FAST CONTINUOUS WAVELET TRANSFORM

Michael Vrhel, Chulhee Lee, Michael Unser

Biomedical Engineering and Instrumentation Program, NCRR, Bldg. 13, Room 3W13
National Institutes of Health, Bethesda, MD 20892 USA

ABSTRACT

We introduce a general framework for the efficient computation of the continuous wavelet transform (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 derive error bounds on the wavelet approximation and show how to obtain any desired level of accuracy through the use of higher order representations. Finally, we present examples of implementation for different wavelets using polynomial spline approximations.

1. INTRODUCTION

The continuous wavelet transform (CWT) has received significant attention in its ability to perform a time-scale analysis of signals. Its ability to zoom in on singularities has made it an attractive tool in the analysis of non-stationary and fractal signals [1-3]. Mathematically, the continuous wavelet transform (CWT) of a continuous signal $s(t)$ can be defined as

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

where α and τ are respectively the continuously varying scaling and shifting parameters, and $\psi(t)$ is the mother wavelet.

In practice, the variables α and τ are sampled over the plane of values. Fast algorithms exist for computing the CWT at dyadic scale values $\alpha=2^i$, where $i\in Z$ [2, 3]. In particular, if the wavelet is derived from a multi-resolution analysis [3], then Mallat's algorithm provides sample values $\alpha=2^i$, $\tau=2^i k$ with a global $O(N)$ complexity. A related approach is the "à trous" algorithm which supplies the sample values $\alpha=2^i$ $\tau=k$ $i,k\in Z$ with $O(N)$ computations per scale [4, 5]. The "à trous" algorithm has also been used to compute the CWT at the integer sample values $\alpha=i$, $\tau=k$, again with $O(N)$ operations per scale [6]. An algorithm for complex wavelet analysis with $O(N)$ complexity per scale is discussed in [7]. This last method allows for an arbitrary sampling of the scale but is restricted to Gabor-like wavelets (i.e., modulated Gaussians). Except for those special cases, the most efficient algorithms to date typically require $O(NLog(N))$ computations per scale [8, 9].

We introduce a fast method for computing general CWTs at the sample values $\alpha=\alpha_o 2^{\frac{i}{Q}}$, $\tau=k$ where Q is a number selected to achieve a desired exponential sampling rate 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 [5, 10]. What distinguishes our method from those previous approaches is that we achieve $O(N)$ complexity per scale, instead of the $O(NLog(N))$ reported in [10]. Another unique feature is that we have full control over the approximation error, and that we can achieve any desired level of accuracy.

2. SCALING FUNCTION REQUIREMENTS

The problem is to compute the values

$$W_\psi s(\alpha,\tau)=(s*\psi_\alpha)(\tau) \qquad (1)$$

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

Direct computation of (1) 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 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) $A\le\sum_{k\in Z}|\hat{\varphi}(\omega+2\pi k)|^2\le B$;

where A and B are strictly positive constants and $\hat{\varphi}(\omega)$ 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

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

0-7803-2431-5/95 $4.00 © 1995 IEEE

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 as discussed in the next section .

3. THE GENERAL ALGORITHM

In order to achieve $O(N)$ complexity per scale, we will replace the computation of the convolution in (1) by its approximation

$$\tilde{W}_\psi s(\alpha,\tau) = (s * \tilde{\psi}_\alpha)(\tau) \qquad (2)$$

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

$$\tilde{\psi}_\alpha(t) = \sum_k q_\alpha(k)\mathring{\varphi}(t-k) \qquad (3)$$

where $\mathring{\varphi}(t)$ is the dual of $\varphi(t)$, $q_\alpha(k)$ is the sequence $q_\alpha(k) = \langle \psi_\alpha, \varphi_k \rangle$, and $\varphi_k = \varphi(t-k)$. The sequence $q_\alpha(k)$ is finite since ψ_α and φ_k are compactly supported by definition.

The motivation for using this dual representation instead of the formula $\tilde{\psi}_\alpha(t) = \sum_k p_\alpha(k)\varphi(t-k)$ with $p_\alpha(k) = \langle \psi_\alpha, \mathring{\varphi}_k \rangle$, is that 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 [1]. 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

$$\alpha_j = \alpha_o 2^{\frac{j}{Q}} \qquad j = 0,.....Q-1 \qquad (4)$$

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

Substituting the approximation (3) into (2), 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

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

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

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

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 1.

4. THE 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. (4)). 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 [11]. 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 is bounded as follows:

$$\varepsilon = \left\|\psi_\alpha - \tilde{\psi}_\alpha\right\| \le \frac{C_\varphi}{\alpha^L}\left\|\psi^{(L)}\right\| = \frac{C_\varphi \cdot C_\psi}{\alpha^L} \qquad (5)$$

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.

Equation (5) 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 ε (worst case scenario). For this purpose, we can make use of the asymptotic relation

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

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 experiments, 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. For our experiments, we selected an error threshold of ε=0.01 and choose to determine the corresponding finer scale parameter α_o by solving (5) 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 $C \geq C_2 \cdot C_\psi$.

5. FAST B-SPLINE IMPLEMENTATION

The centered B-spline functions are compact and satisfy the three properties given in Section 2. Once the wavelet has been selected, and the approximation error level specified, the FIR filters

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

can be calculated, where $\beta_k^n = \beta^n(x-k)$. 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 2 is given by $h(k) = u_2^n(k)$, where $u_2^n(k)$ is a binomial filter [12]. The filter $\left[u_2^{n_1}\right]_{\uparrow 2^i}$ can be decomposed into a cascade of filters given by

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

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 $\left[u_2^0\right]_{\uparrow 2^i}$ consists of only one addition, the binomial filter $\left[u_2^{n_1}(k)\right]_{\uparrow 2^i}$ can be implemented with n_1+1 additions per sample.

The remaining filter in the algorithm is the IIR filter $\left[(a)^{-1}(k)\right]_{\uparrow m} = \left[(b^{2n+1})^{-1}(k)\right]_{\uparrow 2^i}$, 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

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

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

$$A(z^m; z_i) = \frac{1}{(1 - z_i z^{-m})}\frac{-z_i}{(1 - z_i z^m)}. \qquad (6)$$

Values of d_0 and $\{z_i,\ i = 1,.....n_1\}$ for different spline orders are given in [12]. This yields the following recursive filter equations

$$y^+(k) = x(k) + z_i y^+(k-m) \qquad k = m,.....N \quad (7)$$

$$\begin{aligned} y(k) &= -z_i y^+(k) + z_i y^+(k+m) \\ &= z_i\left(y(k+m) - y^+(k)\right) \end{aligned} \qquad (8)$$

The only practical difficulty is to provide the correct initial values for $k = 1,.....,m$ on each side of the signal. This can be done 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. RESULTS AND DISCUSSION

We implemented the algorithm for the wavelets

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

$$\psi_{max}(t) = \begin{cases} K_o\left(1 - \left(\dfrac{t}{\alpha_o}\right)^2\right) e^{\frac{-(t/\alpha_o)^2}{2}} - K_1 & \left|\dfrac{t}{\alpha_o}\right| \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

where K_o is a constant that insures that $\|\psi\| = 1$ and K_1 guarantees the admissibility of ψ_{max} . The values of α_o were selected to achieve a worst case error of $\varepsilon = 0.01$. The value of α_o is 2.69 for ψ_{deriv} and 3.32 for ψ_{max} when using a B-spline of order 1. For a B-spline of order 3 the value of α_o is 1.25 for ψ_{deriv} and 1.40 for ψ_{max} . The wavelet ψ_{max} is shown in Figure 2 with its cubic spline least squares (LS) approximation. The wavelet and its approximation are virtually indistinguishable. We computed the filters q_j for the case of $Q = 12$. This provides a discretization of each octave that corresponds to the musical notes $(A, A^\#, B, C, C^\#, ...)$. The wavelet transform of an impulse function for the cubic approximation of ψ_{max} is shown in Figure 3.

7. CONCLUSION

The general method that was described evaluates the CWT with $O(N)$ complexity per scale. The procedure can approximate any desired wavelet shape and can provide arbitrary sampling along the scale axis. In addition, the approximation error can be easily controlled by adjusting the fine scale resolution or by the use of higher order splines.

8. ACKNOWLEDGMENTS

The authors wish to acknowledge I. Daubechies for her useful input on the error of least squares spline approximations.

REFERENCES

[1] I. Daubechies, Ten lectures on wavelets. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[2] O. Rioul and M. Vetterli, "Wavelets and signal processing", *IEEE Signal Processing Magazine*, vol. 8, pp. 11-38, 1991.

[3] S. G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, pp. 674-693, 1989.

[4] M. J. Shensa, "The discrete wavelet transform : wedding the à trous and Mallat algorithms", *IEEE Trans. Signal Processing*, vol. 40, pp. 2464-2482, 1992.

[5] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms", *IEEE Trans Information Theory*, vol. IT-38, pp. 569-586, 1992.

[6] M. Unser, A. Aldroubi and S. J. Schiff, "Fast implementation of the continuous wavelet transform with integer scales", *IEEE Trans. Signal Processing*, in press.

[7] M. Unser, "Fast Gabor-like windowed Fourier and continuous wavelet transforms", *IEEE Signal Processing Letters*, vol. 1, pp. 76-79, 1994.

[8] D. L. Jones and R. G. Baraniuk, "Efficient approximation of continuous wavelet transforms", *Electronics Letters*, vol. 27, pp. 748-750, 1991.

[9] S. Maes, "The wavelet transform in signal processing, with application to the extraction of the speech modulation model features", Universite Catholique De Louvain, Ph.D., 1994.

[10] S. Maes, "A Fast Quasi-Continous Wavelet Transform Algorithm", *Workshop on Time, Frequency, Wavelets and Multiresolution Theory*, 1994.

[11] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method", in Constructive Aspect of Functional Analysis, Ed. Rome: Edizioni Cremonese, pp. 796-830, 1971.

[12] M. Unser, A. Aldroubi and M. Eden, "The L_2 polynomial spline pyramid", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 364-379, 1993.

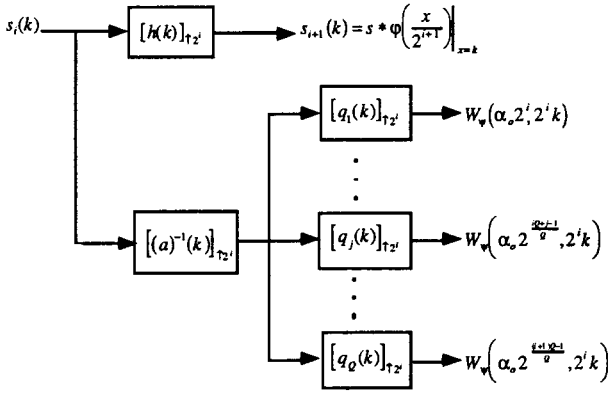

Figure 1: Block Diagram of System

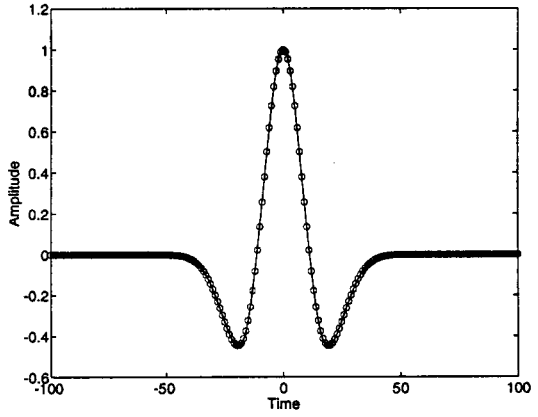

Figure 2: Mexican hat wavelet (solid) and its cubic LS approximation (circles) at an RMS error of 0.01

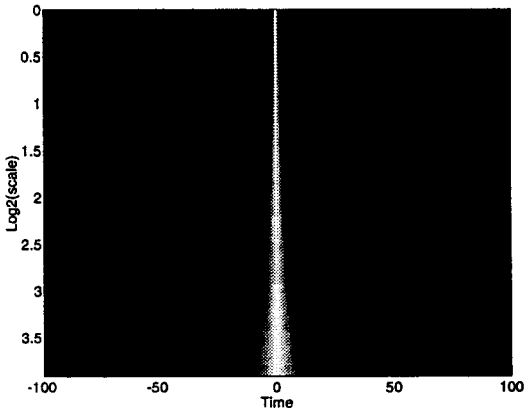

Figure 3: Wavelet Transform of unit impulse with the cubic LS approximation of the mexican hat wavelet and twelve voices per octave.