ELSEVIER

Signal Processing 57 (1997) 103–119

SIGNAL
PROCESSING

Fast continuous wavelet transform: A least-squares formulation

M.J. Vrhel[a,]*, C. Lee[b], M. Unser[a]

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

[b] *Electrical Engineering Department, Yonsei University, 134 Shinchon-Dong Seodaemun-Ku, Seoul 120-749, South Korea*

Received 14 August 1995

Abstract

We introduce a general framework for the efficient computation of the real continuous wavelet transform (CWT) using a filter bank. The method allows arbitrary sampling along the scale axis, and achieves $O(N)$ complexity per scale where N is the length of the signal. Previous algorithms that calculated non-dyadic samples along the scale axis had $O(N\log(N))$ computations per scale. Our approach approximates the analyzing wavelet by its orthogonal projection (least-squares solution) onto a space defined by a compactly supported scaling function. We discuss the theory which uses a duality principle and recursive digital filtering for rapid calculation of the CWT. We derive error bounds on the wavelet approximation and show how to obtain any desired level of accuracy through the use of longer filters. Finally, we present examples of implementation for real symmetric and anti-symmetric wavelets. © 1997 Published by Elsevier Science B.V.

Zusammenfassung

Wir stellen eine allgemeine Methode vor für die effiziente Berechnung der kontinuierlichen Wavelettransformation (KWT) anhand von Filterbänken. Die Methode erlaubt eine beliebige Unterteilung der Skalierungsachse, und erreicht eine Berechnungskomplexität von $O(N)$ pro Skalierung, wo N die Länge des Signals darstellt. Gängige Algorithmen mit einer nicht diadischen Unterteilung der Skalierungsachse benötigen $O(N\log(N))$ Berechnungen. Die vorgestellte Methode approximiert die analisierende Wavelet durch eine orthogonale Projektion (minimaler quadratischer Fehler) auf einen mit kompakten Skalierungsfunktionen definierten Raum. Wir stellen eine Theorie vor welche auf dem Dualitätsprinzip beruht und welche rekursive numerische Filter für eine schnelle Berechnung der KWT gebraucht. Wir berechnen Fehlergrenzen der Waveletapproximierung und zeigen wie eine beliebige Genauigkeit erreicht werden kann durch den Gebrauch von längeren Filtern. Abschliessend stellen wir Beispiele vor für die Implementierung von realen symmetrischen und anti-symmetrischen Wavelets. © 1997 Published by Elsevier Science B.V.

Résumé

Nous introduisons un cadre général pour le calcul efficient de la transformée en ondelettes continue (TOC) à l'aide d'un banc de filtres. La méthode permet un échantillonage arbitraire le long de l'axe des échelles, et présente une complexité en

* Corresponding author. Tel.: (301)435-1950; fax: (301) 496-6608; e-mail: vrhel@helix.nih.gov.

PII S0165-1684(96)00189-2

O(N) par échelle, où N est la longueur du signal. Les algorithmes antérieurs qui calculaient des échantillons non-dyadiques le long de l'axe des échelles nécessitaient O(Nlog(N)) calculs par échelle. Notre approche tire parti d'une fonction d'échelle compacte pour analyser l'ondelette. Nous discutons notre théorie, qui utilise le principe de dualité et le filtrage numérique récursif pour le calcul rapide de la TOC. Nous dérivons des limites d'erreur sur l'approximation d'ondelette et montrons comment obtenir n'importe quel degré de precision en utilisant des filtres plus longs. Enfin, nous présentons des exemples d'implantation pour des ondelettes réelles symétriques et antisymétriques. © 1997 Published by Elsevier Science B.V.

Keywords: Wavelet transform; Approximation theory; B-splines

1. Introduction

The continuous wavelet transform (CWT) has received significant attention in its capacity 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 signals [1]. We define the real CWT of a real continuous signal $s(t)$ as

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

where α and τ are, respectively, the continuously varying scaling and shifting parameters, and the real function $\psi(t)$ is the mother wavelet.[1] Clearly, the CWT is a redundant representation of the signal $s(t)$.

In practice, the variables α and τ are sampled over the plane of values. Fast algorithms exist for computing wavelet transforms at dyadic scale values $\alpha = 2^i$, when the wavelet is associated with a multi-resolution [2, 3]. In particular, if the wavelet is derived from a multi-resolution analysis [2], 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(N log(N)) computations per scale [8, 9].

In this paper, we introduce a fast method for computing general real CWTs at the sample values $\alpha = \alpha_0 2^{i/Q}$, $\tau = k$, 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 [10, 11]. What distinguishes our method from those previous approaches is that we achieve O(N) complexity per scale, instead of the O(N log(N)) reported in [10]. A drawback with the approach described in [10] is that the sequences of coefficients used for filtering are usually infinite, even if the wavelet and scaling functions are compactly supported. Our new method avoids this infinite filter sequence by use of a dual representation of a compactly supported scaling function, which is used to approximate the analyzing wavelet. Another feature is that we have full control over the approximation error, and that we can achieve any desired level of accuracy. We show that the quality of the approximation can be improved by using a higher-order scaling function or by adjusting the size of the finer scale wavelet. In either case, reducing the error may result in longer FIR filters. Low errors are obtained with moderately short filters. For example, to approximate the second derivative Gaussian wavelet using cubic splines, a symmetrical filter length of 17 is needed to achieve an rms

[1] To simplify the notation throughout the paper, we use a definition of the wavelet transform that is a time-reversed version of the conventional one.

error of 0.01. Our approach is a least-squares formulation since the approximation at the finer scales is the orthogonal projection of the wavelet onto a space defined by some scaling function (e.g. cubic B-spline). Finally, although we consider the real case in this paper, this method, as well as most others, is applicable to the complex CWT.

The organization of the paper is as follows. In Section 2, we specify the properties that the scaling function must satisfy to provide an admissible multi-scale approximation of the wavelet. In addition, we discuss the algorithm itself in the general framework of compactly supported scaling functions and wavelets. In Section 3, we provide error bounds on the wavelet approximation and introduce two mechanisms to control the approximation error in the algorithm. Finally, in Section 4, we discuss implementation and present examples for several wavelets using B-spline scaling functions of various degrees.

2. Theory of the fast algorithm

In this section, we formulate the fast wavelet transform algorithm. We discuss the general properties that are required for implementation and present the mathematics of the algorithm. The only requirement at this stage is that both the scaling function and wavelet are compactly supported. Specific examples will be presented in Section 4.

2.1. Scaling function requirements

The problem is to compute the values $W_\psi s(\alpha, \tau)$, which can be expressed in terms of the convolution

$$W_\psi s(\alpha, \tau) = (s * \psi_\alpha)(\tau), \tag{1}$$

where

$$\psi_\alpha(t) = \frac{1}{\sqrt{\alpha}}\psi\left(\frac{t}{\alpha}\right). \tag{2}$$

Direct computation of (1) would involve $O(N^2)$ operations per scale, while an FFT-based method would require $O(N \log(N))$ operations per scale. To achieve $O(N)$ complexity per scale, our approach is to approximate the wavelet with its orthogonal projection onto a subspace defined by a compactly supported scaling function. To insure that the projected wavelet is admissible, and to allow rapid calculation, the compactly supported scaling function φ must satisfy the following three conditions:

(i) $0 < A \leqslant \sum_{k\in Z}|\hat{\varphi}(\omega + 2\pi k)|^2 \leqslant B$; 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

$$V_\varphi = \left\{h(x) = \sum_{k\in Z} c(k)\varphi(x-k), c \in l_2\right\}, \tag{3}$$

and that V_φ is a well-defined (closed) subspace of L_2 [12]. This insures that the process of approximating a wavelet in V_φ makes sense. Property (ii) guarantees that the orthogonal projection of an admissible wavelet ψ onto the subspace defined by φ is an admissible wavelet as well (cf. Proposition 1 below). Property (iii) is usually referred to as the two-scale relation, and its importance in the algorithm is discussed in Section 2.2.

A necessary condition[2] for a function ψ to be an admissible wavelet is

$$\int_{-\infty}^{+\infty} \psi(t)\,\mathrm{d}t = 0 \tag{4}$$

Before proving that the projected wavelet satisfies this condition, it is necessary to introduce the dual of the scaling function φ, which is the unique function $\mathring{\varphi} \in V_\varphi$ for which

$$\langle \mathring{\varphi}_k, \varphi_l \rangle = \delta[k-l], \tag{5}$$

where $\delta[k]$ is the discrete unit impulse at the origin, $\varphi_k = \varphi(x-k)$, and $\langle\ \rangle$ denotes the L_2 inner product.

The least-squares approximation of $\psi(t/\alpha) \in L_2$ in the subspace V_φ will be denoted by a tilde and is

[2] In fact, this condition is sufficient for all practical purposes, cf. [1, p. 26].

given by the following two equations:

$$\tilde{\psi}_\alpha(t) = \begin{cases} \sum\limits_{k\in Z} p_\alpha(k)\varphi(t-k) & \\ \quad \text{where } p_\alpha(k) = \langle \psi_\alpha(t), \mathring{\varphi}(t-k)\rangle, & (6) \\ \sum\limits_{k\in Z} q_\alpha(k)\mathring{\varphi}(t-k) & \\ \quad \text{where } q_\alpha(k) = \langle \psi_\alpha(t), \varphi(t-k)\rangle. & (7) \end{cases}$$

Now let us consider the admissibility of the projection of ψ onto the subspace V_φ.

Proposition 1. *If $\psi \in L_2$ is an admissible wavelet and the scaling function φ satisfies properties* (i) *and* (ii), *then the orthogonal projection $\tilde{\psi} \in V_\varphi$ satisfies the admissibility condition* (4).

Proof. We first note the relationship

$$\int_{-\infty}^{+\infty} \psi(t)\,\mathrm{d}t = 0 \Leftrightarrow \hat{\psi}(0) = 0$$

and the following Fourier transform pair:

$$\sum_{i\in Z} \varphi(t-i) = 1 \Leftrightarrow \hat{\varphi}(2\pi k) = \delta[k], \quad k \in Z. \qquad (8)$$

Taking the Fourier transform of (6), the orthogonal projection of ψ onto V_φ can be expressed as [12]

$$\hat{\tilde{\psi}}(\omega) = p_1(\omega)\hat{\varphi}(\omega), \qquad (9)$$

where

$$p_1(\omega) = \sum_{k\in Z} \hat{\psi}(\omega + 2\pi k)\frac{\hat{\varphi}(\omega + 2\pi k)}{\sum_{l\in Z} |\hat{\varphi}(\omega + 2\pi l)|^2}.$$

Evaluating (9) at $\omega = 0$ and using (8) produces

$$\hat{\tilde{\psi}}(0) = \sum_{k\in Z} \hat{\psi}(2\pi k)\frac{\delta[k]}{\sum_{l\in Z} |\hat{\varphi}(2\pi l)|^2}\delta[0] = \frac{\hat{\psi}(0)}{\sum_{l\in Z} |\hat{\varphi}(2\pi l)|^2} = 0 \qquad \square$$

2.2. Mathematics of the algorithm

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

$$\tilde{W}_\psi s(\alpha, \tau) = (s * \tilde{\psi}_\alpha)(\tau), \qquad (10)$$

where $\tilde{\psi}_\alpha$ is the orthogonal projection of $\psi_\alpha(t) = \alpha^{-1/2}\psi(t/\alpha)$ onto the subspace V_φ. Although $\tilde{\psi}_\alpha$ is an approximation of ψ_α, we can control the quality of the approximation, as will be shown in Section 3.

Given that we have a compact scaling function φ that satisfies properties (i), (ii) and a compactly supported wavelet ψ, the function $\tilde{\psi}_\alpha$ can be completely characterized by a finite sequence of coefficients in the dual expansion (7) which is expressed as

$$\tilde{\psi}_\alpha(t) = (q_\alpha \mathbin{\dot{*}} \mathring{\varphi})(t) = \sum_k q_\alpha(k)\mathring{\varphi}(x-k), \qquad (11)$$

where the symbol $\dot{*}$ denotes the mixed convolution operation.

The motivation for using the dual representation instead of the usual projection formula (cf. Eq. (6)) $\tilde{\psi}_\alpha(t) = (p_\alpha \mathbin{\dot{*}} \varphi)(t)$ is that the sequence $p_\alpha(k)$ is usually infinite, even when ψ and φ 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.

Substituting the approximation $\tilde{\psi}_\alpha$ (cf. Eq. (11)) into Eq. (10) produces

$$\tilde{W}_\psi s(\alpha, \tau) = q_\alpha \mathbin{\dot{*}} (s * \mathring{\varphi})(\tau). \qquad (12)$$

From the definition (cf. Eq. (5)) and the projection formula, the dual of the compact scaling function can be expressed as

$$\mathring{\varphi}(t) = \left((a)^{-1} \mathbin{\dot{*}} \varphi\right)(t), \qquad (13)$$

where $(a^{-1})(l)$ is the convolution inverse of the finite sequence $a(l) = \langle \varphi, \varphi_l \rangle$, i.e. $\left((a)^{-1} * a\right)(l) = \delta[l]$. Substituting Eq. (13) into Eq. (12), we get

$$\tilde{W}_\psi s(\alpha, \tau) = \left((q_\alpha * (a)^{-1} \mathbin{\dot{*}} s_0)(\tau), \qquad (14)$$

where s_0 is

$$s_0(t) = (s * \varphi)(t). \qquad (15)$$

In practice, we will only evaluate $\tilde{W}_\psi s(\alpha, \tau)$ at the integers. It is therefore necessary to calculate the samples $s_0[k] = s_0(t)|_{t=k}$ from the signal values

$s[k] = s(t)|_{t=k}$. To compute the continuous convolution in Eq. (15), it is essential to know the signal $s(t)$ in a continuous fashion. Such a continuous representation can be specified by interpolating the sample values with an interpolating kernel $\varphi_{\mathrm{I}}(t)$ such that $s(t) = \sum_{l \in Z} s[l]\varphi_{\mathrm{I}}(t-l) = (s * \varphi_{\mathrm{I}})(t)$ [12]. One possibility is to choose the interpolation kernel for the space V_φ (cf. Eq. (3)) [13]. Using the interpolator $\varphi_{\mathrm{I}}(t)$, computation of $s_0[k]$ from $s[k]$ is performed using

$$s_0[k] = \langle s(x), \varphi_k \rangle = \langle s * \varphi_{\mathrm{I}}, \varphi_k \rangle = (a_{\mathrm{I}} * s)(k), \quad (16)$$

where $a_{\mathrm{I}}(l-k) = \langle \varphi_{\mathrm{I}}(x-l), \varphi(x-k) \rangle$.

Eq. (16) is the general initialization equation for the algorithm. An alternative to using Eq. (16) is to calculate a discrete approximation of the continuous convolution given in Eq. (15). In such an approach, the samples $s_0[k]$ are approximated as

$$s_0[k] \cong (s * b)(k), \quad (17)$$

where $b(k) = \varphi(t)|_{t=k}$.

Sampling Eq. (14) at $\tau = k$ produces $\tilde{W}_\psi s(\alpha, k) = (q_\alpha * (a)^{-1} * s_0)(k)$, which involves filtering of the sequence $s_0(k)$ with an FIR filter $q_\alpha(k)$ and an IIR filter $(a)^{-1}(k)$, which can be implemented recursively. Details of implementation are discussed in Section 4.

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

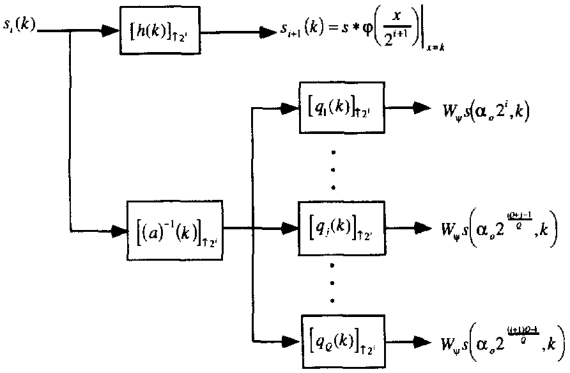
$$\alpha_j = \alpha_0 2^{j/Q}, \quad j = 0, \dots, Q-1, \quad (18)$$

and then use property (iii) of the scaling function to compute the CWT for Q scales in each of the higher octaves. The samples in the ith octave can be efficiently calculated from the values

$$s_i(k) = \left(s * \varphi\left(\frac{x}{2^i}\right)\right)\bigg|_{x=k}.$$

Given the FIR filters

$$q_j(k) = \langle \psi_{\alpha_j}, \varphi_k \rangle, \quad j = 0, \dots, Q-1, \quad (19)$$



$s_i(k)$
$[h(k)]_{\uparrow 2^i}$
$s_{i+1}(k) = s * \varphi\left(\frac{x}{2^{i+1}}\right)\Big|_{x=k}$
$[q_1(k)]_{\uparrow 2^i}$
$W_\psi s(\alpha_o 2^i, k)$
$[(a)^{-1}(k)]_{\uparrow 2^i}$
$[q_j(k)]_{\uparrow 2^i}$
$W_\psi s\left(\alpha_o 2^{\frac{iQ+j-1}{Q}}, k\right)$
$[q_Q(k)]_{\uparrow 2^i}$
$W_\psi s\left(\alpha_o 2^{\frac{(i+1)Q-1}{Q}}, k\right)$

Fig. 1. Schematic representation of the general algorithm.

the samples $s_i(k)$, and the IIR filter $(a)^{-1}(k)$, the values for the ith octave are

$$\tilde{W}_\psi s(\alpha_0 2^{(iQ+j/Q)}, k) = ([q_j * (a)^{-1}]_{\uparrow 2^i} * s_i)(k),$$
$$j = 0, \dots, Q-1. \quad (20)$$

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 $s_{i+1}(k)$ can be efficiently computed from $s_i(k)$ using

$$s_{i+1}(k) = (s_i * [h]_{\uparrow 2^i})(k). \quad (21)$$

Details of the above equations are given in Appendix A. A diagram of the system is shown in Fig. 1.

3. The approximation error

The above approach provides an approximation to the wavelet transform, the quality of which depends upon the wavelet and scaling functions. There are essentially two ways to control the error: change the scaling function, or adjust the size of the finer scale α_0 (cf. Eq. (18)). In either case, reducing the error may result in longer FIR filters.

In general, the approximation power of a scaling function φ will depend on its ability to reproduce polynomials up to a certain degree n. This maximum degree plus one gives the order of accuracy of the representation. It is also directly related to the multiplicity of the zeros of $\hat{\varphi}(\omega)$ (the Fourier transform of φ) at all frequencies $\omega = 2\pi k$, where k is a non-zero integer. The precise formulation of these mathematical properties is provided by the Strang–Fix conditions, which are discussed in Appendix B. These conditions imply that we can control the error by adjusting the sampling step h. In our situation, where the sampling step is fixed, this is equivalent to dilating the wavelet by a factor $\alpha = h^{-1}$. For a scaling function of order L, the general error bound (B.6) can be rewritten in the following equivalent form:

$$\varepsilon = \| \psi_\alpha - \tilde{\psi}_\alpha \| \leqslant \frac{C_\varphi}{\alpha^L} \| \psi^{(L)} \| = \frac{C_\varphi \cdot C_\psi}{\alpha^L}, \qquad (22)$$

where $\psi^{(L)}$ is the Lth derivative of ψ, and $\tilde{\psi}_\alpha$ is the approximation of ψ_α in V_φ; note that the factor $\alpha^{-1/2}$ in the definition of ψ_α (cf. Eq. (2)) provides the correct inner product normalization for the dilation of the wavelet and its approximation. The numerator on the right-hand side of (22) is the product of two constants: a first term C_φ, which is a function of the representation, and a second wavelet-dependent term $C_\psi = \| \psi^{(L)} \|$, which can be pre-computed by integrating in the time or frequency domain:

$$C_\psi = \sqrt{\int_{-\infty}^{+\infty} |\psi^{(L)}(x)|^2 \, \mathrm{d}x} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{+\infty} |\omega^L \hat{\psi}(\omega)|^2 \, \mathrm{d}\omega}.$$

An implicit assumption is that the L first derivatives of ψ are defined in the L_2-sense. Eq. (22) indicates that the approximation error decreases with the Lth power of the scale. Clearly, the error will be maximum at the finer scale α_0. Our design strategy is therefore to select the parameters α_0 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 (B.9). Specifically, we can rewrite this equation as

$$\| \psi_\alpha - \tilde{\psi}_\alpha \| = \frac{C_2 \cdot C_\psi}{\alpha^L} \quad \text{as } \alpha \to +\infty, \qquad (23)$$

where the constant C_2 is given by (B.10). 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, then $\hat{\varphi}(\omega) = \mathrm{sinc}^{n+1}(\omega/2\pi)$, and it is possible to compute the constant C_2 explicitly:

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

where $|B_{2n}|$ is the modulus of Bernoulli's number of degree $2n$ (cf. [14, Eqs. (23.1.2) and (23.1.18)]),

$$|B_{2n}| = 2(2n)! \sum_{k=1}^{+\infty} (2\pi k)^{-2n}.$$

This yields the following numerical value: $C_2 = 3.72678 \times 10^{-2}$ for $n = 1$, $C_2 = 9.09241 \times 10^{-4}$ for $n = 3$, and $C_2 = 2.29874 \times 10^{-5}$ for $n = 5$.

This general behavior of the error as a function of the scale was verified experimentally for several examples of wavelets. Fig. 2 provides the graph of the approximation error for an anti-symmetric wavelet (first derivative of a Gaussian) (cf. Fig. 5) obtained with our least-squares method using piece wise linear ($n = 1$), and cubic splines ($n = 3$). The two curves exhibit the characteristic $O(1/\alpha^L)$ behavior predicted by the theory. The asymptotic curves given by (23) are also represented and are in perfect agreement with the experimental results as α becomes sufficiently large. Note that the curve reaches its asymptotic regime much quicker when a lower-order approximation is used. Fig. 3 shows a similar graph for the approximation of a Mexican hat wavelet (second derivative of a Gaussian) (cf. Fig. 6). In either case, the improvement that can be achieved at a given scale by switching to a higher-order representation is quite substantial.

For our experiments, we selected an error threshold of $\varepsilon = 0.01$ and choose to determine the corresponding finer scale parameter α_0 by solving (23) as a function of α. An alternative approach, which may be more appropriate for higher-order splines, 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 $C \geqslant C_2 \cdot C_\psi$ (cf. Eq. (B.6) in Appendix B). In our system, this error analysis is only applicable to the first octave, because the higher octaves all use the same wavelet approximations dilated by a power of two. In fact, the error is exactly the same for all octaves because the wavelets have a normalized energy.

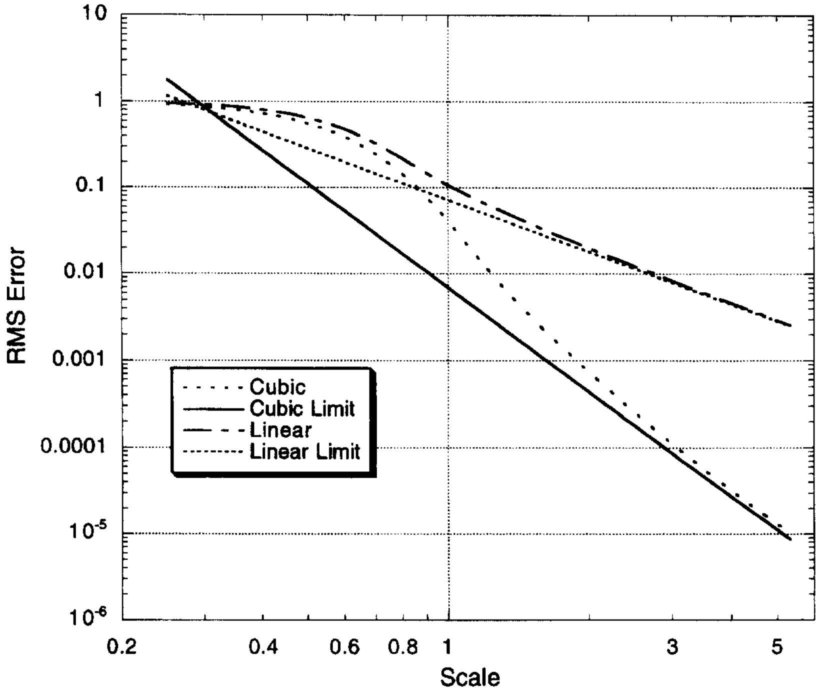


Fig. 2. Approximation error for anti-symmetric wavelet.

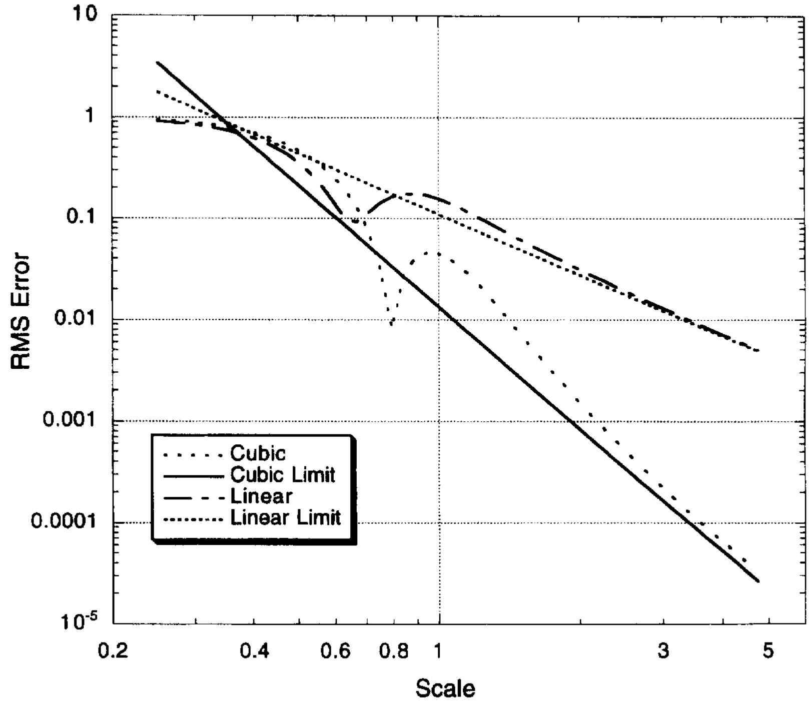


Fig. 3. Approximation error for Mexican hat wavelet.

4. Implementation of the algorithm

In this section, we look at implementing the algorithm for various wavelets, using centered B-spline scaling functions to approximate the wavelet. The B-spline functions are compact and satisfy the three properties given in Section 2. In addition, they are flexible enough to produce a good approximation of almost any wavelet shape. A B-spline function of degree n, represented by $\beta^n(x)$, is simply the $(n+1)$-fold convolution of the unit pulse function. The integer sample values of the nth degree B-spline are denoted by $b^n(k) = \beta^n(x)|_{x=k}$.

4.1. *The filter coefficients*

Once the wavelet has been selected, an approximation error level must be specified as discussed in Section 3. This will set the degree of the B-spline and the fine scale α_0. When these parameters are known, the FIR filters

$$q_j(k) = \langle \psi_{\alpha_j}, \beta_k^n \rangle, \quad j = 0, \dots, Q-1, \tag{24}$$

can be calculated (cf. Eq. (19)). These filters are computed by numerical integration methods. Again the length of the FIR filters depends upon the degree of the B-spline, the fine scale α_0, and the support of the wavelet.

With a B-spline implementation, property (iii) in Section 2 is given by $\beta^n(x/2) = \sum_{k\in Z} u_2^n(k)\beta^n(x-k)$ where the $u_2^n(k)$ are the binomial filters [15]

$$u_2^n(k) = \begin{cases} \dfrac{1}{2^n}\dbinom{n+1}{k+(n+1)/2}, & |k| \leqslant (n+1)/2, \\ 0, & \text{otherwise.} \end{cases}$$

The remaining filter in the algorithm is the IIR filter $(a)^{-1} = (b^{2n+1})^{-1}$. Note that the algorithm involves the enlarged filter $[(b^{2n+1})^{-1}]_{\uparrow 2^i}$, which will require the execution of an IIR 'à trous' filter. Fast implementation of this IIR 'à trous' filter and the FIR 'à trous' filters are discussed in the next section.

Table 1
Algorithm filter coefficients and parameters

Spline order n_1	FIR filter b^{n_1} coefficients	IIR filter $(a)^{-1}$ parameters	
		d_a	$\lvert z_i\rvert < 1$
1	1	1	—
3	1/6 (•, 4, 1)	6	$z_1 = -2+\sqrt{3}$
7	1/5040 (•, 2416, 1191, 120,1)	5040	$z_1 = -0.53528$ $z_2 = -0.122555$ $z_3 = -0.00914869$

Note: The notation • indicates that the filter is symmetric.

4.2. *Fast B-spline implementation*

(i) *Initialization.* Initialization of the algorithm involves the one time calculation of $s_0(k)$. Using a spline of degree n_1 in approximating the wavelet, and a spline of degree n_2 in interpolating $s[k]$, the computation in producing an approximation to $s_0(k)$ is (cf. Eq. (17))

$$s_0(k) \cong b^{n_1} * s[k], \tag{25}$$

where b^{n_1} is a symmetric FIR filter of length n_1+1 (sampled B-spline of degree n_1). Coefficients for the filter are given in Table 1 for various spline degrees.

If the exact initialization is performed, then the computation is (cf. Eq. (16))

$$s_0(k) = b^{n_1+n_2+1} * (b^{n_2})^{-1} * s[k],$$

where the filter $(b^{n_2})^{-1}$ transforms the sample values from the cardinal spline representation to the basic spline representation [13]. As in the approximation (25), $b^{n_1+n_2+1}$ is a symmetric FIR filter whose coefficients are given in Table 1 for several degrees.

(ii) *The FIR 'à trous' filters.* The algorithm requires implementation of the FIR filters $[u_2^{n_1}]_{\uparrow 2^i}$ and $[q_\alpha]_{\uparrow 2^i}$. The filter $[u_2^{n_1}]_{\uparrow 2^i}$ can be decomposed into a cascade of filters given by

$$[u_2^{n_1}(k)]_{\uparrow 2^i} = \frac{1}{2^{n_1}} \underbrace{[u_2^0]_{\uparrow 2^i} * [u_2^0]_{\uparrow 2^i} * \cdots * [u_2^0]_{\uparrow 2^i}(k-k_0)}_{n_i+1 \text{ times}},$$

where the shift $k_0 = (i+1)(n_1+1)/2$ is due to the definition of $u_2^0(k)$ which is $u_2^0(k) = \delta[k] + \delta[k-1]$. Each filter $[u_2^0]_{\uparrow 2^i}$ consists of only one addition, which means

Table 2
Algorithm filter coefficients for cubic B-spline approximation of the Mexican hat wavelet. Since the filters are symmetric only half the coefficients are shown

(a)

q_0	q_1	q_2	q_3	q_4	q_5
7.86839E − 1	8.06546E − 1	8.24748E − 1	8.41499E − 1	8.56865E − 1	8.70918E − 1
3.58644E − 1	4.02186E − 1	4.44946E − 1	4.86533E − 1	5.26626E − 1	5.64971E − 1
− 2.43728E − 1	− 2.10886E − 1	− 1.71461E − 1	− 1.26294E − 1	− 7.63618E − 1	− 2.27152E − 2
− 3.25052E − 1	− 3.49672E − 1	− 3.6693E − 1	− 3.75476E − 1	− 3.74407E − 1	− 3.63305E − 1
− 1.44864E − 1	− 1.83494E − 1	− 2.24816E − 1	− 2.66787E − 1	− 3.07035E − 1	− 3.43085E − 1
− 3.35908E − 2	− 5.17306E − 2	− 7.59024E − 2	− 1.06397E − 1	− 1.42863E − 1	− 1.84213E − 1
− 4.45724E − 3	− 8.70529E − 3	− 1.58655E − 2	− 2.71156E − 2	− 4.36474E − 2	− 6.64414E − 2
− 3.36269E − 4	− 8.9972E − 4	− 2.13945E − 3	− 4.62615E − 3	− 9.21115E − 3	− 1.70128E − 2
− 8.19113E − 6	− 4.50248E − 5	− 1.73427E − 4	− 5.28104E − 4	− 1.37056E − 3	− 3.17197E − 3
	− 1.4759E − 7	− 3.58314E − 6	− 2.78422E − 5	− 1.27731E − 4	− 4.24214E − 4
			4.98443E − 8	− 2.71909E − 6	− 2.66356E − 5
					− 6.22115E − 8

(b)

q_6	q_7	q_8	q_9	q_{10}	q_{11}
8.83735E − 1	8.95398E − 1	9.05985E − 1	9.15579E − 1	9.24256E − 1	9.32091E − 1
6.01381E − 1	6.35728E − 1	6.67938E − 1	6.97985E − 1	7.25879E − 1	7.51663E − 1
3.35801E − 2	9.14932E − 2	1.50067E − 1	2.08444E − 1	2.65886E − 1	3.21777E − 1
− 3.42239E − 1	− 3.11713E − 1	− 2.72587E − 1	− 2.25989E − 1	− 1.73211E − 1	− 1.15622E − 1
− 3.7259E − 1	− 3.93546E − 1	− 4.04443E − 1	− 4.04361E − 1	− 3.92987E − 1	− 3.70576E − 1
− 2.28653E − 1	− 2.73816E − 1	− 3.16997E − 1	− 3.55427E − 1	− 3.8655E − 1	− 4.08251E − 1
− 9.60139E − 2	− 1.32198E − 1	− 1.74017E − 1	− 2.1969E − 1	− 2.66774E − 1	− 3.12411E − 1
− 2.93152E − 2	− 4.7375E − 2	− 7.21576E − 2	− 1.04061E − 1	− 1.42679E − 1	− 1.868E − 1
− 6.68919E − 3	− 1.2981E − 2	− 2.33402E − 2	− 3.91265E − 2	− 6.15053E − 2	− 9.11483E − 2
− 1.15417E − 3	− 2.76264E − 3	− 5.98962E − 3	− 1.18915E − 2	− 2.17834E − 2	− 3.70742E − 2
− 1.32958E − 4	− 4.50669E − 4	− 1.23004E − 3	− 2.9526E − 3	− 6.41176E − 3	− 1.27234E − 2
− 3.90106E − 6	− 3.88602E − 5	− 1.85645E − 4	− 5.97465E − 4	− 1.57948E − 3	− 3.71234E − 3
	− 2.65512E − 7	− 9.60464E − 6	− 7.76508E − 5	− 3.15764E − 4	− 9.24475E − 4
		− 6.10238E − 10	− 1.84295E − 6	− 3.20561E − 5	− 1.79798E − 4
				− 2.23123E − 7	− 1.29972E − 5
					− 8.47032E − 9

that the binomial filter $[u_2^{n_1}(k)]_{\uparrow 2^i}$ can be implemented with $n_1 + 1$ additions per sample. Alternatively, $[u_2^{n_1}(k)]_{\uparrow 2^i}$ can be implemented in the 'à trous' fashion with $(n_1 + 3)/2$ multiplications and $n_1 + 1$ additions per sample.

The filter $[q_\alpha]_{\uparrow 2^i}$ is defined by a vector $\boldsymbol{q}$ of length n_q (cf. Table 2). If the wavelet is symmetric or anti-symmetric with n_q odd, then this filter requires $(n_q + 1)/2$ multiplications and $n_q - 1$ additions per sample.

(iii) *The IIR recursive 'à trous' filter.* The filter $[(a)^{-1}(k)]_{\uparrow m}$ can be implemented as a recursive filter. Since $(a)^{-1}$ is a symmetrical all pole filter, the z-transform of its up-sampled version $[(a)^{-1}(k)]_{\uparrow m}$ can be written in the standard form

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

where d_0 and $\{c_k, k = 0, \dots, n_1 - 1\}$ are constant coefficients. The filter is then expressed as a cascade of simple first-order causal/anti-causal components

$$A_{n_1}(z^m) = d_0 \prod_{i=1}^{n_1} A(z^m; z_i),$$

where $A(z^m; z_i)$ is defined as

$$A(z^m; z_i) = \frac{-z_i}{(1 - z_i z^{-m})(1 - z_i z^m)} = \left(\frac{1}{1 - z_i z^{-m}}\right)\left(\frac{-z_i}{1 - z_i z^m}\right). \tag{26}$$

Values of d_0 and $\{z_i, i = 1, \dots, n_i\}$ for different spline orders are given in Table 1.

Assuming that the input signal is $\{x(k)\}_{k=0,\dots,N-1}$, the right-hand side of (26) yields the following recursive filter equations:

$$y^+(k) = x(k) + z_i y^+(k - m),$$

$$k = m, \dots, N - 1, \tag{27}$$

$$y(k) = -z_i y^+(k) + z_i y(k + m)$$

$$= z_i(y(k + m) - y^+(k)),$$

$$k = N - 1 - m, \dots, 0. \tag{28}$$

Fig. 4 shows the realization of the filter. In order to calculate $y^+(k)$ recursively using Eq. (27), we need to know $y^+(k)$ for $k = 0, \dots, m - 1$. These initial values are computed using

$$y^+(k) = (a^+ * x)(k) = \sum_{j \geq 0} x(k - mj) z_i^j = \sum_{j=0}^{k_0} x(k - mj) z_i^j$$

$$(k = 0, \dots, m - 1), \tag{29}$$

where k_0 is chosen to ensure that $z_i^{k_0}$ is smaller than some pre-specified level of precision. In our application, we set $k_0 = \log(10^{-8})/\log(|z_i|)$. Eq. (29) is typically computed by extending the signal by its mirror image, i.e., $x(-k) = x(k)$, $k = 0, \dots, N - 1$, and $x(k) = x(2N - 2 - k)$, $k = N, \dots, 2N - 1$.

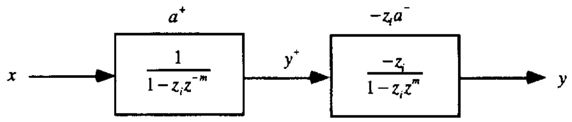


Fig. 4. Implementation of an elementary IIR filter module.

The recursive computation of (28) also requires $y(k)$ to be known for $k = N - 1 - m, \dots, N - 1$. In order to obtain these values, we rewrite Eq. (26) as

$$A(z^m; z_i) = \frac{-z_i}{(1 - z_i^2)}\left(\frac{1}{1 - z_i z^{-m}} + \frac{1}{1 - z_i z^m} - 1\right) = \frac{-z_i}{(1 - z_i^2)}(A^+(z^m) + A^-(z^m) - 1), \tag{30}$$

which suggests the following computation:

$$y(k) = \frac{-z_i}{(1 - z_i^2)}(y^+(k) + y^-(k) - 1). \tag{31}$$

Specifically, we first calculate $y^-(k)$ for $k = N - 1 - m, \dots, N - 1$, as in Eq. (29), and then use (31) to compute the m initial values for $y(k)$.

Hence, with the proposed method, this basic filtering operation requires $2N + 2k_0 m$ additions and $2N + (2k_0 + 1)m$ multiplications.

4.3. Results and discussion

We implemented the algorithm for the wavelets

$$\psi_{\mathrm{deriv}}(t) = \begin{cases} -K_0 \dfrac{t}{\alpha_0} \mathrm{e}^{-(t/\alpha_0)^2/2}, & \left|\dfrac{t}{\alpha_0}\right| \leqslant 5, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_{\mathrm{max}}(t) = \begin{cases} K_0\left(1 - \left(\dfrac{t}{\alpha_0}\right)^2\right) \mathrm{e}^{-(t/\alpha_0)^2/2} - K_1, & \left|\dfrac{t}{\alpha_0}\right| \leqslant 5, \\ 0, & \text{otherwise,} \end{cases}$$

where K_0 is a constant that insures that $\|\psi\| = 1$ and K_1 guarantees the admissibility of ψ_{mex} (cf. Eq. (4)). The values of α_0 were selected to achieve a worst-case error of $\varepsilon = 0.01$ as noted in Section 3. The value of α_0 is 2.69 for ψ_{deriv} and 3.32 for ψ_{mex} when using a B-spline of degree 1. For a

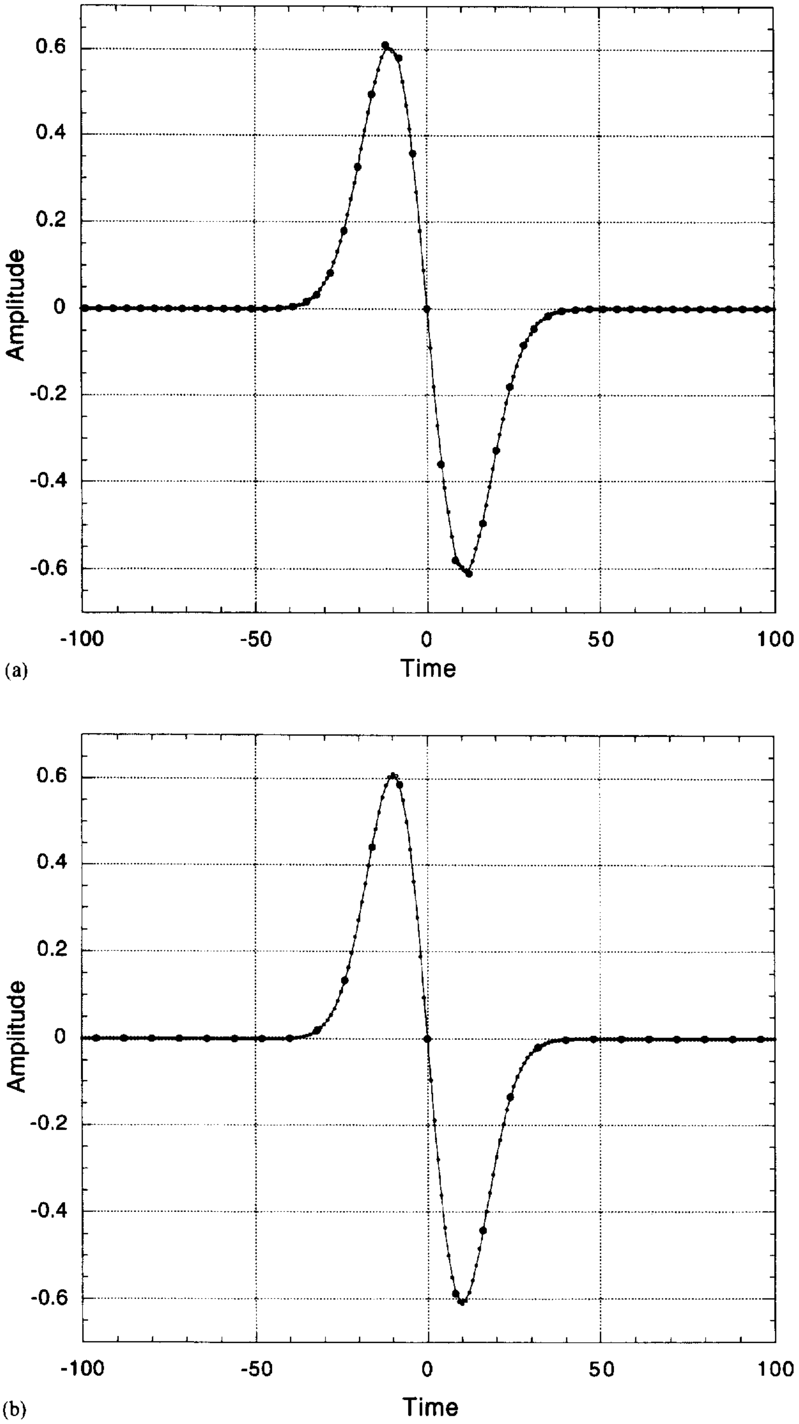


Fig. 5. (a) Anti-symmetric wavelet (line) and its linear least-squares approximation (small circles) at an rms error of 0.01. The large black circles represent the spline knots. (b) Anti-symmetric wavelet (line) and its cubic least-squares approximation (small circles) at an rms error of 0.01. The large black circles represent the spline knots.

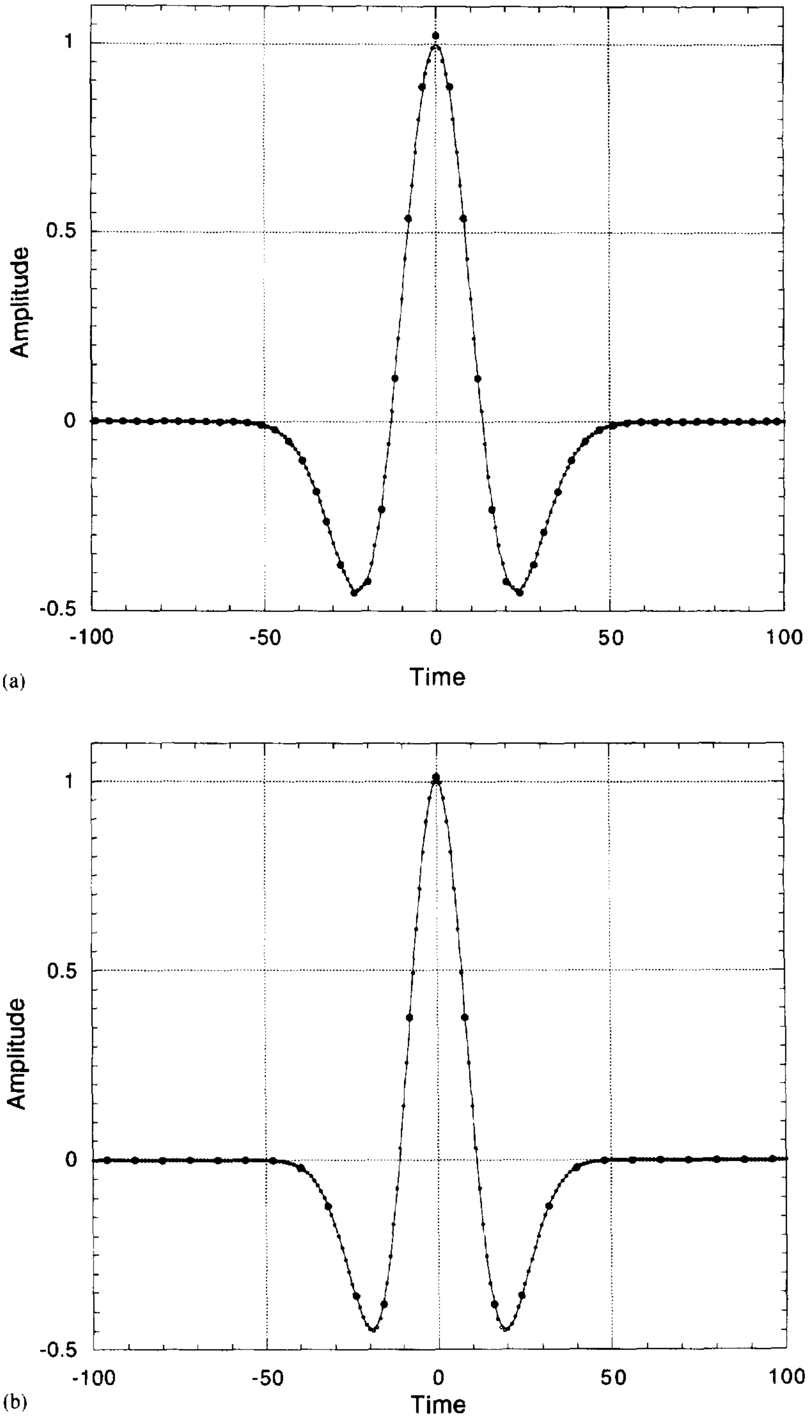


Fig. 6. (a) Mexican hat wavelet (line) and its linear least-squares approximation (small circles) at an rms error of 0.01. The large black circles represent the spline knots. (b) Mexican hat wavelet (line) and its cubic least-squares approximation (small circles) at an rms error of 0.01. The large black circles represent the spline knots.

B-spline of degree 3 the value of α_0 is 1.25 for ψ_{deriv} and 1.40 for ψ_{mex}. The wavelets are shown in Figs. 5(a), (b) and 6(a), (b) with their linear and cubic spline least-squares (LS) approximations. The wavelets and their approximations are virtually indistinguishable. We computed the filters q_j (cf. Eq. (24)) for the case of $Q = 12$. This provides a discretization of each octave that corresponds to the musical notes $(A, A^{\#}, B, C, C^{\#}, \ldots)$. The filter coefficients q_j are contained in Table 2 for the cubic B-spline implementation of the ψ_{mex} wavelet.

The initialization approximation was used, as given in Eq. (25) and the 2-scale filter $[u_{2^i}^{n}(k)]_{\uparrow 2^i}$ was implemented using the zero-padded algorithm. The impulse response of the system for each of the wavelets is shown in Figs. 7(a), (b) and 8(a), (b).

As implemented, the sample values provide redundant information about the signal. While this redundancy is useful in signal analysis, in some applications less information may be required, especially as the scale variable becomes large. To reduce the amount of data, it may be desirable to

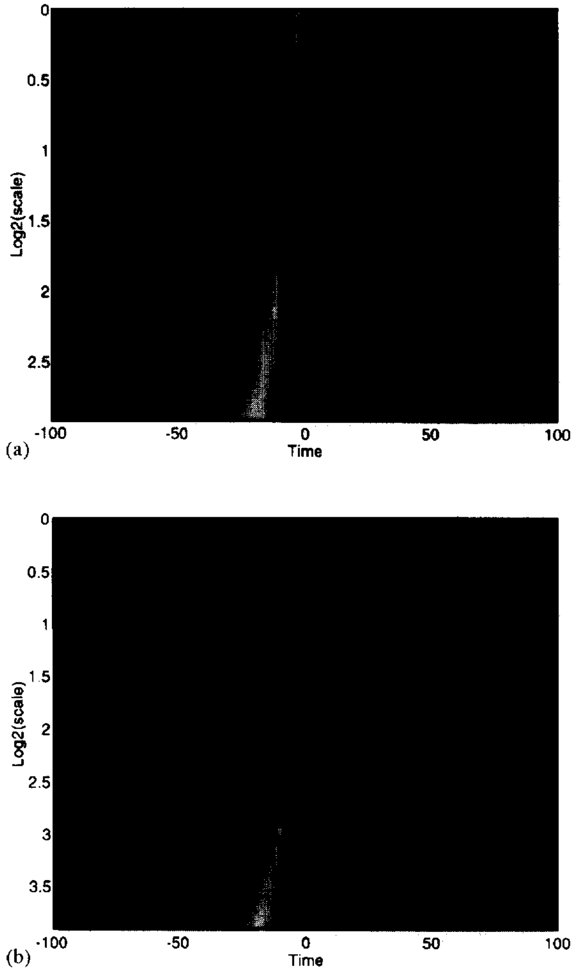


Fig. 7. Wavelet transform of unit impulse of the anti-symmetric wavelet for 12 voices per octave: (a) piecewise linear approximation; (b) cubic spline approximation.

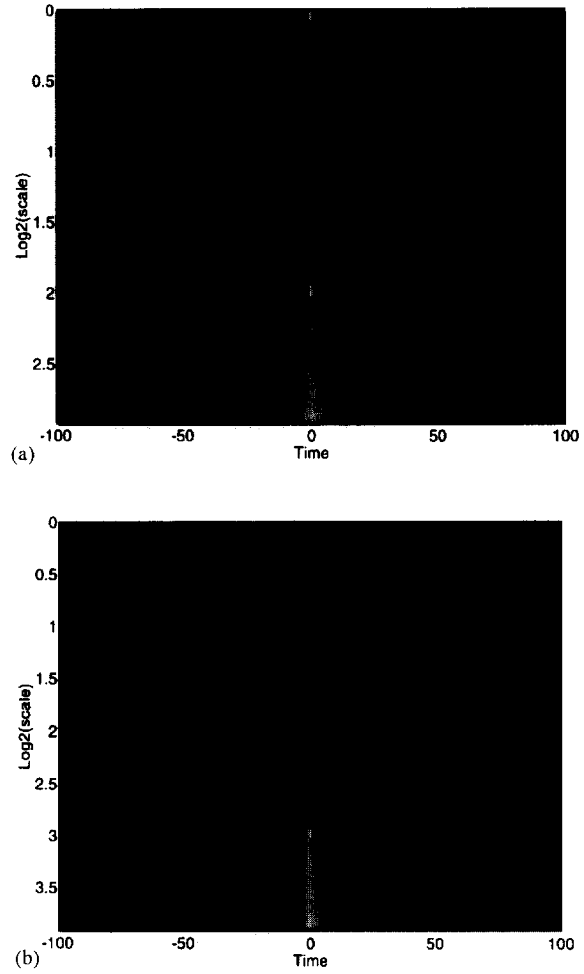


Fig. 8. Wavelet transform of unit impulse of the Mexican hat wavelet for 12 voices per octave: (a) piecewise linear approximation; (b) cubic spline approximation.

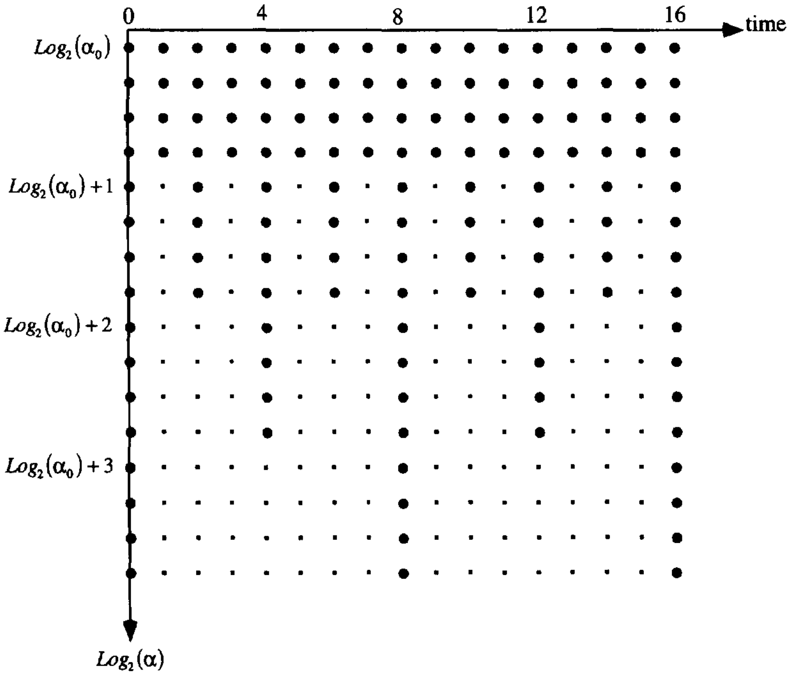


Fig. 9. Sample values of CWT for the system with dyadic subsampling at each octave.

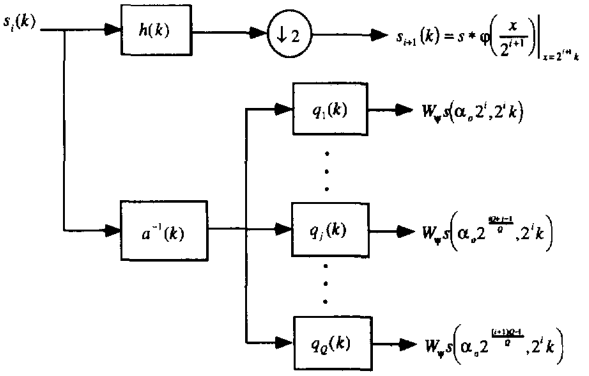

$s_i(k)$
$h(k)$
$\downarrow 2$
$s_{i+1}(k) = s * \varphi\left(\frac{x}{2^{i+1}}\right)\Big|_{x=2^{i+1}k}$
$a^{-1}(k)$
$q_1(k)$
$q_j(k)$
$q_Q(k)$
$W_\psi s(\alpha_o 2^i, 2^i k)$
$W_\psi s\left(\alpha_o 2^{\frac{iQ+j-1}{Q}}, 2^i k\right)$
$W_\psi s\left(\alpha_o 2^{\frac{(i+1)Q-1}{Q}}, 2^i k\right)$

Fig. 10. System which performs dyadic subsampling at each octave.

subsample at each scale. In this case, the sample values in the time scale plane are as displayed in Fig. 9. A system to implemented this algorithm is shown in Fig. 10. The system is simpler than the one in Fig. 1 which fills in the samples values denoted by the small points in Fig. 9.

5. Conclusion

We have described a general procedure for the fast evaluation of the continuous WT. The method that was presented has the following attractive features:

- The procedure can approximate any desired wavelet shape. The filter coefficients are determined by performing simple inner products.
- The number of operation per scale is $O(N)$, which is the lowest possible order of complexity.
- The method can provide arbitrary sampling along the scale axis. It is especially useful for computing more than one scale per octave.
- The approximation error can be easily controlled by either adjusting the fine scale resolution or by selecting a scaling function with a higher order of approximation. The behavior of the approximation error has been characterized asymptotically, and used in the design of the wavelet filters.

- The procedure as described provides the finest possible sampling along the time dimension. The system can be easily adapted for a less redundant representation by dyadic subsampling within each octave.

Acknowledgements

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

Appendix A. Two-scale relationship between octaves

In this appendix we derive the relationships given in Eqs. (20) and (21), which define the algorithm. We wish to compute the samples

$$\tilde{W}_{\psi}(\alpha_0 2^{i+1}, k) = \int_{-\infty}^{+\infty} s(t)\tilde{\psi}_{\alpha_0 2^{i+1}}(t-k)\,\mathrm{d}t. \tag{A.1}$$

From Eqs. (11) and (13) we have the following:

$$\tilde{\psi}_{\alpha_0 2^{i+1}}(t) = \tilde{\psi}_{\alpha_0}\left(\frac{t}{2^{i+1}}\right) = \sum_k p_\alpha(k)\varphi\left(\frac{t}{2^{i+1}} - k\right), \tag{A.2}$$

where $p_\alpha(k) = (q_{\alpha_0} * (a)^{-1}(k)$.

A simple algebraic manipulation of Eq. (A.2) yields $\tilde{\psi}_{\alpha_0 2^{i+1}}(t) = \sum_k [p_\alpha(k)]_{\uparrow 2^{i+1}}\varphi((t-k)/2^{i+1})$. Substituting into Eq. (A.1) produces $\tilde{W}_\psi(\alpha_0 2^{i+1}, k) = ([p_\alpha]_{\uparrow 2^{i+1}} * s_{i+1})(k)$, where

$$s_{i+1}(k) = \left(s * \varphi\left(\frac{t}{2^{i+1}}\right)\right)\bigg|_{t=k}, \tag{A.3}$$

which is equivalent to Eq. (20).

Now consider the computation of $s_{i+1}(k)$. From property (iii) we have

$$\varphi\left(\frac{t}{2^{i+1}}\right) = \sum_{k\in Z} h(k)\varphi\left(\frac{t}{2^i} - k\right),$$

which can be written as

$$\varphi\left(\frac{t}{2^{i+1}}\right) = ([h]_{\uparrow 2^i} \dot{*} \varphi_{2^i})(t), \tag{A.4}$$

where we have used the notation $\varphi_{2^i}(t) = \varphi(t/2^i)$. Substituting Eq. (A.4) into Eq. (A.3) leads to

$$s_{i+1}(k) = [h]_{\uparrow 2^i} \dot{*} (s * \varphi_{2^i})(t)|_{t=k} = ([h]_{\uparrow 2^i} * s_i)(k),$$

which is equivalent to Eq. (21).

Appendix B. Strang–Fix conditions and least squares approximation

In approximation theory, the standard way to control the error is to vary the step size (or sampling interval) h. The corresponding rescaled version of our approximation subspace is $V_h(\varphi) = \mathrm{span}\{\varphi(x/h - k)\}_{k\in Z}$ and the approximation of s at the scale h is denoted by s_h. As h gets smaller, the approximation error $\| s - s_h \|$ generally descreases and eventually becomes negligible as h goes to zero. It turns out that the general behavior of this error as a function of h depends on the ability of the functions $\{\varphi(x-k)\}_{k\in Z}$ to reproduce polynomials up to a certain degree n. This result is expressed by the Strang–Fix [16] conditions, which relate the approximation power of the representation to the spectral characteristics of the generating function φ. These conditions also play an important role in the theory of the wavelet transform [17].

Strang–Fix conditions [16]. The following four statements are equivalent:

(i) The function space $V(\varphi)$ reproduces all polynomials up to degree n. Specifically, there exists a function $\varphi_{\mathrm{int}} \in V(\varphi)$ (not necessarily unique) that interpolates all polynomials $p_n(x)$ of degree n:

$$\forall p_n(x) \in \pi^n, \quad \sum_{k\in Z} p_n(k)\varphi_{\mathrm{int}}(x-k) = p_n(x). \tag{B.1}$$

(ii) $\hat{\varphi}(\omega)$, the Fourier transform of φ, is non-vanishing at the origin and has zeros of at least multiplicity $(n+1)$ at all non-zero frequencies that are integer multiples of 2π.

(iii) There exists a function $\varphi_{\mathrm{int}} \in V(\varphi)$ (the same as in (B.1)) such that

$$\forall x \in R, \sum_{k\in Z} \varphi_{\mathrm{int}}(x-k) = 1, \tag{B.2}$$

$$\forall x \in R, \sum_{k\in Z} (x-k)^m \varphi_{\mathrm{int}}(x-k) = 0,$$

$$m = 1, \dots, n, \tag{B.3}$$

or equivalently,

$$\hat{\varphi}_{\text{int}}(2\pi k) = \delta[k], \tag{B.4}$$

$$\hat{\varphi}^{(m)}_{\text{int}}(2\pi k) = 0, \quad m = 1, \dots, n, \tag{B.5}$$

where $\hat{\varphi}^{(m)}_{\text{int}}$ denotes the mth derivative of the Fourier transform of φ_{int} with respect to ω.

(iv) The approximation error at step size h is bounded as

$$\forall s \in W^{(n+1)}, \inf_{s_h \in V_h} \| s - s_h \| \leqslant C h^{n+1} \| s^{(n+1)} \|, \tag{B.6}$$

where $W^{(n+1)}$ is Sobolev's space of order $(n+1)$, i.e., the space of smooth functions whose $(n+1)$ first derivatives are defined in the L_2-sense (bounded energy).

Remarks and comments

(a) By definition, the order of accuracy (or approximation power) of the representation is $L = n + 1$ where n is the maximum degree for which any of these conditions is satisfied. For instance, polynomial splines of degree n have an order of accuracy $L = n + 1$.

(b) Functions satisfying (B.1) are called quasi-interpolants. A particular example is the fundamental function (interpolator) which is one at the origin and zero at all other integers.

(c) Eqs. (B.2), (B.3) and (B.4), (B.5) form a discrete Fourier transform pair since we are dealing with periodized signals. The implication of (B.4) and (B.5) is that the transfer function of a quasi-interpolant is flat at the origin.

For our particular application, we would like to get a better handle on the constant C in (B.6). For this purpose, we can examine the behavior of the error pointwise and determine the asymptotic behavior of $\| s - s_h \|$ as h becomes sufficiently small. The main steps of this derivation are as follows; for more details, we refer to [18].

We start by writing the least-squares approximation of the function $s \in L_2$ in V_h as

$$s_h(x) = \sum_{k \in Z} \left\langle s(y), h^{-1} \mathring{\varphi}\left(\frac{y}{h} - k\right)\right\rangle \varphi\left(\frac{x}{h} - k\right)$$

$$= \int_{y \in R} s(y) h^{-1} K\left(\frac{x}{h}, \frac{y}{h}\right) \mathrm{d}y, \tag{B.7}$$

where $K(x, y)$ is the reproducing kernel associated with the approximation space $V(\varphi)$:

$$K(x, y) = \sum_{k \in Z} \varphi(x - k) \mathring{\varphi}(y - k).$$

By using the Taylor series expansion of the signal $s(y)$ around x in (B.7), we can express the approximation error as

$$s(x) - s_h(x) = -\frac{h^L}{L!} e_L\left(\frac{x}{h}\right) s^{(L)}(x) + \mathrm{O}(h^L), \tag{B.8}$$

where $s^{(L)}(x)$ is the Lth derivative of s and where

$$e_L(x) = \int_{-\infty}^{+\infty} (y - x)^L K(x, y) \mathrm{d}y.$$

Note thus there are no lower-order terms in the error since $K(x, y)$ perfectly reproduces all polynomials up to degree $L - 1$. The function $e_L(x)$ is zero-mean with periodicity one. It is best characterized by its Fourier series representation, which can be determined to be $e_L(x) = (-j)^L \sum_{k \neq 0} \hat{\varphi}^{(L)}(2\pi k) e^{j2\pi k t}$, where $\hat{\varphi}^{(L)}$ denotes the Lth derivative of the Fourier transform of φ. Interestingly, this formula is valid for any $\varphi \in V(\varphi)$ such that $\hat{\varphi}(0) = 1$ (not just quasi-interpolants).

For h sufficiently small, the $\mathrm{O}(h^{L+1})$ terms in (B.8) become negligible and the L_2-norm of the error has the following asymptotic form:

$$\lim_{h \to 0} \frac{\| s(x) - s_h(x) \|}{h^L} = \frac{1}{L!} \lim_{h \to 0} \left\| s^{(L)}(x) e_L\left(\frac{x}{h}\right) \right\|.$$

Note that $e_L(x/h)$ is periodic with period h and that its square modulus is given by

$$\frac{1}{h} \int_0^h |e_L(x/h)|^2 \mathrm{d}x = \int_0^1 |e_L(x)|^2 \mathrm{d}x$$

$$= \sum_{k \neq 0} |\hat{\varphi}^{(L)}(2\pi k)|^2.$$

As the period decreases ($h \to 0$), $s^{(L)}(x)$ can be approximated as a constant within each interval which leads to the derivation of the following asymptotic limit [18]:

$$\lim_{h \to 0} \frac{\| s - s_h \|}{h^L} = C_2 \cdot \| s^{(L)} \|, \tag{B.9}$$

where

$$C_2 = \frac{1}{L!}\left(\sum_{k \neq 0} |\hat{\varphi}^{(L)}(2\pi k)|^2\right)^{1/2}. \qquad \text{(B.10)}$$

References

[1] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[2] S.G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation", *IEEE Trans. PAMI*, Vol. PAMI-11, No. 7, 1989, pp. 674–693.

[3] P. Abry and A. Aldroubi, "Designing multiresolution analysis-type wavelets and their fast algorithms", *J. Fourier Anal. Appl.*, Vol. 2, No. 2, 1995, pp. 135–159.

[4] P. Dutilleux, "An implementation of the algorithme à trous to compute the wavelet transform", *Proc. Wavelets: Time-Frequency Methods and Phase Space*, 1989, pp. 298–304.

[5] M.J. Shensa, "The discrete wavelet transform: wedding the à trous and Mallat algorithms", *IEEE Trans. Signal Process.*, Vol. 40, No. 10, 1992, pp. 2464–2482.

[6] M. Unser, A. Aldroubi and S. J. Schiff, "Fast implementation of the continuous wavelet transform with integer scales", *IEEE Trans. Signal Process.*, Vol. 42, No. 12, 1994, pp. 3519–3523.

[7] M. Unser, "Fast Gabor-like windowed Fourier and continuous wavelet transforms", *IEEE Signal Process. Lett.*, Vol. 1, No. 5, 1994, pp. 76–79.

[8] D.L. Jones and R.G. Baraniuk, "Efficient approximation of continuous wavelet transforms", *Elect. Lett.*, Vol. 27, No. 9, 1991, pp. 748–750.

[9] S. Maes, "A fast quasi-continuous wavelet transform algorithm", *Proc. Workshop on Time, Frequency, Wavelets and Multiresolution Theory*, INSA-Lyon, 1994.

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

[11] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms", *IEEE Trans. Inform. Theory*, Vol. IT-38, No. 2, 1992, pp. 569–586.

[12] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory", *Numer. Funct. Anal. Optim.*, Vol. 15, Nos. 1&2, 1994, pp 1–21.

[13] 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, 1991, pp. 277–285.

[14] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, 1972.

[15] M. Unser, A. Aldroubi and M. Eden, "The L_2 polynomial spline pyramid", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 15, No. 4, 1993, pp. 364–379.

[16] 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.

[17] G. Strang, "Wavelets and dilation equations", *SIAM Rev.*, Vol. 31, 1989, pp. 614–627.

[18] M. Unser and I. Daubechies, "On the approximation power of convolution-based least squares vs. interpolation", *IEEE Trans. Signal Process.*, In press.