Fractal dimension estimation using the fast continuous wavelet transform

Michael J. Vrhel, Chulhee Lee, and Michael Unser

Biomedical Engineering and Instrumentation Program, Bldg. 13, Room 3N17,
National Center for Research Resources, National Institutes of Health,
Bethesda, MD 20892, USA

ABSTRACT

We first review a method for the characterization of fractal signals introduced by Muzy et. al.[4] This approach uses the continuous wavelet transform (CWT) and considers how the wavelet values scale along maxima lines. The method requires a fine scale sampling of the signal and standard dyadic algorithms are not applicable. For this reason, a significant amount of computation is spent evaluating the CWT. To improve the efficiency of the fractal estimation, we introduce a general framework for a faster computation of the CWT. The method allows arbitrary sampling along the scale axis, and achieves $O(N)$ complexity per scale where N is the length of the signal. Our approach makes use of a compactly supported scaling function to approximate the analyzing wavelet. We discuss the theory of the fast wavelet algorithm which uses a duality principle and recursive digital filtering for rapid calculation of the CWT. We also provide error bounds on the wavelet approximation and show how to obtain any desired level of accuracy. Finally, we demonstrate the effectiveness of the algorithm by using it in the estimation of the generalized dimensions of a multi-fractal signal.

1. INTRODUCTION

Multi-fractal dimension estimation is a problem with applications in the quantification of turbulence and diffusion limited aggregation.[1, 5, 7] Muzy et. al. have introduced a wavelet-based approach for estimating the fractal dimensions of a signal.[4] The method involves computing the continuous wavelet transform (CWT) of the fractal signal and examining the maxima of the wavelet coefficients. To obtain sufficient detail in a fractal signal, a large number of samples are required. In addition, the CWT coefficients must be computed for many fine (non-dyadic) scales, to obtain an accurate estimate of how the wavelet maxima vary with respect to scale. For these reasons, the most expensive computation of the algorithm is the calculation of the CWT.

Most approaches for implementing the CWT are Fourier based and therefore achieve $O(NLog(N))$ operations per scale. We are interested in improving the efficiency of the fractal estimation algorithm and propose a faster method for computing general real CWTs at fine sample values along the scale axis. This fine sampling of the scale is obtained by approximating wavelets of various sizes using a compactly supported scaling function, a principle that has been previously used by several authors.[3, 6] What distinguishes our method from those previous approaches is that we achieve $O(N)$ complexity per scale, instead of $O(NLog(N))$. Another unique feature is that we have full control over the approximation error, and that we can obtain any desired level of accuracy.

In the paper, we will discuss the problem of multi-fractal dimension estimation using the wavelet approach. We will then introduce our general method for computing the CWT and briefly discuss its approximating properties. Finally, we will use the fast algorithm in making an estimate of the dimension of a known fractal signal.

0-8194-1928-1/95/$6.00

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

2. MULTI-FRACTAL SIGNALS AND DIMENSION ESTIMATION

For a detailed discussion on using the CWT for estimating the dimension of multi-fractal signals the reader is referred to the paper by Muzy et. al.[4] Here we will cover the basic concepts and introduce an algorithm which can improve the efficiency of the method with no loss in accuracy.

In defining the generalized fractal dimension of a signal $s(t)$ we will consider the mass of the signal within boxes (or intervals) of size ε, as shown in Figure 1. We will denote the mass of the signal in the ith box by $m_i(\varepsilon)$ which is the integration or measure of the signal within that box. As the size of the boxes decreases, with a corresponding increase in the number of boxes, the generalized fractal dimensions D_q of the signal are defined by the following equations:

$$P_q(\varepsilon) = \sum_{i=1}^{M(\varepsilon)} \left(m_i(\varepsilon)\right)^q \qquad -\infty < q < \infty \qquad \text{(1a)}$$

$$\lim_{\varepsilon \to 0^+} P_q(\varepsilon) = C\, \varepsilon^{(q-1)D_q} \qquad \text{(1b)}$$

where $M(\varepsilon)$ is the number of boxes and C is a constant. For $q = 0$ the value D_0 is the well known Hausdorff dimension. The values D_1 and D_2 are known respectively as the information dimension and the correlation dimension. The above equations are the basis for the box counting algorithms.

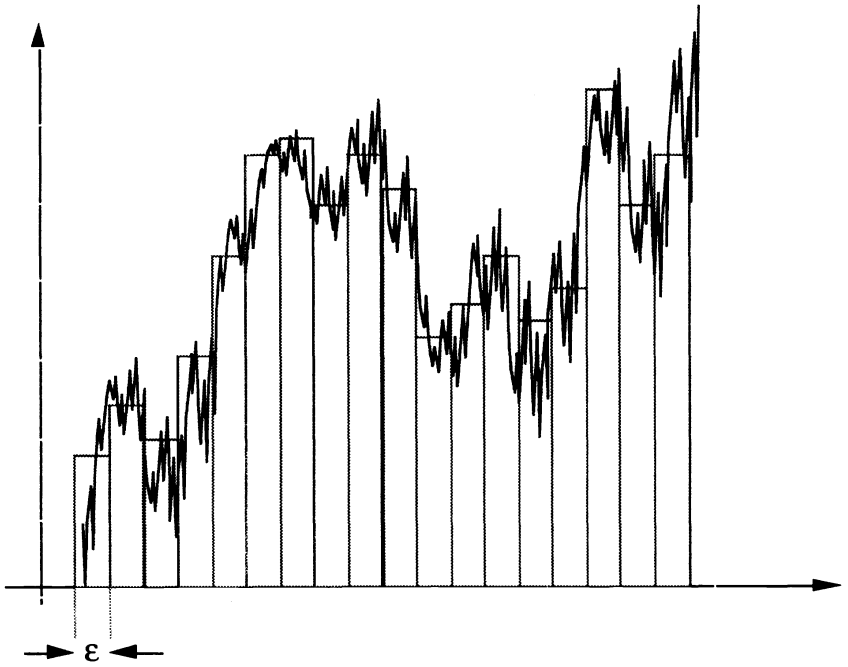


Figure 1: Box counting algorithm for fractal dimension estimation

In using wavelets to estimate the fractal dimension of a signal, the obvious approach would be to substitute the box of size ε with the wavelet of scale α. In this case, the wavelet coefficients are similar to the masses $m_i(\varepsilon)$. Since the CWT is a continuous function we replace the finite summation in (1a) with an integration and substitute the wavelet coefficients for the masses, resulting in the following equation

$$P_q^{\psi}(\alpha) = \int \left(W_{\psi} s(\alpha, \tau)\right)^q d\tau$$

where the wavelet coefficients are

$$W_{\psi} s(\alpha, \tau) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{+\infty} s(t) \psi\left(\frac{\tau - t}{\alpha}\right) dt$$

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

α and τ are respectively the continuously varying scaling and shifting parameters, $s(t)$ is the fractal signal and $\psi(t)$ is the mother wavelet. A problem with such a method is that it is undetermined for values of $q < 0$ where $W_\psi s(\alpha,\tau)=0$. An approach that avoids such problems is thoroughly described by Muzy et.al. and is referred to as the wavelet transform modulus maxima (WTMM) method. The WTMM algorithm considers how the wavelet transform values scale along maxima lines which are related to the singularities of the signal. In the approach, the positions of the local maxima at each scale are calculated resulting in maxima lines like those shown in Fig. 8. The function

$$Z_q(\alpha) = \sum_{l \in L(\alpha)} \left(\sup_{\alpha \le \alpha'} \left| W_\psi s\left(\alpha', b_l(\alpha')\right) \right| \right)^q \qquad (2)$$

is then computed for each scale where the summation is over the set of maxima lines at scale α, the supremum is along each maxima line for scales less than α, and $b_l(\alpha')$ is the spatial location of the maxima line l at scale α'. Note that the form of Eq. (2) is similar to that of Eq. (1a). The fractal dimensions are computed from how $Z_q(\alpha)$ decays or grows with respect to α. Specifically, one uses the relationship

$$\lim_{\alpha \to 0^+} Z_q(\alpha) = C\alpha^{(q-1)D_q}$$

where C is again a constant (similar to Eq. (1b)). From the above relationship, $(q-1)D_q$ is estimated from the slope of the line

$$\log\left(Z_q(\alpha)\right) = C_l + (q-1)D_q \log(\alpha) \qquad (3)$$

using a least squares criterion.

In this algorithm a significant amount of time is spent in computing the wavelet coefficients. Sampled fractal signals are usually quite long in order to contain sufficient information about the self-similarity of the signal. In addition, it is necessary to compute the CWT at more than just the dyadic scales to obtain sufficient points for the estimation of $(q-1)D_q$ in Eq. (3). For this reason an increase in computational efficiency could be achieved by using an algorithm which computes the wavelet transform coefficients with $O(N)$ operations per scale instead of the $O(NLog(N))$ achieved with traditional methods. In the next section, we introduce such an approach, with little loss in accuracy.

3. THE FAST CONTINUOUS WAVELET TRANSFORM

In computing the CWT, the problem is to calculate the values

$$W_\psi s(\alpha,\tau) = (s * \psi_\alpha)(\tau) \qquad (4)$$

where $\psi_\alpha(t) = \left(1/\sqrt{\alpha}\right)\psi(t/\alpha)$.

Direct computation of (4) would involve $O(N^2)$ operations per scale, while an FFT based method would require $O(NLog(N))$ operations per scale. To achieve $O(N)$ complexity per scale, our approach is to approximate the wavelet $\psi_\alpha(t)$ by its orthogonal projection onto a subspace defined by a compactly supported scaling function. To insure the admissibility of the projected wavelet and allow rapid calculation, the compactly supported scaling function φ must satisfy the following three conditions:

(i) $0 \le A \le \sum_{k \in Z} \left|\hat{\varphi}(\omega + 2\pi k)\right|^2 \le B$; where $\hat{\varphi}(\omega)$ is the Fourier transform of $\varphi(t)$.

(ii) $\sum_{k \in Z} \varphi(t-k) = 1$ (partition of unity)

(iii) $\varphi(t/2) = \sum_{k \in Z} h(k)\varphi(t-k)$ (two-scale relation)

The stability property (i) implies that $\{\varphi(x-k)\}_{k \in Z}$ is a Riesz basis of the subspace

$$V_\varphi = \left\{ h(x) = \sum_{k \in Z} c(k)\varphi(x-k) \qquad c \in l_2 \right\}$$

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

and that V_φ is a well defined (closed) subspace of L_2. Property (ii) guarantees that the orthogonal projection of an admissible wavelet ψ onto the subspace defined by φ is an admissible wavelet as well. Property (iii) is important in achieving a fast algorithm since it allows the same filters to be used to compute the wavelet coefficients from octave to octave.

We will replace the computation of the convolution in (4) by its approximation

$$\tilde{W}_\psi s(\alpha,\tau) = (s * \tilde{\psi}_\alpha)(\tau) \qquad (5)$$

where $\tilde{\psi}_\alpha$ is the orthogonal projection of $\psi_\alpha(t) = \alpha^{-1/2}\psi(t/\alpha)$ onto the subspace V_φ. Mathematically the orthogonal projection $\tilde{\psi}_\alpha$ can be expressed as

$$\tilde{\psi}_\alpha(t) = \sum_{k\in Z} p_\alpha(k)\varphi(t-k) \quad \text{where } p_\alpha(k) = \langle \psi_\alpha(t), \mathring{\varphi}(t-k)\rangle \qquad (6a)$$

$$= \sum_{k\in Z} q_\alpha(k)\mathring{\varphi}(t-k) \quad \text{where } q_\alpha(k) = \langle \psi_\alpha(t), \varphi(t-k)\rangle. \qquad (6b)$$

where $\mathring{\varphi}(t)$ is the dual of $\varphi(t)$. The sequence $q_\alpha(k)$ is finite since ψ_α and φ_k are compactly supported by definition. This finite support is the motivation for using this dual representation instead of the formula $\tilde{\psi}_\alpha(t) = \sum_k p_\alpha(k)\varphi(t-k)$. The sequence $p_\alpha(k)$ is usually infinite, even when φ and ψ are compactly supported. This follows from the fact that the dual of a symmetrical compactly supported scaling function is generally infinite, except for the Haar case where φ is a unit rectangular pulse[2]. Since we are approximating wavelets that are either symmetric or anti-symmetric, it is essential to use a symmetric scaling function in order to preserve the wavelet symmetry properties.

In order to compute the samples at a particular scale, it is necessary to know the FIR filter coefficients $q_\alpha(k)$ associated with ψ_α. In practice, however, it is not essential to have an FIR filter for every scale. Instead, we will only use Q FIR filters to calculate the CWT for the Q scales in the first octave

$$\alpha_j = \alpha_0 2^{\frac{j}{Q}} \qquad j = 0,.....Q-1 \qquad (7)$$

and then use property (iii) of the scaling function to compute the CWT for Q scales in each of the higher octaves.

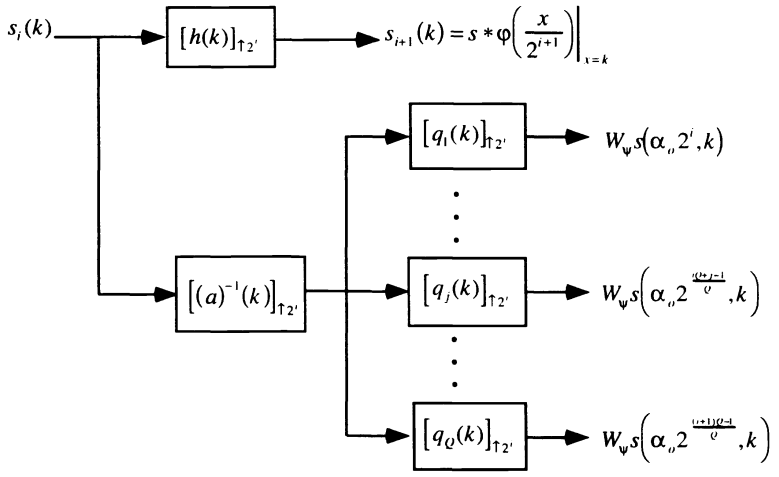


Fig. 2: Schematic representation of the general algorithm.

Substituting the approximation (6b) into (5), sampling, and performing simple algebraic operations produces the following general algorithm:

Initialization: The algorithm requires the computation of the continuous convolution of the signal with the scaling function. This convolution is approximated as

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

$$s_o(k) = (s * \varphi)(x)\big|_{x=k} \equiv b * s[k] \qquad (8)$$

where $b(k)$ and $s[k]$ are respectively the samples of the scaling function and input signal.

Iteration: Once $s_o(k)$ is computed, the wavelet coefficients are determined using

$$\tilde{W}_\psi s(\alpha_o 2^{\frac{iQ+j-1}{Q}}, k) = ([q_j * (a)^{-1}]_{\uparrow 2^i} * s_i)(k) \quad j = 0, \ldots\ldots Q-1$$

where $a(k-l) = \langle \varphi_l, \varphi_k \rangle$, $\left((a)^{-1} * a\right)(k) = \delta[k]$, and the notation $[v]_{\uparrow 2^i}$ indicates that $v(k)$ is expanded by a factor of 2^i (i.e. 2^{i-1} zeros between sequence samples). The sequence $s_{i+1}(k)$ is computed iteratively from $s_i(k)$ using $s_{i+1}(k) = (s_i * [h]_{\uparrow 2^i})(k)$. A block diagram of the system is shown in Figure 2.

4. APPROXIMATION ERROR

There are essentially two ways to control the error of the approximation: change the scaling function, or adjust the size of the finer scale wavelet α_o (cf. Eq. (7)). In either case, reducing the error may result in longer FIR filters. The precise formulation of the approximation power of a scaling function is provided by the Strang-Fix conditions[8]. These conditions imply that we can control the error by dilating the wavelet or increasing the order of the representation. For a scaling function of order L, the approximation error e is bounded as follows:

$$e = \left\| \psi_\alpha - \tilde{\psi}_\alpha \right\| \le \frac{C_\varphi}{\alpha^L} \left\| \psi^{(L)} \right\| = \frac{C_\varphi \cdot C_\psi}{\alpha^L} \qquad (9)$$

where $\psi^{(L)}$ is the Lth derivative of ψ, and $\tilde{\psi}_\alpha$ is the approximation of ψ_α in V_φ. The numerator of the right hand side is the product of two constants : a first term C_φ, which is a function of the representation, and a second wavelet-dependent term $C_\psi = \left\| \psi^{(L)} \right\|$, which can be pre-computed by integration in the time or frequency domain.

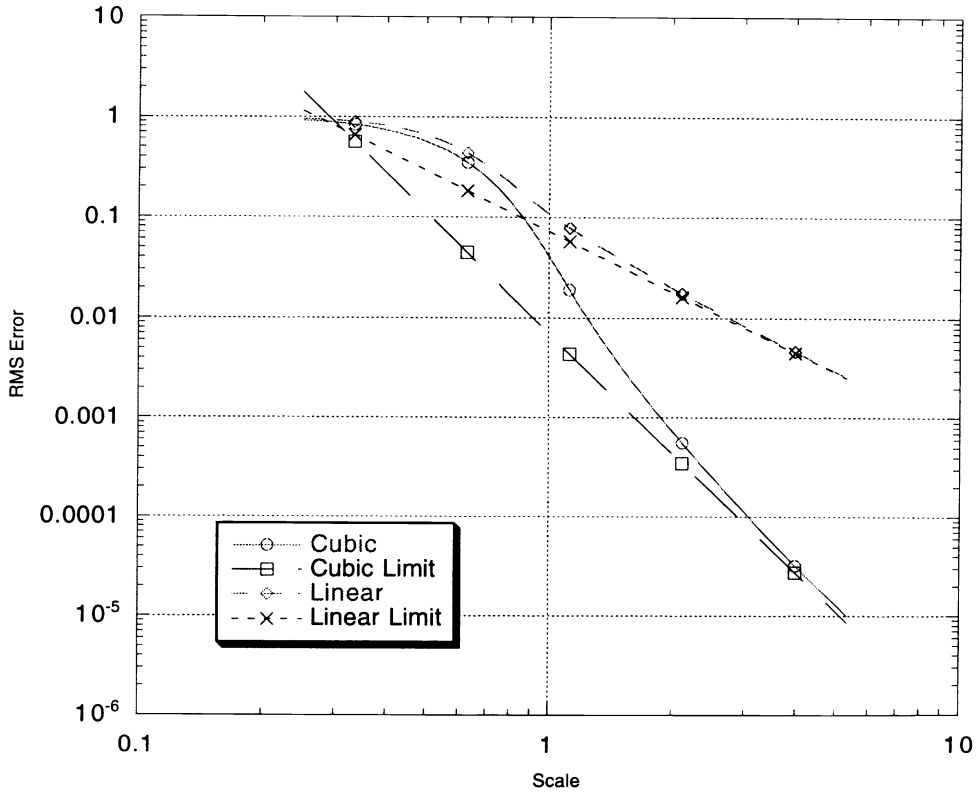


Fig. 3: Cubic and Linear approximation errors for anti-symmetric wavelet

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

Equation (9) indicates that the approximation error decreases with the Lth power of the scale. Clearly, the error will be maximum at the finer scale α_o. Our design strategy is therefore to select the parameters α_o and L such as to maintain this error below a certain threshold (worst case scenario). For this purpose, we can make use of the asymptotic relation

$$\left\|\psi_\alpha - \tilde{\psi}_\alpha\right\| \equiv \frac{C_2 \cdot C_\psi}{\alpha^L} \qquad \text{as} \quad \alpha \to +\infty,$$

where the constant C_2 is given by

$$C_2 = \frac{1}{L!}\left(\sum_{k\neq 0}\left|\hat{\varphi}^{(L)}(2\pi k)\right|^2\right)^{1/2}.$$

For our implementation, we used polynomial splines of degree n, which have an order of approximation $L = n+1$. If φ is the B-spline of degree n, which we denote by $\beta^n(x)$, then the constant C_2 can be computed explicitly:

$$C_2 = \frac{1}{L!}\sqrt{\sum_{k\neq 0}\left(\frac{L!}{(2\pi k)^L}\right)^2} = \sqrt{\frac{|B_{2L}|}{(2L)!}}$$

where $|B_L|$ is the Bernoulli's number of degree L.

The $O(\alpha^{-L})$ behavior of the error as a function of the scale was verified experimentally for several examples of wavelets. Figure 3 shows the error as a function of scale for a cubic and linear spline least squares approximation of a 1st derivative Gaussian wavelet.

5. FAST B-SPLINE IMPLEMENTATION

The centered B-spline functions are compact and satisfy the three properties given in Section 3. Once the wavelet has been selected, and the approximation error level specified (which selects α_o and L), the FIR filters

$$q_j(k) = \left\langle \psi_{\alpha_j}, \beta_k^n \right\rangle \qquad j = 0,.....Q-1 \qquad (10)$$

can be calculated, where $\beta_k^n = \beta^n(x-k)$ is a shifted B-spline of degree n. These filters are computed by numerical integration methods. If the wavelet is symmetric or anti-symmetric with $q(k)$ of length n_q, then this filter requires $(n_q+1)/2$ multiplications and $n_q - 1$ additions per sample.

For an nth degree B-spline implementation, $h(k)$ in property (iii) of Section 3 is given by $h(k) = u_2^n(k)$, where $u_2^n(k)$ is a binomial filter.[9] The filter $\left[u_2^{n_1}\right]_{\uparrow 2^l}$ can be decomposed into a cascade of filters given by

$$\left[u_2^{n_1}(k)\right]_{\uparrow 2^l} = \frac{1}{2^{n_1}}\underbrace{\left[u_2^0\right]_{\uparrow 2^l} * \left[u_2^0\right]_{\uparrow 2^l} \cdots * \left[u_2^0\right]_{\uparrow 2^l}}_{n_1+1 \text{ times}}(k-k_o)$$

where the shift $k_o = (i+1)(n_1+1)/2$ is due to the definition of $u_2^0(k)$ which is $u_2^0(k) = \delta[k] - \delta[k-1]$. Since $\left[u_2^0\right]_{\uparrow 2^l}$ consists of only one addition, the binomial filter $\left[u_2^{n_1}(k)\right]_{\uparrow 2^l}$ can be implemented with n_1+1 additions per sample. Alternatively, $\left[u_2^{n_1}(k)\right]_{\uparrow 2^l}$ can be implemented in the "à trous" fashion with $(n_1+3)/2$ multiplications and n_1+1 additions per sample.

The remaining filter in the algorithm is the IIR filter $\left[(a)^{-1}(k)\right]_{\uparrow m} = \left[(b^{2n+1})^{-1}(k)\right]_{\uparrow 2^l}$, which can be implemented as a recursive filter. Since $(a)^{-1}$ is a symmetrical all pole filter, the Z-transform of its up-sampled version $\left[(a)^{-1}(k)\right]_{\uparrow m}$ can be written in the following standard form

$$A_{n_1}(z^m) = \frac{d_0}{\left[z^{n_1 m} + z^{-n_1 m}\right] + \left(\sum_{k=1}^{n_1-1} c_k\left[z^{km} + z^{-km}\right]\right) + c_0}$$

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

where d_0 and $\{c_k,\ k = 0,.....n_1 - 1\}$ are constant coefficients. The filter can be expressed as a cascade of simple first order causal/anti-causal components $A_{n_1}(z^m) = d_o \prod_{i=1}^{n_1} A(z^m; z_i)$ where $A(z^m; z_i)$ is defined as

$$A(z^m; z_i) = \frac{1}{(1 - z_i z^{-m})} \frac{-z_i}{(1 - z_i z^m)}.$$

This yields the following recursive filter equations

$$y^+(k) = x(k) + z_i y^+(k - m) \qquad k = m,.....N$$

$$\begin{aligned} y(k) &= -z_i y^+(k) + z_i y(k + m) \\ &= z_i \left(y(k + m) - y^+(k) \right) \end{aligned}$$

The only practical difficulty is to provide the correct initial values for $k = 1,.....,m$ on each side of the signal. This can be achieved by using mirror boundary conditions. With this technique, if the signal length is N and the expansion factor is m, then this filter will require $2N + 2k_o m$ additions and $2N + (2k_o + 1)m$ multiplications, where $k_o = \log(\varepsilon_b / z_i)$ is a parameter that depends upon the required accuracy (ε_b) for the initialization (boundary conditions).

6. EXPERIMENTAL RESULTS

We implemented the fast wavelet algorithm for the anti-symmetric wavelet

$$\psi_{dem}(t) = \begin{cases} -K_o \dfrac{t}{\alpha_o} e^{\frac{-(t/\alpha_o)^2}{2}} & ; \quad \left|\dfrac{t}{\alpha_o}\right| \le 5 \\ 0 & ; \quad \text{otherwise} \end{cases}$$

where K_o is a constant that insures that $\|\psi\| = 1$. The values of α_o were selected to achieve a worst case error of $\varepsilon = 0.01$. We determined α_o by solving (9) as a function of α. An alternative approach, which may be more appropriate for splines of higher degree, is to compute a few error values which can then be used to determine an upper error bound of the form $C\alpha^{-L}$ where $C \ge C_2 \cdot C_\psi$. For a B-spline of degree 3 the value of α_o is 1.25.

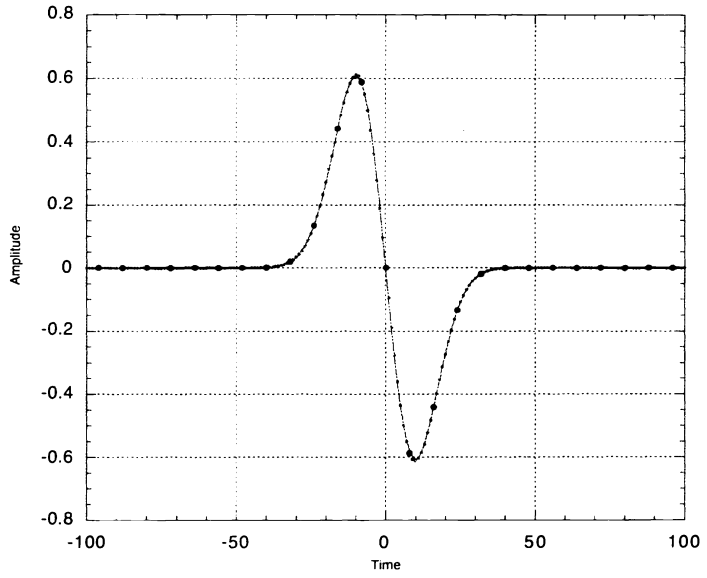


Fig. 4: Anti-symmetric wavelet (solid) and its cubic LS approximation (small circles) at an rms error of 0.01. The black circles represent the spline knots.

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

The wavelet is shown in Figure 4 with its cubic spline least squares (LS) approximation. We computed the filters q_j (cf. Eq. (10)) for the scales $\alpha = \alpha_0 2^{\frac{j}{12}}$ which provides a discretization of each octave that corresponds to the musical notes $\left(A, A^{\#}, B, C, C^{\#}, \ldots\right)$. The initialization approximation was used, as given in Eq. (8) and the 2-scale filter $\left[u_2^{n_1}(k)\right]_{\uparrow 2^l}$ was implemented using the zero-padded algorithm. The impulse response of the system is shown in Fig. 5.

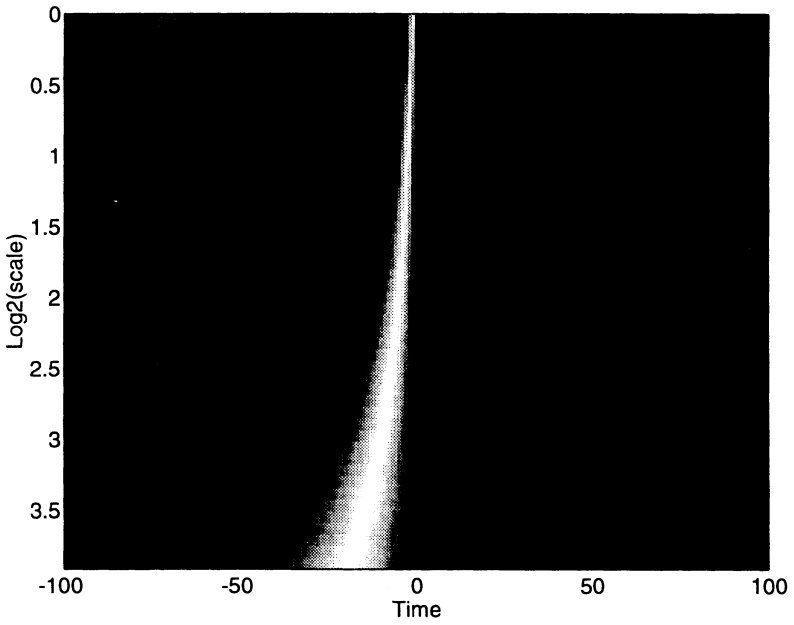


Fig. 5: Wavelet transform of unit impulse with the cubic LS approximation of the anti-symmetric wavelet for twelve voices per octave.

To demonstrate that our fast wavelet transform has sufficient accuracy for use in the WTMM algorithm, we used the fast CWT in the estimation of a multi-fractal signal of known dimension. The fractal signal is a generalized devil staircase and is shown in Fig. 6. The scalogram of the generalized devils staircase, as computed by the fast CWT, is shown in Figure 7. The maxima lines of the scalogram were found and are displayed in Figure 8. From the wavelet values along those maxima lines $Z_q(\alpha)$ was calculated for $q = -10,\ldots,10$ at each scale (cf. Eq. (2)). The slopes $\hat{\tau}(q)$ for the lines

$$\log\left(Z_q(\alpha)\right) = C_l + \hat{\tau}(q)\log(\alpha) \qquad q = -10,\ldots,10$$

were estimated using a least squares criterion. The values computed are shown in Figure 9 with the exact equation for $\tau(q)$. The estimated values are in close agreement with their actual values.

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

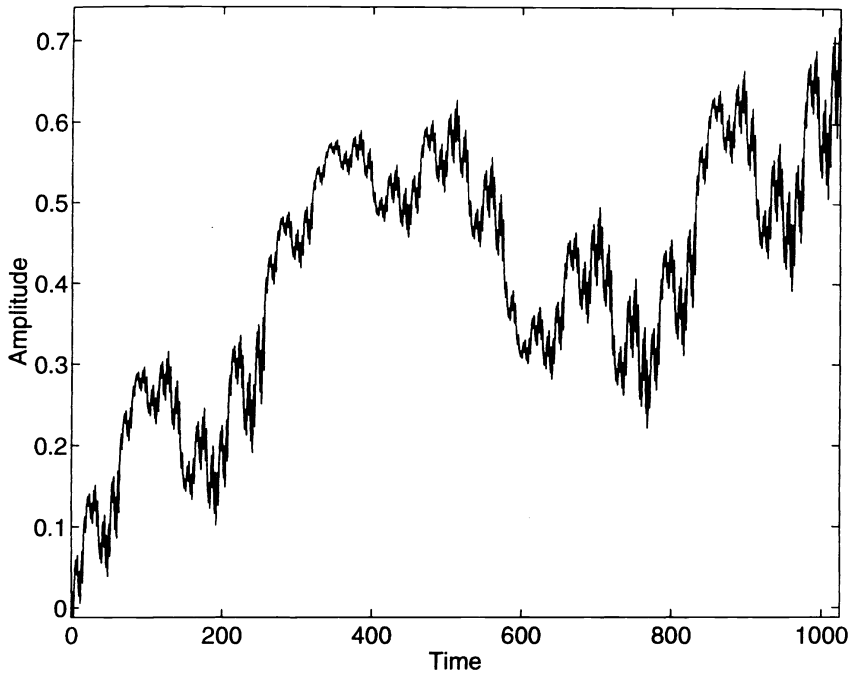
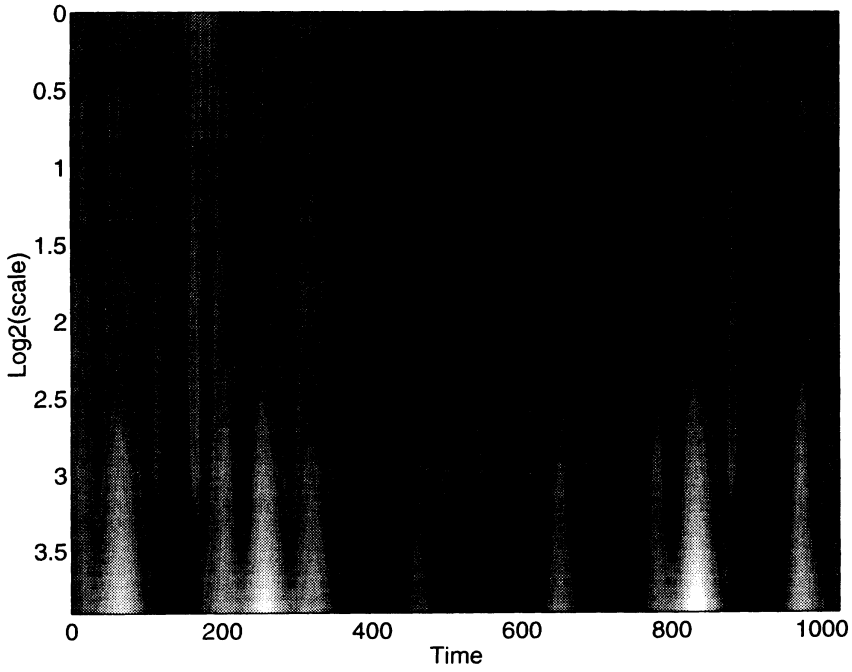


Fig. 6: Generalized devil staircase signal.



Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

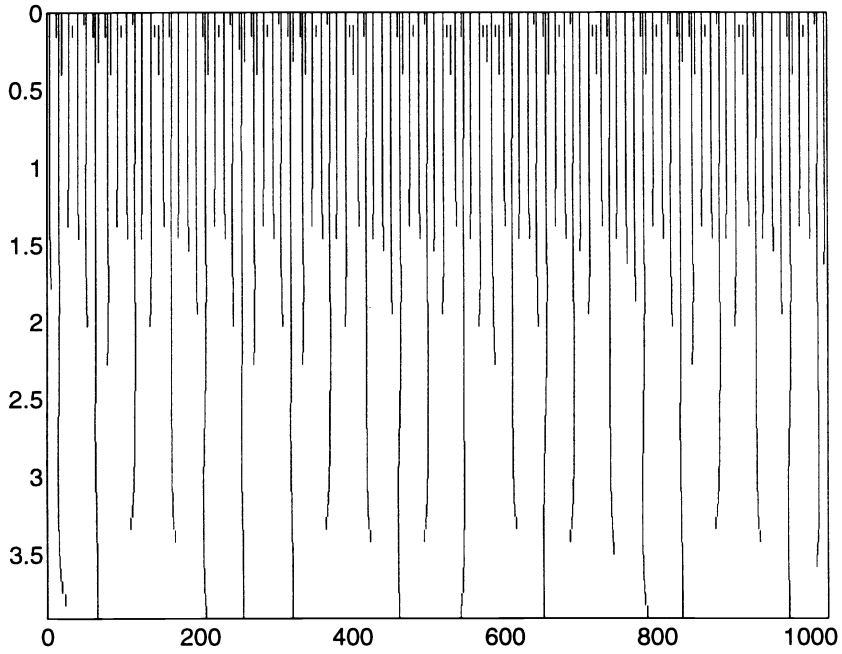


Fig. 8: Maxima lines for scalogram shown in Figure 6.

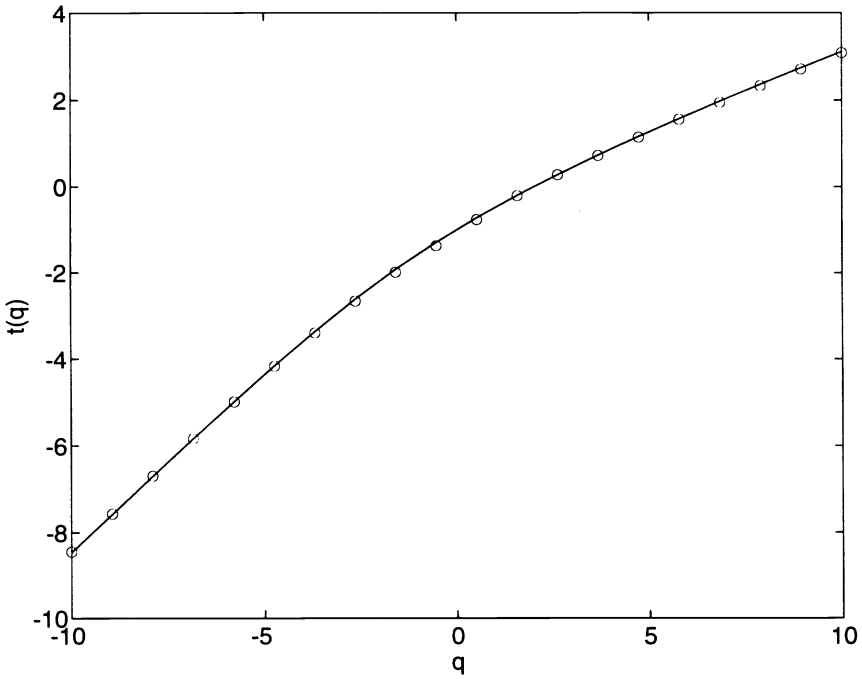


Fig. 9: Comparison of estimated and exact values of $\tau(q) = (q-1)D_q$. The solid line is the exact form of $\tau(q)$ and the circles represent the values estimated from the signal.

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms

7. CONCLUSION

We have introduced a method to improve the efficiency of wavelet-based fractal estimation schemes. Sampled fractal signals are typically long, and wavelet coefficients at scales finer than dyadic sampling are required. This favors our approach since the method we have introduced achieves $O(N)$ operations per scale while previous methods obtained $O(N\,Log(N))$ calculations per scale. We have complete control over the accuracy of our method and implemented it in an algorithm for estimating the generalized fractal dimensions of a multi-fractal signal.

9. REFERENCES

1. F. Argoul, A. Arneodo, J. Elezgaray, G. Grasseau and R. Murenzi, "Wavelet analysis of the self-similarity of diffusion-limited aggregates and electrodeposition clusters", *Physical Review A*, Vol. 41, No. 10, pp. 5537-5560, 1990.
2. I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadelphia, PA, 1992.
3. S. Maes, "A Fast Quasi-Continuous Wavelet Transform Algorithm", *Proc. Workshop on Time, Frequency, Wavelets and Multiresolution Theory*, INSA-Lyon, 1994.
4. J.F. Muzy, E. Bacry and A. Arneodo, "Multifractal formalism for singular signals based on wavelet analysis.", in: Y. Meyerand and S. Rogue, ed., *Progress in Wavelet Analysis and Applications*, Paris, 1993, pp. 323.
5. J.F. Muzy, E. Barcy and A. Areneodo, "Wavelets and multifractal formalism for singular signals: Application to turbulence data", *Physical Review Letters*, Vol. 67, No. 25, pp. 3515-3518, 1991.
6. O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms", *IEEE Trans. Info. Th.*, Vol. IT-38, No. 2, pp. 569-586, 1992.
7. M. Schroeder, *Fractals, Chaos, Power Laws*, W. H. Freeman and Company, New York, NY, 1991.
8. G. Strang and G. Fix, "A Fourier analysis of the finite element variational method", in: *Constructive Aspect of Functional Analysis*, Edizioni Cremonese, Rome, 1971, pp. 796-830.
9. M. Unser, A. Aldroubi and M. Eden, "The L_2 polynomial spline pyramid", *IEEE Trans. PAMI*, Vol. 15, No. 4, pp. 364-379, 1993.

Downloaded from SPIE Digital Library on 19 Oct 2011 to 128.178.48.127. Terms of Use: http://spiedl.org/terms