point as an edge. Pass $W(m, n)$ to $W_{\text{new}}(m, n)$, and reset $W(m, n)$ and $Corr_2(m, n)$ to 0. Otherwise, we assume $W(m, n)$ is produced by noise and then retain $W(m, n)$ and $Corr_2(m, n)$.

4) Repeat 2) and 3) until the power of $W(m, n)$ is nearly equal to some reference noise power at the $m$th wavelet scale.

Finally, we get the vector $W_{\text{new}}(m, n)$ and then reconstruct the signal. In [2], the reference noise power is not shown. Here, we give a reference by supposing that original noise is white Gaussian.

It is well known that $\sqrt{PX/N}$ is an asymptotically unbiased estimation of $\sigma$ for a sequence $X \sim N(0, \sigma^2)$, where $PX = \sum_n x(n)^2$, and $N$ is the length of $X$. Suppose $K$ points have been extracted and that $W'(m, n)$ denotes the unextracted points in $W(m, n)$. If $W'(m, n)$ can be viewed as produced exactly by noise, $PW'(m)/(N - K)$ will be an asymptotically unbiased estimation of $\sigma_m^2$. $(N - K)\sigma_m^2$ can be taken as the reference noise power.

In fact, $W'(m, n) = W'_s(m, n) + W'_n(m, n)$, where $W'_s(m, n)$ and $W'_n(m, n)$ are the wavelet transform of true signals and, respectively, noise. Then, we have

$$PW'(m) = (N - K) \cdot E\{W'(m, n)^2\}$$
$$= (N - K) \cdot E\{W'_s(m, n)^2 + W'_n(m, n)^2$$
$$+ 2W'_s(m, n)W'_n(m, n)\}$$
$$= (N - K) \cdot E\{W'_s(m, n)^2\} + (N - K) \cdot \sigma_m^2.$$

$E\{W'_s(m, n)^2\}$ is always positive. $(N - K)\sigma_m^2$ should not be the reference noise power at coarse scales. We multiply $(N - K)\sigma_m^2$ by a factor $th(m)$ to be a new reference, where $th(m) > 1$. For different signals, $E\{W'_s(m, n)^2\}$ will be different; therefore, $th(m)$ should vary. Fortunately, the filtering results are not sensitive to $th(m)$, and we can choose a common $th(m)$ to the general case. $E\{W'_s(m, n)^2\}$ will increase with scale, whereas $\sigma_m^2$ decreases with scale. At fine scales, $\sigma_m^2$ is dominating in $PW'(m)$, but at coarse scales, $E\{W'_s(m, n)^2\}$ will be dominating. Based on our experience,

### TABLE I
New Entire Filtering Process in the Wavelet Transform Domain Described by a "Control Flow" Type of Scheme

First, Save a copy of $W(m,n)$ to $WW(m,n)$.
Initialize the "spatial filter mask": *mask(1:M,n)* to 0's, *mask(M+1,n)* to 1's
Choose $\lambda(m)$ and *th(m)* in advance.

Loop for each wavelet scale $m$ from $M$ to 1
{
  *K=0*
  Loop for the iteration process
    {
      Compute the power of $Corr_2(m,n)$ and $W(m,n)$:
$$PCorr(m) = \sum_n Corr_2(m,n)^2, PW(m) = \sum_n W(m,n)^2$$
      Rescale the power of $Corr_2(m,n)$ to that of $W(m,n)$.
      Loop for each pixel point $n$
      {
$$NewCorr_2(m,n) = Corr_2(m,n)\sqrt{PW(m)/PCorr(m)}$$
      }end loop $n$
      Loop for each pixel point $n$
      {
        Compare pixel values in $NewCorr_2(m,n)$ and $W(m,n)$:
        if $\left|NewCorr_2(m,n)\right| \geq \lambda(m)\left|W(m,n)\right|$
        {
          Extract edge information from $W(m,n)$ and $Corr_2(m,n)$,
          and save it in the "spatial filter mask":
          $Corr_2(m,n)$=0.0, $W(m,n)$ =0.0
          *mask(m, n)*=1, *K=K+1*
        }end if
      }end loop $n$
    }iterate until $\left(PW(m) - th(m) \cdot (N-K)\sigma_m^2\right) \leq 0.05 \cdot PW(m)$
    *mask(m, n)=mask(m, n).\* mask(m+1,n)*
    Apply the "spatial filter mask" to the save copy, $WW(m,n)$, at scale $m$.
    Save the filtered data to $W_{new}(m,n)$:
      Loop for each pixel point $n$
      {
$$W_{new}(m,n) = mask(m,n).\* WW(m,n)$$
      }end loop $n$
}end loop $m$

we can choose $th(1) = 1.1$–$1.2$, $th(2) = 1.2$–$1.4$, $th(3) = 1.4$–$1.6$ and $th(m) = 1.6$–$1.8$ when $m > 3$.

At fine scales, noise is dominating except some sharp edges. If we compare $|New\,Corr_2(m, n)|$ with $|W(m, n)|$ directly, then too much noise will be extracted as edges. To avoid this, we multiply $|W(m, n)|$ by a weight $\lambda(m) \geq 1$ and impose that only when $|New\,Corr_2(m, n)| \geq \lambda(m)|W(m, n)|$ can we extract $W(m, n)$ as edges. In the simulations, we take $\lambda(m) = [1.15, 1.06, 1, 1, \ldots, 1]$, and the results are satisfying.

The edges will appear at all scales; therefore, we can assume that if there is no edge to be extracted at coarser scales, then we will not extract edge at finer scales at the corresponding indexes. Thus, we will extract edges from coarser scales to finer scales only at the indexes that have been extracted as edges. This will avoid extracting a lot of noise as edges at fine scales.

The new algorithm is summarized in Table I, where $M$ denotes the total number of scales.

## V. Estimation of the Standard Deviation $\sigma$

At fine wavelet scales, noise is dominating; thus, $\sigma$ can be estimated from the first two scales.

If $|New\,Corr_2(1, n)| \geq \lambda(1)|W(1, n)|$, reset the corresponding data in $W(1, n)$ to 0. Refer to the remainder of $W(1, n)$ as $\tilde{W}(1, n)$.
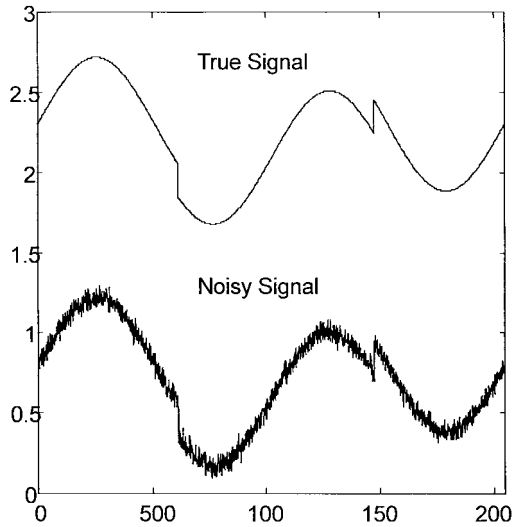
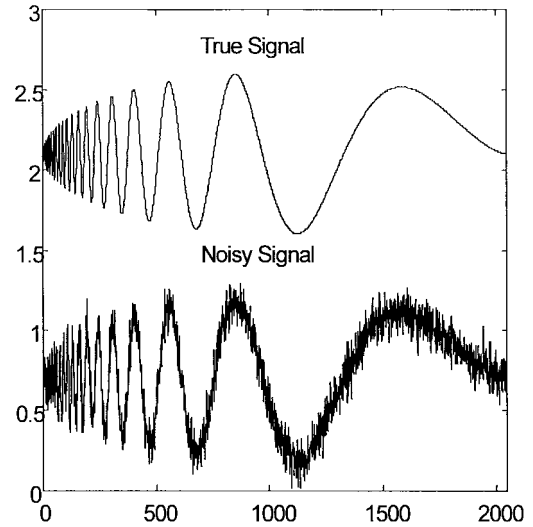Fig. 4.   True HeaviSine and the noisy one (SNR $= 20.17$ db).



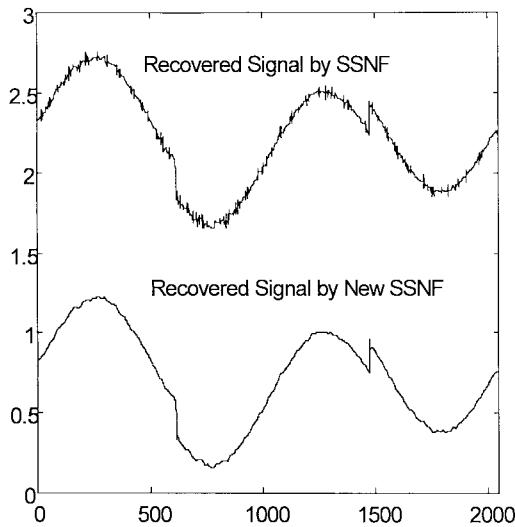Fig. 6.   True Doppler and the noisy one (SNR $= 12.46$ db).



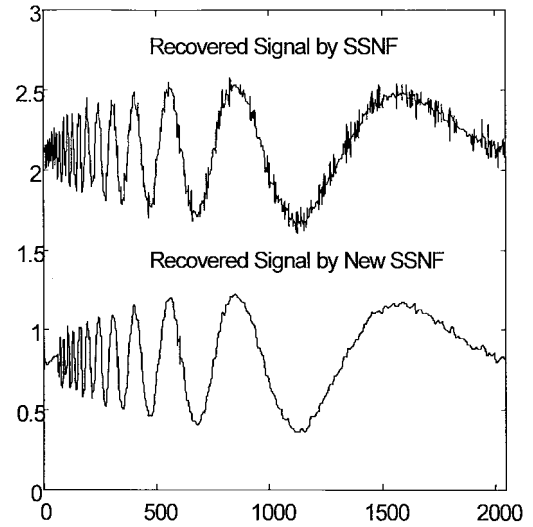Fig. 5.   Recovered HeaviSine by SSNF (SNR_$G = 7.08$ db) and by new SSNF (SNR_$G = 11.54$ db).



Fig. 7.   Recovered Doppler by SSNF (SNR_$G = 4.74$ db) and by new SSNF (SNR_$G = 7.68$ db).

Suppose $K$ points are killed totally; then, $\tilde{W}(1, n)$ can be roughly considered to be produced by noise. From $\sigma = \sigma_m / \|g_n^0\|$ and the asymptotically unbiased estimation of $\sigma_1^2$, i.e., $P\tilde{W}(1)/(N - K)$, we can get

$$\hat{\sigma} = \sqrt{P\tilde{W}(1)/(N - K)}/\|g_n^0\|. \tag{6}$$

## VI. THRESHOLD-BASED DENOISING BY UDWT

Threshold-based denoising was first proposed by Donoho [3]–[5]. It is very simple and of satisfying performance. It can be divided into three steps:

1) Transform the noisy signal $y$ into wavelet coefficient $w$.
2) Employ a hard or soft threshold $t$ at each scale $m$.
3) Transform back to the original domain, and get the estimated signal.

In case of orthogonal wavelet transform (OWT), Donoho gave the following soft threshold:

$$\eta_t(w) = \text{sgn}(w)(|w| - t)_+ \tag{7}$$

where $t = \sigma\sqrt{2 \log N}$, and $N$ is the length of signal.

Lack of a translation invariant will make denoising by OWT exhibit visual artifacts. In this correspondence, we describe UDWT and hard threshold [see (8)]. Although Donoho proved the optimality of soft threshold in theory, hard threshold has shown better results for certain applications [14].

$$\hat{w}(m, n) = \begin{cases} w(m, n), & w(m, n) \geq t(m) \\ 0, & w(m, n) < t(m) \end{cases} \tag{8}$$

We choose $t(m) = c \cdot \sigma_m$, where $c$ is a constant. It is well known that for i.i.d. Gaussian noise $X \sim N(0, \sigma^2)$, a threshold $t = \sigma, 2\sigma, 3\sigma, \dots$ will suppress 68.26%, 95.44%, and 99.74% of its values. Therefore, by imposing $c$ between 3–4, we will have good results. Donoho's threshold is varying with $N$, and when $N$ is too large, the threshold may oversmooth the signal.

Why does UDWT do better than OWT on denoising? We will explain this briefly as follows. From Figs. 2 and 3, suppose that we only decompose white Gaussian noise $X$ at the first scale. $[Se_1, We_1]$ and $[So_1, Wo_1]$ denote the wavelet coefficients for keeping even and odd index, respectively. For the case of OWT, we apply the same
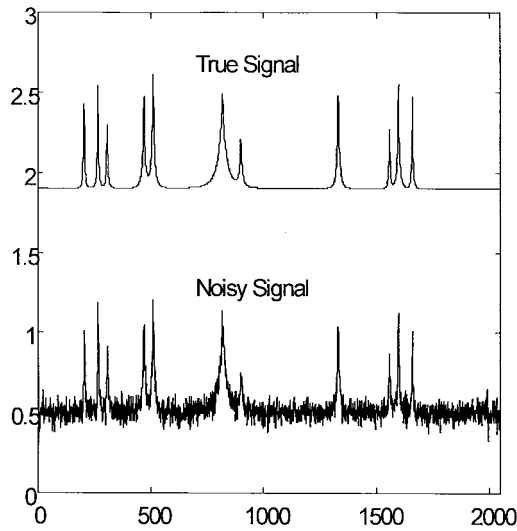
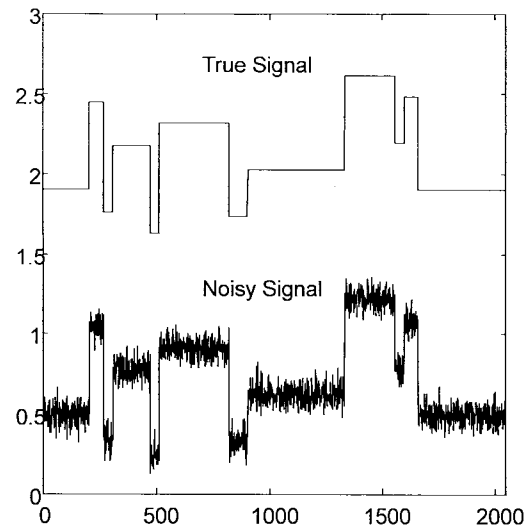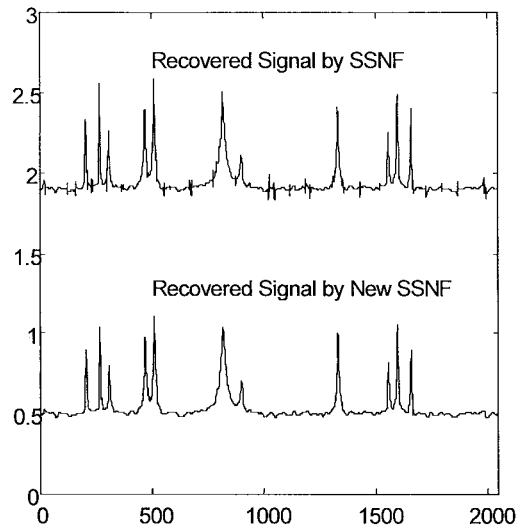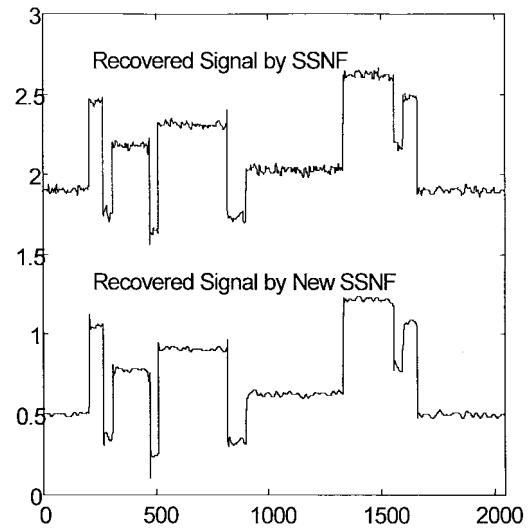Fig. 8.   True Bumps and the noisy one (SNR $= 7.72$ db).



Fig. 10.   True Blocks and the noisy one (SNR $= 15.89$ db).



Fig. 9.   Recovered Bumps by SSNF (SNR_$G = 7.45$ db) and by new SSNF (SNR_$G = 7.36$ db).



Fig. 11.   Recovered Blocks by SSNF (SNR_$G = 10.32$ db) and by new SSNF (SNR_$G = 10.68$ db).

threshold $t$ to $We_1$ and $Wo_1$, and after up-sampling, we have

$$Se_1(\omega) = \sum_{l=1}^{N/2} Se_1(2l)e^{-j2l\omega}, \quad We_1(\omega) = \sum_{l=1}^{N/2} We_1(2l)e^{-j2l\omega}$$

$$So_1(\omega) = \sum_{l=1}^{N/2} So_1(2l-1)e^{-j(2l-1)\omega}$$

$$Wo_1(\omega) = \sum_{l=1}^{N/2} Wo_1(2l-1)e^{-j(2l-1)\omega}.$$

$X_e$ and $X_o$ reconstructed by $[Se_1, We_1]$ and $[So_1, Wo_1]$, respectively. $X_e$ and $X_o$ should have the same deviation $\sigma_1$.

$$X_e(\omega) = Se_1(\omega)\tilde{H}(\omega) + We_1(\omega)\tilde{G}(\omega)$$
$$X_o(\omega) = So_1(\omega)\tilde{H}(\omega) + Wo_1(\omega)\tilde{G}(\omega).$$

In the case of UDWT, we apply $t/2$ to $uW_1$

$$uS_1(\omega) = \frac{1}{2}Se_1(\omega) + \frac{1}{2}So_1(\omega)$$

$$uW_1(\omega) = \frac{1}{2}We_1(\omega) + \frac{1}{2}Wo_1(\omega).$$

Therefore, the reconstructed signal $X_u$

$$X_u(\omega) = uS_1(\omega)\tilde{H}_0(\omega) + uW_1(\omega)\tilde{G}_0(\omega)$$
$$= \frac{1}{2}X_e(\omega) + \frac{1}{2}X_o(\omega)$$

i.e., $X_u = \frac{1}{2}X_e + \frac{1}{2}X_o$. The deviation of $X_u$ will be

$$\sigma_u^2 = \frac{1}{2}\sigma_1^2 + \frac{1}{2}E(X_e X_u) \le \frac{1}{2}\sigma_1^2 + \frac{1}{4}(D(X_e) + D(X_o)) = \sigma_1^2.$$

If we use SNR as the measure of filtering performance, we can see that UDWT will be better.

In practice, the noise is superimposed onto the signal. However, in fine scales, the wavelet coefficients are dominated by noise except some sharp edges, and the effect of signal can be ignored.

## VII. SIMULATION RESULTS

The simulations are made by using new SSNF method in this section. Fig. 4 shows the true HeaviSine and the noisy version. Fig. 5 shows the filtering results by the new method and the original one. Figs. 6–11 show the results for Doppler, Bumps, and Blocks,
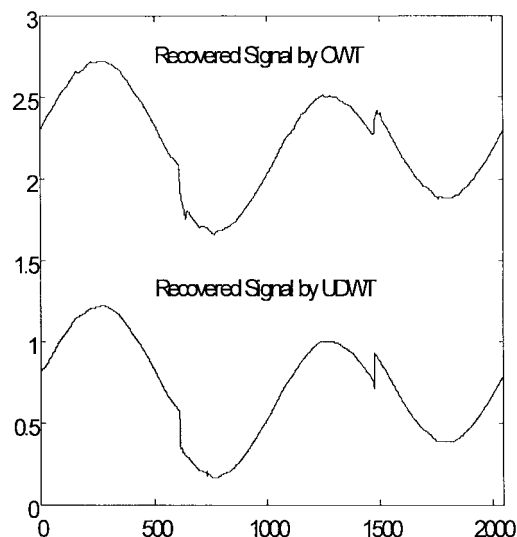
Fig. 12. Recovered HeaviSine by OWT (SNR_$G$ = 10.10 db) and by UDWT (SNR_$G$ = 13.95 db).
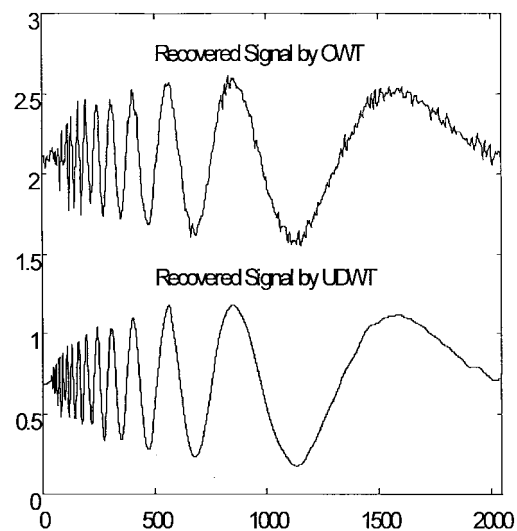


Fig. 14. Recovered Bumps by OWT (SNR_$G$ = 5.56 db) and by UDWT (SNR_$G$ = 10.88 db).



Fig. 13. Recovered Doppler by OWT (SNR_$G$ = 6.18 db) and by UDWT (SNR_$G$ = 10.65 db).
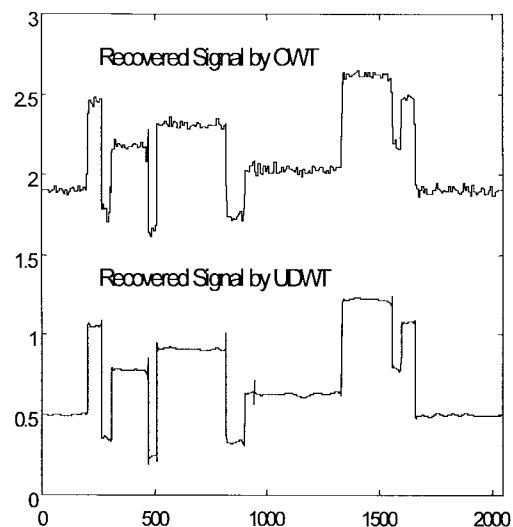


Fig. 15. Recovered Blocks by OWT (SNR_$G$ = 5.23 db) and by UDWT (SNR_$G$ = 14.85 db).

respectively. It can be seen that the new method is much better than the old one for HeaviSine and Doppler. For Bumps and Blocks, the two methods are almost the same.

Figs. 12–15 are the denoising results by Donoho's method (OWT) [3], [4] and the new threshold method of this correspondence (UDWT). It can be seen that the new method is always better than Donoho's.

Compared with the SSNF technique, threshold-based methods perform better and need less computation. However, the SSNF technique can analyze edges well and can be easily extended to edge detection, image enhancement, and other applications.

## VIII. CONCLUSION

We improve the SSNF technique and present a new threshold-based denoising method by using UDWT. In the new SSNF method, we propose a noise power reference and an estimation of the standard deviation of original noise. In addition, we introduce parameters $\lambda(m)$ and $th(m)$ and extract edges from coarse scales to fine scales to improve the f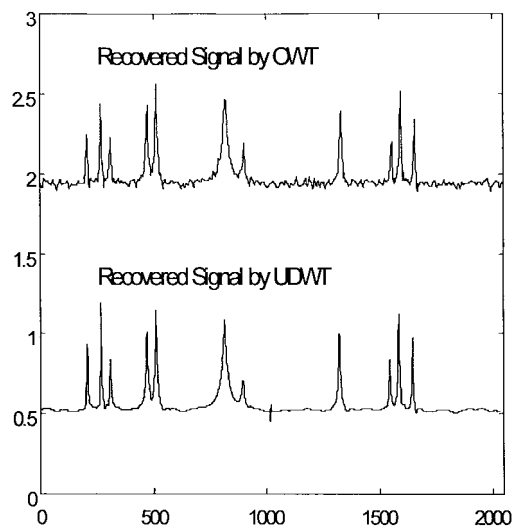iltering performance. Simulations show that the improved SNNF method performs much better than the old one when signals are of good smoothness. This correspondence presents a new threshold-based method by using UDWT. We illustrate briefly that by thresholding with UDWT, the standard deviation of noise would be smaller. Simulation results also show that the new method performs better for typical signals. Comparing the SSNF technique with the threshold-based method, the latter performs more satisfactorily and needs less computation, whereas the former can analyze edges satisfactorily and can be extended to edge detection, image enhancement, and other applications.

### REFERENCES

[1] S. Mallat, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Machine Intell,*, vol. 14, July 1992.
[2] Y. Xu *et al.*, "Wavelet transform domain filters: A spatially selective noise filtration technique," *IEEE Trans. Image Processing*, vol. 3, Nov. 1994.
[3] D. Donoho, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, 1995.
[4] ——, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, 1995.

[5] ——, "Wavelet shrinkage: Asymptopia?," *J. R. Stat. Soc.*, Series B, vol. 57, pp. 301–369, 1995.
[6] I. Daubechies, "Orthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, 1988.
[7] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
[8] A. Cohen, I. Daubedhies, and J. C. Feaurean, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, 1990.
[9] R. Coifman and D. Donoho, *Translation Invariant De-noising*. New York: Springer, 1994, pp. 125–150.
[10] V. Strela, P. N. Heller, and G. Strang, "The application of multiwavelet filter banks to image processing," Tech. Rep., 1995.
[11] M. Vetterti and C. Herly, "Wavelet and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, Sept. 1992.
[12] Y. Meyer, *Wavelets: Algorithm and Application*. Philadelpha, PA: SIAM, 1993.
[13] G. Beylkin *et al.*, "Fast wavelet transforms and numerical algorithms," *Commun. Pure Appl. Math.*, vol. 44, pp. 141–183, 1991.
[14] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal based and the minimum description length criterion," *SPIE*, vol. 2242, pp. 224–235, 1994.

# Wavelet-Based Estimation of $1/f$-Type Signal Parameters: Confidence Intervals Using the Bootstrap

Angelo M. Sabatini

*Abstract*— We propose to construct confidence intervals of parameters of $1/f$-type signals using a nonparametric wavelet-based bootstrap method. Bootstrap-based confidence intervals of maximum likelihood parameter estimates are compared to the confidence intervals derived from the Cramér–Rao lower bound (CRLB). For moderately large data sample sizes, the bootstrap approach achieves the nominal coverage and may perform better than the CRLB-based parametric approach.

*Index Terms*—Bootstrap, $1/f$-type stochastic processes, wavelets.

## I. INTRODUCTION

In statistical signal processing, $1/f$-type stochastic processes are regarded as useful models for phenomena, which exhibit long-term dependencies among observations, and a statistical self-similarity property [1]. A popular model for $1/f$-type stochastic processes is the fractional Brownian motion (fBm) [2]. The fBm $B_H(t)$ is a zero-mean nonstationary Gaussian random process with the covariance function

$$R_{B_H}(t,s) = E B_H(t) B_H(s) = \frac{\sigma_H^2}{2} \left[ |t|^{2H} + |s|^{2H} - |t-s|^{2H} \right] \tag{1}$$

with the Hurst exponent $0 < H < 1$.

Since the statistics of self-similar processes are invariant to dilations and contractions of the time axis to within an amplitude factor, orthonormal wavelet bases have been proposed as an appropriate

tool to analyze and synthesize $1/f$-type signals [3]. In this regard, several detection and estimation problems involving $1/f$-type stochastic processes are approached using the maximum likelihood (ML) estimation theory [1], [4]. The ML estimation theory for estimating the parameters of a given model, and the Cramér-Rao lower bound (CRLB) approach for estimating the statistical accuracy of the ML estimates, demand that the joint probability density function (PDF) of the observed data is known [5]. Furthermore, for the CRLB to be valid, the observation sample size must be large and tend to infinity. Unfortunately, available signals are usually finite-length, resolution-limited time series. Hence, the CRLB may be of limited value because parameter estimates are biased in consequence of the finite-size effect [6], and its computation may require the (unknown) parameter values.

The bootstrap permits us to solve the statistical problems at hand—in this correspondence, the construction of confidence intervals of $1/f$-type signal parameters—with minimal modeling assumptions, regardless of the validity of asymptotic conditions [7]. The main difficulty is that $1/f$-type stochastic processes are nonstationary, with a long-term correlation structure, whereas typical bootstrap implementations for time series require stationarity and weak dependence among observations [8]. The connection we establish between wavelet analysis and the use of bootstrap relies on the capability of a wavelet decomposition to whiten a broad class of covariance kernels, including the one associated with fBm signals [1], under conditions that are valid for almost any wavelet function [9], [10]. This argument allows us to cast the bootstrap as a procedure of multiple, uncorrelated block-resamplings applied to the stationary, weakly dependent sequences that are obtained from the multiresolution analysis of a single $1/f$-type time series.

This correspondence is organized as follows: Section II provides the description of the proposed algorithm. Computer simulation experiments are reported in Section III. The discussion of the obtained results is developed in Section IV. The conclusions are in Section V.

## II. MOVING BLOCKS BOOTSTRAP OF fBm TIME SERIES

### A. Estimation Algorithm

The parameters of a Gaussian $1/f$ signal $B_H(t)$ are estimated from a noisy time series $r[n]$ composed of $N$ samples

$$r[n] = B_H[n] + w[n] \qquad n = 0, \cdots, N-1. \tag{2}$$

Without loss of generality, the sampling interval is assumed to be unity; $w[n]$ is modeled as a zero-mean white Gaussian noise with variance $\sigma_w^2$ uncorrelated with the signal $B_H[n]$. The discrete wavelet transform (DWT) is used in [1] to decompose $r[n]$ into a collection of wavelet detail coefficients $d_j[n]$, which can be modeled, at first approximation, as stationary, mutually uncorrelated, zero-mean random variables whose variances obey the power law

$$\mathrm{var}\, d_j^2[n] = \sigma_j^2 = \sigma^2 2^{\gamma j} + \sigma_w^2 \qquad j = J_{\mathrm{M_{min}}}, \cdots, J_{\mathrm{M_{max}}} \tag{3}$$

where $0 < \gamma < 2R$ with $\gamma = 2H + 1$; $J_{\mathrm{M_{min}}}$ and $J_{\mathrm{M_{max}}}$ are, respectively, the finest and the coarsest of the $M$ available scales. $R$ is the order of regularity of the selected wavelet function. Provided that $R \geq 1$, which is a mathematical prerequisite for any wavelet function, the decomposition of a nonstationary $1/f$-type stochastic process is known to produce a stationary output [10].

In [1], the variances of the wavelet coefficients for the $M$ scales are fed to an iterative estimate-maximize (EM) algorithm, which computes $\hat{H}$, $\hat{\sigma}^2$, and $\hat{\sigma}_w^2$. In this correspondence, the EM algorithm