FAST ALGORITHMS FOR RUNNING WAVELET ANALYSES

Michael Unser and Akram Aldroubi

Biomedical Engineering and Instrumentation Program, Bldg 13, Room 3W13,
National Center for Research Resources, National Institutes of Health,
Bethesda, Maryland 20892, USA.

ABSTRACT

We present a general framework for the design and efficient implementation of various types of running (or over-sampled) wavelet transforms (RWT) using polynomial splines. Unlike previous techniques, the proposed algorithms are not necessarily restricted to scales that are powers of two; yet they all achieve the lowest possible complexity : $O(N)$ per scale, where N is signal length. In particular, we propose a new algorithm that can handle any integer dilation factor and use wavelets with a variety of shapes (including Mexican-Hat and cosine-Gabor). A similar technique is also developed for the computation of Gabor-like complex RWTs. We also indicate how the localization of the analysis templates (real or complex B-spline wavelets) can be improved arbitrarily (up to the limit specified by the uncertainty principle) by increasing the order of the splines. These algorithms are then applied to the analysis of EEG signals and yield several orders of magnitude speed improvement over a standard implementation.

Keywords : oversampled wavelet transform, quasi-continuous wavelet transform, fast algorithms, splines, non-dyadic scales, signal analysis, time-frequency analysis, uncertainty principle.

1. INTRODUCTION

The wavelet transform (WT) decomposes a signal s by performing inner products with a collection of analysis functions $\{\psi_{a,b}\}$, which are scaled and translated version of the wavelet ψ; i.e.,

$$\left(W_\psi s\right)(b;a) := \langle s, \psi_{a,b} \rangle, \tag{1}$$

$$\psi_{a,b}(x) = a^{-1/2}\psi(a^{-1}(x-b)). \tag{2}$$

The amplitude of the WT therefore tends to be maximum at those scales and locations where the signal most resembles the analysis template (matched filter interpretation). The WT also provides a natural tool for time-frequency signal analysis since each template $\psi_{a,b}$ is predominantly localized in a certain region of the time-frequency plane with a central frequency that is inversely proportional to a. What distinguishes it from the short-time Fourier transform is the multiresolution nature of the analysis. This property enables the WT to zoom in on singularities and makes it very attractive for the analysis of transient signals[5, 7, 20].

0-8194-1627-4/94/$6.00

The distinction between the various types of transforms essentially depends on the way in which the scale and translation parameters (a and b, respectively) are discretized[6]. In the case of the continuous wavelet transform (the most redundant representation), these parameters vary in a continuous fashion[9]. At the other extreme, the analysis is performed only for scales that are powers of two (i.e., $a = 2^i$) and for translation parameters that are critically sampled; i.e, $b = k \cdot 2^i$. This type of non-redundant feature extraction is the one that is used for a signal representation in terms of wavelet bases[1, 4, 14, 15]. Such wavelet decompositions can be computed efficiently using the fast wavelet algorithm[14], which has an overall $O(N)$ complexity where N is the length of the input signal.

For analysis purposes, it is often desirable to use a finer sampling of both the scale and the translation parameters. Accordingly, we will adopt the following discrete definition of the *running* wavelet transform (RWT)

$$W_\psi s[k;a] := a^{-1/2} \sum_{l \in Z} s[l] \psi\left(\frac{l-k}{a}\right) = (w_a * s)[k], \tag{3}$$

where the usual integral is replaced by a summation over the integers. The term "running" is used to indicate that the WT is now evaluated for every shift index $k \in Z$. Thus, for a fixed value of a, the interpretation of (3) is that of a discrete convolution between the input signal s and the enlarged (and rescaled) wavelet template $w_a[k] = a^{-1/2} \psi(-k/a)$. Note that the evaluation of (2) requires $O(N^2)$ operations per scale for the direct evaluation, or $O(N\log N)$ if one uses an FFT-based algorithm[11]. A more efficient approach is the so-called "à trous" algorithm[19, 26], which has an $O(N)$ complexity per scale. Unfortunately, it is only applicable for scales that are powers of two.

The purpose of this paper is to introduce several alternative spline-based approaches that are equally effective, but allow for a finer discretization of the scale parameter. In particular, we will describe a very efficient mechanism for convolving a signal with a spline wavelet dilated by an integer factor m, and not just a power of two. The disadvantages of the method are minimal because splines are flexible enough to approximate virtually any desirable wavelet shape. In addition, splines provide the standard illustration of a multiresolution analysis; as a result, there are many examples of such wavelet bases in the literature[10, 13, 30, 35]. Also, unlike other scaling functions and wavelets, splines have a simple explicit analytical form in both the time and frequency domain, which often facilitates their manipulations.

The presentation is organized as follows. In Section 2, we start with a brief review of splines and a discussion of their relevant properties. In Section 3, we present methods for the evaluation of real valued spline RWTs. After a brief review of the "à-trous" algorithm for dyadic scales, we introduce a new procedure that is equally effective but is applicable for all integer scales. In Section 4, we consider the complex case and present efficient algorithms for the evaluation of Gabor-like (or Morlet) RWTs. In Section 5, we apply our algorithm to the analysis of EEG signals. We also briefly indicate how the present methodology may provide efficient implementations for other signal processing tasks such as Gaussian smoothing and narrowband lowpass filtering.

2. SPLINE FUNCTIONS : MAIN PROPERTIES

A polynomial spline of order n (odd) is a function of the continuous variable x that is a polynomial of degree n for each interval $[k,k+1), k \in Z$. The polynomial segments are connected at the integer knots in a way that guarantees the continuity of the function and its derivatives up to order n-1 [24]. Any such spline can be represented by a weighted sum of shifted B-splines

$$s(x) = \sum_{k \in Z} c(k) \beta^n (x-k), \tag{4}$$

where $c(k)$ are the so-called B-spline coefficients[24, 25]. $\beta^n(x)$ is the central B-spline of order n; it is a symmetrical bell-shape function that is obtained from the (n+1)-fold convolution of a unit rectangular pulse (i.e., the B-spline of order zero).

Given a discrete signal $\{s[k]\}, k \in Z$, there is a unique spline of the form (4) that provides an exact interpolation; i.e. $s(x)|_{x=k} = s[k]$. This mapping is expressed by the following discrete convolution equations[31, 34]

$$s[k] = b^n * c(k) \quad \Leftrightarrow \quad c(k) = \left(b^n\right)^{-1} * s[k] \ , \tag{5}$$

where

$$b^n[k] := \beta^n(x)\big|_{x=k} \tag{6}$$

is the discrete B-spline kernel of degree n, and where $\left(b^n\right)^{-1}$ denotes the inverse filter operator, which exists and is stable[2] for any order n.

A basic spline function enlarged by an integral factor m is still a spline with knots at the integers. This simply follows from the fact that if a dilated function is a polynomial of degree n for each enlarged interval $[k \cdot m, (k+1) \cdot m)$, then it is also polynomial on the smaller integer intervals $[k,k+1), k \in Z$. Consequently, there must exist a certain sequence $u_m^n(k)$ such that

$$\beta^n(x/m) = \sum_{k \in Z} u_m^n(k) \beta^n(x-k). \tag{7}$$

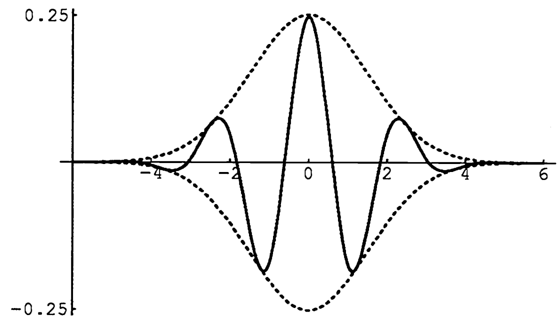
We have shown that the z-transform of u_m^n is given by[36]

$$U_m^n(z) = \frac{z^{k_0}}{m^n} \left(\sum_{k=0}^{m-1} z^{-k} \right)^{n+1}, \tag{8}$$

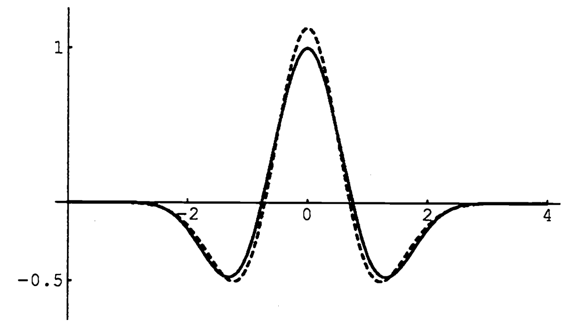
with $k_0 = (n+1)(m-1)/2$. Thus, this sequence can be generated from the (n+1)-fold convolution of a discrete rectangular pulse of length m.

3. FAST SPLINE WAVELET TRANSFORMS (REAL)

We will now use this scaling property of splines to derive two algorithms for the efficient computation of real-valued spline RWT. The first approach is a rather standard extension of Mallat's fast wavelet algorithm for the over-sampled case. It uses zero-padded filters and relies heavily on the multi-resolution properties on the underlying functions spaces; it is therefore only applicable for scales that are powers of two. The second approach, which is specific to splines, has lesser restrictions and it is applicable for any integer scale a=m.



(a)



(b)

Fig. 1 : Two examples of cubic spline wavelets. (a) Semi-orthogonal B-spline wavelet and its Gaussian envelope (dashed line). (b) Spline (2nd derivative) approximation of a Mexican Hat (dashed line).

In all cases, we require the wavelet to be a polynomial spline that is characterized by its B-spline expansion

$$\psi(x) = \sum_{k \in Z} p(k) \beta^n (x-k), \tag{9}$$

where p is a given FIR sequence of coefficients. If the desired wavelet is not itself a spline, we can still approximate it by constructing its least squares approximation[33], or by simply interpolating the function values at the integers. Two examples of such cubic spline wavelets (n=3) are shown in Fig. 1; their corresponding B-spline coefficients are also given in Table 1. The first one is the cubic B-spline wavelet which is very similar to a cosine-Gabor function and is therefore extremely well localized in time and frequency[32]. The second corresponds to the second derivative of a quintic spline, which provides a very close approximation of a Mexican Hat (2nd derivative of a Gaussian).

3.1 The "à trous" algorithm

If the analysis is restricted to dyadic scales ($a = 2^i$), it is possible to use the modified "à trous" version of the fast WT algorithm for the non-sampled case. To derive this algorithm directly, we use the two-scale equation (7) for B-splines with m=2 and the wavelet definition (9), and show that the sampled versions of the dilated B-splines and wavelets satisfy the recursive equations

$$b^n_{2^i}[k] := \beta^n(x/2^i)\big|_{x=k} = [h]_{\uparrow 2^{i-1}} * b^n_{2^{i-1}}[k] \tag{10}$$

$$w^n_{2^i}[k] := \psi^n(x/2^i)\big|_{x=k} = [p]_{\uparrow 2^i} * b^n_{2^i}[k] \tag{11}$$

where $h = u^n_2$ is the binomial filter and where p represents the wavelet B-spline coefficients in (9); the symbol $[\cdot]_{\uparrow m}$ denotes the upsampling operator by a factor of m. Hence, in order to analyze the discrete signal $s[k]$, one starts by computing

$$c_0[k] = b^n * s[k] \tag{12}$$

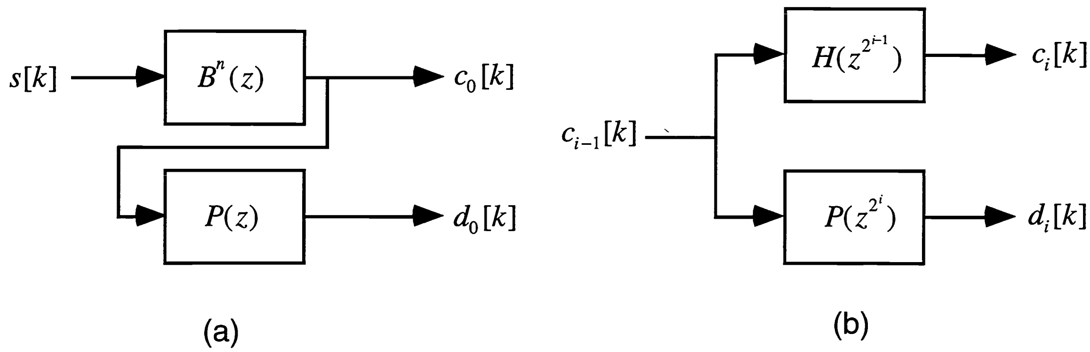


Fig. 2 : Schematic representation of the "à-trous" (or zero-padded) RWT algorithm for dyadic scales. (a) Initialization. (b) Basic processing module.

$$d_0[k] := w_1 * s[k] = p * c_0[k] \tag{13}$$

where $b^n[k]$ is the discrete (FIR) B-spline kernel defined by (6). The running wavelet decomposition is evaluated iteratively for i=1 down to I by successive convolution with the enlarged kernels h and p

$$\begin{cases} c_i[k] := b^n_{2^i} * s[k] = [h]_{\uparrow 2^{i-1}} * c_{i-1}[k] \\ d_i[k] := w^n_{2^i} * s[k] = [p]_{\uparrow 2^i} * c_i[k] \end{cases} \tag{14}$$

In this formulation, p is arbitrary and can be chosen to approximate any desired wavelet shape. This algorithm is schematically represented in Fig. 2. The relevant FIR filter parameters for the cubic spline wavelets in Fig. 1 are given in Table 1.

TABLE 1 : FILTER PARAMETERS FOR THE CUBIC SPLINES WAVELETS IN FIG. 1.

Sequences	Templates (symmetrical)
Discrete cubic B-spline : b^3	$\boldsymbol{b}^3=\frac{1}{6}(1, 4, 1)$
Binomial kernel (n=3) : $h = u_2^3$	$\boldsymbol{h}=\frac{1}{8}(1, 4, 6, 4, 1)$
Cubic B-spline wavelet : p	$\boldsymbol{p}_a=(\cdots, 0.6018, -0.4584, 0.196, -0.04159, 0.003075, 0.0000248)$
Cubic spline Mexican-Hat: p	$\boldsymbol{p}_b=(-1, 2,-1)$
Lowpass (cardinal spline)	$\boldsymbol{p}^3_{\text{card}}=(\cdots, 1.7321, -0.4641, 0.12436, -0.03332, 0.00893, -0.00239, 0.0006410, -0.0001718, \ldots)$

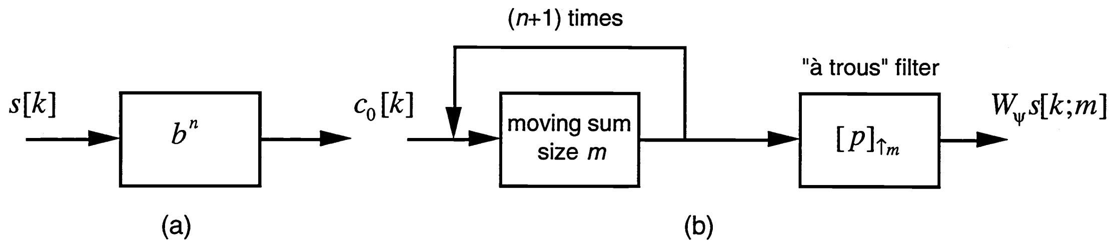


Fig. 3 : Schematic representation of the fast RWT algorithm for real spline wavelets and integer scales. (a) Initialization. (b) Individual scale processing.

3.2 Fast wavelet transforms with integer scales

The technique that is described next is more general in the sense that it is valid for all integer scales $a=m$. Using the wavelet definition (9), it is not difficult to show that the sampled version of the dilated analysis template that appears on the right hand side of (2) is given by

$$w_m[k] = m^{-1/2} \cdot [p]_{\uparrow m} * b_m^n[k] = m^{-1/2} \cdot [p]_{\uparrow m} * u_m^n * b^n[k] \qquad (15)$$

where $b_m^n[k] := \beta^n(k/m)$ is the discrete B-spline enlarged by a factor of m, and where the last factorization directly follows from (7). Accordingly, the running WT can be computed from the following convolution

$$W_m s[k;m] = m^{-1/2} \cdot [p]_{\uparrow m} * u_m^n * c_0[k] \qquad (16)$$

where $c_0[k]$ is the sequence defined by (12), and where u_m^n is the digital filter specified in (8).

This filtering can be implemented very efficiently since the convolution with u_m^n is equivalent to a cascade of (n+1) moving sums (cf. Eq. (8)), each of which can be performed with two additions only per sample[27]. The last step is a convolution with the up-sampled kernel p ("à trous" filter); the filter coefficients should also be re-scaled appropriately to account for the all remaining normalization factors, such as the constant $m^{-1/2}$ in (2). The whole procedure is summarized in Fig. 3. Note that there is also a very similar approach that provides an exact spline computation of the WT in its continuous formulation; for a complete treatment, we refer the reader to [36].

This approach has the same complexity per scale as the one in Section 3.1 ($O(N)$). What distinguishes it from multiresolution-based approaches is that it is non-iterative across scale. Its structure is therefore well suited for a parallel implementation with one processor per scale. Moreover, the algorithm is not restricted to scales that are powers of two.

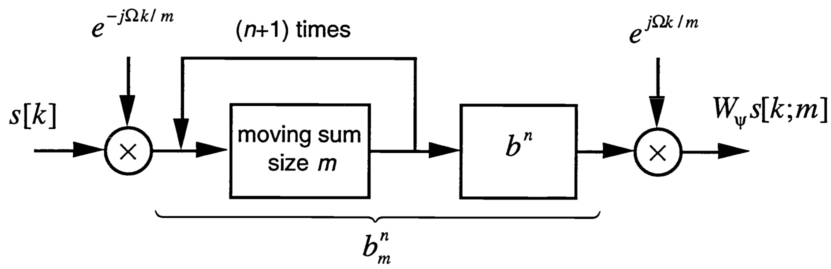


Fig. 4 : Schematic representation of the fast RWT algorithm for Gabor-like complex B- spline wavelets.

4. FAST GABOR-LIKE WAVELET TRANSFORMS (COMPLEX)

The standard technique for constructing a complex wavelet is to multiply a certain window function $\varphi(x)$ by a complex exponential. The optimum time-frequency localization is achieved when $\varphi(x)$ is a Gaussian (Morlet or Gabor wavelet)[8, 17].

4.1 Complex B-spline wavelet transform (integer scales)

Here, we approximate this Gaussian by a B-spline of order n and define the following complex B-spline wavelet

$$\psi(x) = \beta^n(x)e^{j2\pi x} \quad \xleftrightarrow{Fourier} \quad \hat{\psi}(f) = \text{sinc}^{n+1}(f-1), \tag{17}$$

which satisfies the usual admissibility condition for the CWT. The localization of this wavelet can be chosen arbitrarily close to the optimum since $\beta^n(x)$ converges to a Gaussian as n goes to infinity.

In order to derive our algorithm[29], we substitute (17) in the definition of the discretized WT and rewrite this expression as follows

$$W_\psi s[l,m] = e^{j\Omega l/m}\sum_{k\in Z} s_m[k]b^n_m[k-l] = e^{j\Omega l/m}\left(b^n_m * s_m\right)[l] \tag{18}$$

where the auxiliary signal $s_m[k]$ is defined as

$$s_m[k] = m^{-1/2}s[k]e^{-j\Omega k/m}, \tag{19}$$

and where $b^n_m[k] := \beta^n(k/m)$ is the discrete B-spline enlarged by a factor of m. These two equations suggest a simple computational procedure, which is outlined in Fig. 4. The input signal is first pre-multiplied by the complex exponential $e^{-j\Omega k/m}$. This auxiliary signal is then filtered with the enlarged B-spline window function and finally demodulated. Since the filtering with a B-spline window of size m can be performed recursively using the same technique as before, we obtain a fast wavelet algorithm with a complexity $O(N)$ independent of m.

In practice, the order of the B-spline window does not need to be very high. For instance, the time-frequency bandwidth product of the complex cubic B-spline wavelet (n=3) is within 0.5% of the limit specified by the uncertainty principle, which should be sufficient for most applications. The corresponding cubic spline WT can be computed with as few as 8 (real) multiplications and 22 additions per output sample.

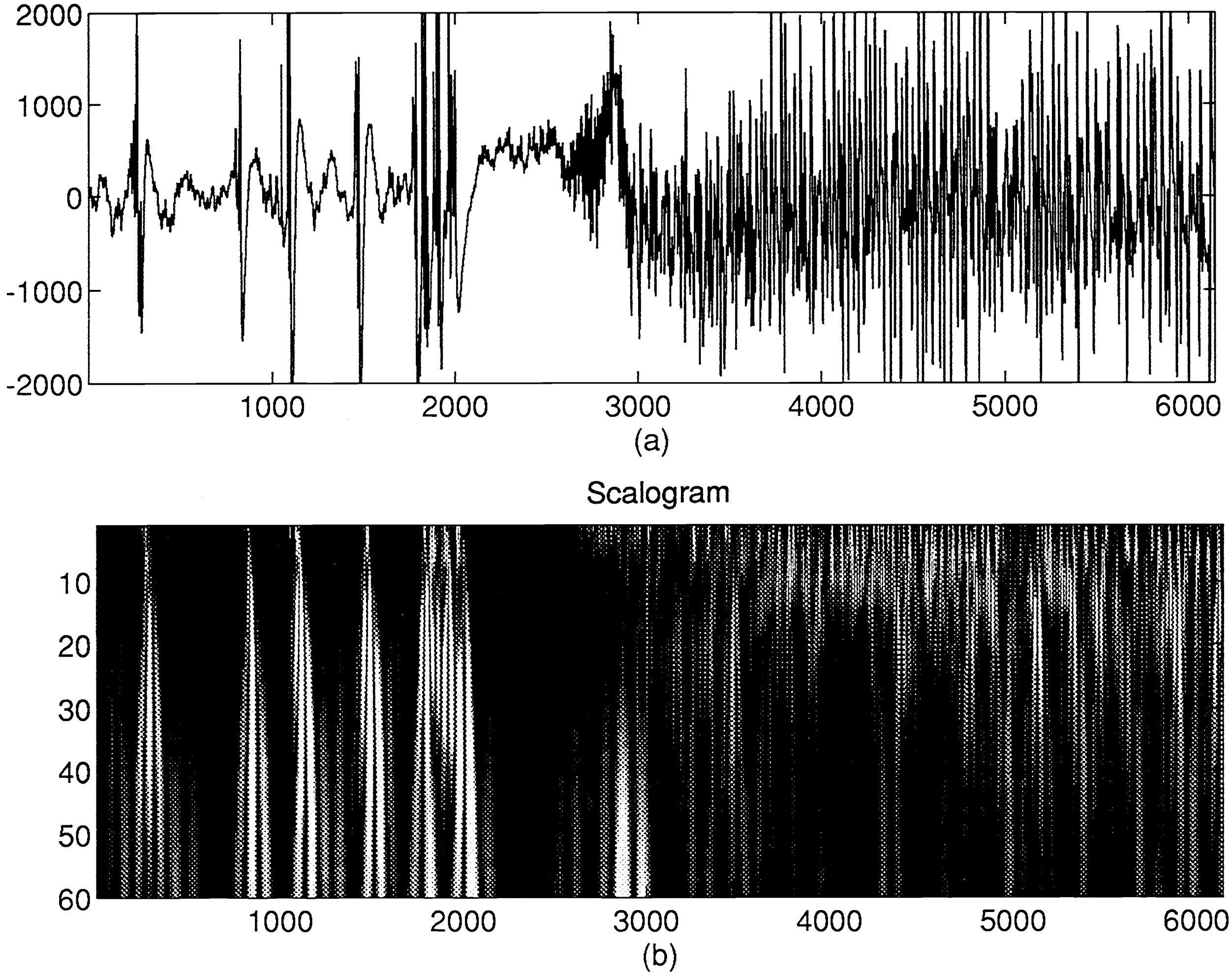


Fig. 5 : Example of signal analysis using the continuous wavelet transform.
(a) EEG signal, (b) local energy map in the wavelet domain.

4.2 Complex wavelet transform (arbitrary scales)

It is also possible to extend the previous approach for arbitrary scales. For this purpose, we can select a quasi-gaussian window function φ generated from the n-fold convolution symmetrical exponentials (instead of a rectangular pulse as in the B-spline case). Each elementary exponential filter can be implemented recursively with as few as 2 adds and 2 multiplies per samples. Here too, the number of computations is independent of the window size and we have an $O(N)$ algorithm. By increasing the number of elementary filtering modules, the time-frequency localization of the analysis wavelets can be chosen arbitrarily close to the limit specified by the uncertainty principle, a property that follows directly from the central limit theorem. More details on this approach can be found elsewhere[29]. Finally, we should note that the same modulation technique, which has been used for many years in analog spectrum analyzers, can also be applied for computing the windowed Fourier transform[28, 29].

5. RESULTS AND APPLICATIONS

In this section, we present an application of the real RWT algorithm in Section 3.2 to the analysis of EEG data[21]. We also briefly indicate how the same computational ideas could also be used for efficient Gaussian scale-space smoothing, as well as narrowband lowpass filtering.

5.1 Analysis of EEG signals

All fast WT algorithms were implemented using the Pro-Matlab software package[16] with some of the critical subroutines (iterated sum and "à trous" filter) coded in C to speed up computations. The wavelets were all chosen to be cubic splines (n=3); the corresponding FIR filtering kernels can be found in Table I.

Fig. 5a shows a 6144-point EEG signal recorded at 200 Hz using a Telefactor Beehive 64CTE. This 30 second recording from an epileptic patient was obtained from an electrode overlying the seizure focus on the surface of the frontal lobe of the brain. The normalized Mexican-Hat scalogram of this signal with m=1,..., 64 is shown in Fig. 5b. This scalogram represents the local energy distribution of the corresponding wavelet transform. It was computed as follows

$$e_m(k) = \frac{w_m^2(k)}{\sum_{k=1}^{N} w_m^2(k)/N}. \tag{20}$$

Note that the denominator provides an estimate of the variance in the mth channel since the wavelet coefficients are zero mean by definition. Clearly the maxima of this representation can be used to localize the most salient features of our signal. A simple 5% threshold was applied to this energy map to identify areas of unexpected signal activity. This method was successful in resolving the polyspike and wave activity, a slow potential shift, and finally, the onset of a well developed seizure.

This simple detection strategy is very similar in principle to the statistical threshold technique described by Schiff[22, 23], which relies on the generation of surrogate data with equivalent spectral characteristics. Specifically, if we were analyzing a stationary random Gaussian signal, we would find that the coefficients in each WT channel are Gaussian distributed with a certain variance σ_m^2. If in addition, we assume that this random process has the same spectral power density as our EEG recording, this variance can be estimated from the denominator of (20). Therefore, the quantity $e_m(k)$ defined by (16) may be interpreted as a measure of the quantity of information with respect to such a reference statistical model. The larger $e_m(k)$, the rarer the event under the stationary Gaussian assumption.

We found our fast WT algorithm to provide several orders of magnitude speed improvement over previous implementations. Specifically, the computation of the fast wavelet transform in Fig. 5b required less then 45 seconds CPU on a low-cost workstation (Macintosh IIfx), while the direct implementation with full length Mexican-Hat filters used in our previous studies typically took several days on a Sun SparcStation IPx[23].

5.2 Scale-space filtering

Scale-space filtering is a technique used in computer vision by which a signal is convolved with Gaussian kernels of increasing width[3, 12, 37]. The main characteristic usually analyzed is the systematic behavior of the extrema as a function of scale. Since the B-splines tend to a Gaussian as the order is increased, we can use the algorithm described above without the last step (i.e., $p=1$) to compute the scale-space representation in a very efficient way. For $n=3$, the relative integrated root mean square error between the cubic spline and its Gaussian approximation is 2.38%, which should be sufficient for most applications. This filtering technique can easily be extended in higher dimensions through the use of a separable formulation. Interestingly, the corresponding tensor product splines are nearly circularly symmetric because they closely approximate a Gaussian, the sole function that is both separable and circularly symmetric.

5.3 Low complexity narrowband lowpass filtering

In general, the complexity of a lowpass filter is directly related to its selectivity, at least in the FIR case. The nice feature of the present algorithm is that it offers a simple mechanism for dilating filters by an integer factor m without any increase in computations. The effect of this scaling in the frequency domain is a reduction of the bandwidth in the same proportion. We can take advantage of this property to obtain efficient implementation of narrowband filters, in a way that is similar to the interpolated FIR approach of Neuvo et al.[18]. The lowpass filter design strategy that we propose here is to approximate the sinc function (ideal lowpass filter) using B-splines. We can then directly use these coefficients in the algorithm described above to obtain a fast implementation of a lowpass filter with an adjustable cutoff frequency $f_0 = 1/(2m)$.

One approach is to consider a cardinal or *fundamental* spline of order n which is a function that is one at the origin and vanishes at all other integers. These functions have been shown to converge to sinc(x) as n goes to infinity[2]. The corresponding weighting sequence in (9) is $p = \left(b^n\right)^{-1}$ (the convolution inverse of the discrete B-spline of order n). Although this filter has an infinite impulse response, it has a very efficient recursive implementation[31]. Alternatively, we can use a truncated version. The first few significant filter coefficients for the cubic spline case ($\boldsymbol{p}^3_{\text{card}}$) are also given in Table I.

Note that it is always possible to design filters with better lowpass characteristics by either increasing the order of the splines or by simply approximating an enlarged sinc function.

6. CONCLUSION

In this paper we have described several computational techniques for the efficient evaluation of RWTs. Two of these procedures (the real and complex algorithms in Sections 3.2 and 4.1, respectively) are specific to splines and offer certain advantages that deserve to be emphasized :

• Both algorithms can handle any integer dilation factor; unlike previous approaches, they are not restricted to scales that are powers of two.

• Their complexity does not depend on the size of the convolution kernel; it the same for all scales.

• The complexity per scale is only proportional to the size of the input signal ($O(N)$). Moreover, the proportionality constant is surprisingly small; for example, the implementation of cubic spline Mexican Hat wavelet requires no more than 10 additions plus 2 multiplications per sample (if one neglects the initialization phase which is only performed once).

• Both algorithms are non-iterative across scale and their structure is very regular. They are well suited for a parallel implementation with one processor per scale.

• Together these procedures offer great flexibility in the choice of the filtering templates. Examples of Mexican Hat, cosine-Gabor, and Morlet-type wavelets have been presented. In addition, we have shown that it is possible to design spline wavelets (real or complex) with a time-frequency localization that is as close as one wishes to the limit specified by the uncertainty principle.

Acknowledgments : We thank Dr. Steven J. Schiff of the Children's National Medical Center for encouraging us to develop fast wavelet transform algorithms and providing us with the EEG data.

REFERENCES

1. A. Aldroubi and M. Unser, "Families of multiresolution and wavelet spaces with optimal properties", *Numerical Functional Analysis and Optimization*, Vol. 14, No. 5-6, pp. 417-446, 1993.
2. A. Aldroubi, M. Unser and M. Eden, "Cardinal spline filters : stability and convergence to the ideal sinc interpolator", *Signal Processing*, Vol. 28, No. 2, pp. 127-138, August 1992.
3. J. Babaud, A.P. Witkin, M. Baudin and R.O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, No. 1, pp. 26-33, January 1986.
4. I. Daubechies, "Orthogonal bases of compactly supported wavelets", *Comm. Pure Appl. Math.*, Vol. 41, pp. 909-996, 1988.
5. I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", *IEEE Trans. Inform. Theory*, Vol. 36, No. 5, pp. 961-1005, September 1990.
6. I. Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
7. P. Flandrin, "Some aspects of non-stationary signal processing with emphasis on time-frequency and time-scale methods", in: J.M. Combes, A. Grossmann and P. Tchamitchian, ed., *Wavelets: Time-frequency methods and phase space*, Springer-Verlag, New York, 1989, pp. 68-98.
8. D. Gabor, "Theory of communication", *J. Inst. Elec. Eng.*, Vol. 93, No. III, pp. 429-457, 1946.
9. A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape", *SIAM J. Math. Anal.*, Vol. 15, No. 4, pp. 723-736, July 1984.
10. A. Haar, "Zur Theorie der orthogonalen Funktionensysteme", *Math. Ann.*, Vol. 69, pp. 331-371, 1910.
11. D.L. Jones and R.G. Baraniuk, "Efficient approximation of continuous wavelet transforms", *Electronics Letters*, Vol. 27, No. 9, pp. 748-750, April 1991.
12. J. Koenderink, "The structure of images", *Biol. Cybern.*, Vol. 50, pp. 363-370, 1984.
13. P.-G. Lemarié, "Ondelettes à localisation exponentielles", *J. Math. pures et appl.*, Vol. 67, No. 3, pp. 227-236, 1988.
14. S.G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, No. 7, pp. 674-693, 1989.
15. Y. Meyer, *Ondelettes et Opérateurs I : Ondelettes*, Hermann, Paris, France, 1990.
16. C. Moler, J. Little and S. Bangert, *PRO-MATLAB User's Guide*, The Math Works Inc., Sherborn, MA, 1987.

17. J. Morlet, G. Arens, I. Fourgeau and D. Giard, "Wave propagation and sampling theory", *Geophysics*, Vol. 47, pp. 203-236, 1982.

18. Y. Neuvo, C.-Y. Dong and S.K. Mitra, "Interpolated finite impulse response filters", *IEEE Trans. Acoust. Speech Signal Proc.*, Vol. 32, pp. 563-570, June 1984.

19. O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms", *IEEE Trans Information Theory*, Vol. IT-38, No. 2, pp. 569-586, March 1992.

20. O. Rioul and M. Vetterli, "Wavelets and signal processing", *IEEE Signal Processing Magazine*, Vol. 8, No. 4, pp. 11-38, October 1991.

21. S. Schiff, A. Aldroubi, M. Unser and S. Sato, "Fast wavelet transformation of EEG", *Electroencephalography and Clinical Neurophysiology*, in press.

22. S.J. Schiff, "Resolving time-series structure with a controlled wavelet transform", *Optical Engineering*, Vol. 31, pp. 2492-2495, November 1992.

23. S.J. Schiff, J. Heller, S.L. Weinstein and J. Milton, "Wavelet transforms and surrogate data for electroencephalographic spike and seizure detection", *Optical Engineering*, in press.

24. I.J. Schoenberg, "Contribution to the problem of approximation of equidistant data by analytic functions", *Quart. Appl. Math.*, Vol. 4, pp. 45-99, 112-141, 1946.

25. I.J. Schoenberg, *Cardinal spline interpolation*, Society of Insdustrial and Applied Mathematics, Philadelphia, PA, 1973.

26. M.J. Shensa, "The discrete wavelet transform : wedding the à trous and Mallat algorithms", *IEEE Trans. Signal Processing*, Vol. 40, No. 10, pp. 2464-2482, October 1992.

27. M. Unser, "Recursion in Short Time Signal Analysis", *Signal Processing*, Vol. 5, No. 3, pp. 229-240, May 1983.

28. M. Unser, "Comments on 'A new approach to recursive Fourier transform'", *Proc. IEEE*, Vol. 76, No. 10, pp. 1395-1396, October 1988.

29. M. Unser, "Fast Gabor-like windowed Fourier and continuous wavelet transforms", *IEEE Signal Processing Letters*, Vol. 1, No. 5, pp. 1-4, May 1994.

30. M. Unser and A. Aldroubi, "Polynomial splines and wavelets - A signal processing perspective", in: C.K. Chui, ed., *Wavelets - A tutorial in theory and applications*, Academic Press, San Diego, 1992, pp. 91-122.

31. M. Unser, A. Aldroubi and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 3, pp. 277-285, March 1991.

32. M. Unser, A. Aldroubi and M. Eden, "On the asymptotic convergence of B-spline wavelets to Gabor functions", *IEEE Trans. Information Theory*, Vol. 38, No. 2, pp. 864-872, March 1992.

33. M. Unser, A. Aldroubi and M. Eden, "Polynomial spline signal approximations : filter design and asymptotic equivalence with Shannon's sampling theorem", *IEEE Trans. Information Theory*, Vol. 38, No. 1, pp. 95-103, January 1992.

34. M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing. Part I : theory", *IEEE Trans. Signal Processing*, Vol. 41, No. 2, pp. 821-833, February 1993.

35. M. Unser, A. Aldroubi and M. Eden, "A family of polynomial spline wavelet transforms", *Signal Processing*, Vol. 30, No. 2, pp. 141-162, January 1993.

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

37. A.P. Witkin, "Scale-space filtering", *Proc. 4th Int. Joint Conf. Artificial Intelligence*, pp. 1019-1022, 1983.