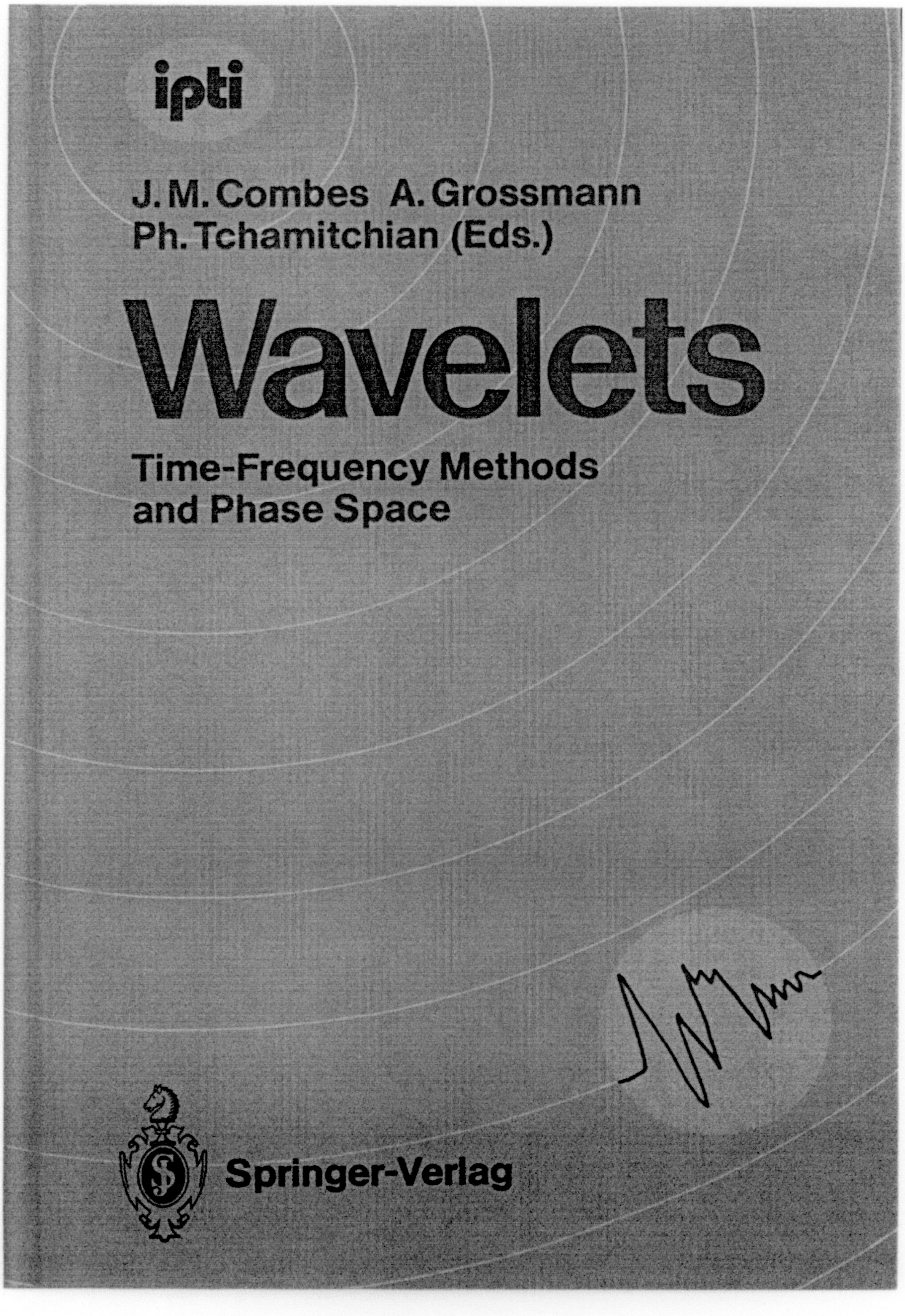


ipti

J. M. Combes A. Grossmann
Ph. Tchamitchian (Eds.)

Wavelets

Time-Frequency Methods and Phase Space

Springer-Verlag

Professor Jean-Michel Combes
Professor Alexander Grossmann
Professor Philippe Tchamitchian

Centre National de la Recherche Scientifique
Luminy – Case 907, F-13288 Marseille Cedex 9, France

ISBN 3-540-51159-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-51159-8 Springer-Verlag New York Berlin Heidelberg

Printing: Druckhaus Beltz, 6944 Hemsbach/Bergstr.
Binding: J. Schäffer GmbH & Co. KG., D-6718 Grünstadt
2157/3150-543210 – Printed on acid-free paper

A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform

M. Holschneider [1;*], *R. Kronland-Martinet* [2], *J. Morlet* [3], *and Ph. Tchamitchian* [4]

[1] Centre de Physique Théorique, CNRS Luminy, Case 907,
F-13288 Marseille Cedex, France
[2] Faculté des Sciences de Luminy and Laboratoire de Mécanique
et d'Acoustique, C.N.R.S., 31, Chemin J. Aiguier,
F-13402 Marseille Cedex 09, France
[3] TRAVIS, c/o O.R.I.C. 371 bis, Rue Napoléon Bonaparte,
F-92500 Rueil-Malmaison, France
[4] C.P.T. et Faculté des Sciences et Techniques de Saint-Gérôme,
F-13397 Marseille Cedex 13, France
*Permanent address: Mathematisches Institut, Ruhr Universität,
NA3 Postfach 102148, Universitätsstraße 150,
D-5630 Bochum 1, Fed. Rep. of Germany

1. Introduction

The purpose of this paper is to present a real-time algorithm for the analysis of time-varying signals with the help of the wavelet transform.We shall briefly describe this transformation in the following. For more details, we refer to the literature [1].

The main goal of the wavelet transform is to decompose an arbitrary signal into elementary contributions which are labeled by a scale parameter a. Consider a fairly arbitrary function g(t), which is localized both in the time and the frequency domain, and look at all its translated and dilated versions g((t-b)/a). Then the wavelet transform S(b,a) of a signal s(t) with respect to the wavelet g(t) is given by:

$$(1.1)\quad S(b,a) = \frac{1}{\sqrt{a}} \int \bar{g}\left(\frac{t-b}{a}\right) s(t)\, dt \qquad \text{(the bar denotes the complex conjugate).}$$

Expressing equation (1.1) in terms of Fourier transform we obtain the following:

$$(1.2)\quad S(b,a) = \sqrt{a} \int \bar{g}(a\omega)\, e^{ib\omega}\, s(\omega)\, d\omega$$

where the Fourier transform of a function f(t) is defined by : $f(\omega) = (2\pi)^{-1/2} \int f(t)\, e^{-i\omega t}\, dt$. So for the simplicity of notation we shall distinguish a function f(t) from its Fourier transform f(ω) only by its argument. Formulae (1.1) and (1.2) allow us to interpret the wavelet transform as a time-frequency analysis of s(t) with filters g (aω) of constant relative frequency resolution ($\Delta\omega/\omega = C^{te}$).

For mathematical reasons [1], the wavelet g(t) should satisfy the admissibility condition, which reads in Fourier space:

$$(1.3) \quad c_g = 2\pi \int |\hat{g}(\omega)|^2 \frac{d\omega}{|\omega|} < \infty.$$

This condition essentially means that g(t) is of zero mean $\int g(t)\, dt = 0$. In this case, the wavelet transform is invertible:

$$(1.4) \quad s(t) = \frac{1}{c_g} \int \int S(b,a) \frac{1}{\sqrt{a}}\, g\left(\frac{t-b}{a}\right) \frac{da\, db}{a^2}.$$

Here, we have supposed that the signal s(t) was of finite energy, $\int |s(t)|^2\, dt < \infty$. There exist many other reconstruction formulae. Some of them use only the values of S on a suitable grid [3].

The main properties of the transformation are :

- the correspondence s --> S is linear,

- the transformation preserves energy :

$$\int |s(t)|^2\, dt = 1/c_g \iint |S(b,a)|^2\, db\, da / a^2$$

In practice however, one works with sampled signals obtained from s(t) by measurements at the instants $t_i = i.T_s$ $(i \in \mathbb{Z})$, where $1/T_s$ is the sampling frequency. Therefore, formula (1.1) should be replaced by its discrete version:

$$(1.6) \quad S(iT_s,a) = T_s\, a^{-1/2} \sum_n s(n.T_s)\, \bar{g}\left(\frac{(n-i)\, T_s}{a}\right)$$

Now, suppose that the wavelet g(t) has finite support. In this case, the number of sampling points of g(t) at the scale a growths linearly with a. So the calculation of S with an algorithm based on the formula (1.6) cannot in general be satisfying on today's machines, especially in audio acoustic where the dilation parameter a ranges typically from 1 to 2^{10}, which corresponds to frequency analysis of the signal s(t) over 10 octaves. So, the need for a more elaborated algorithm is imperious.

2. A real time algorithm.

2.1 Notations and definitions

As a general notation we use the arguments of the functions to distinguish the different spaces. We define the following operators:

Let $r,h \in L^2$

Dilations: $(\mathfrak{D}_a r)(x) = a^{-1/2}\, r(x/a) \qquad (x \in \mathbb{R},\ a>0)$

Convolution: $(\mathcal{K}_h r)(x) = \int h(x-y)\, r(y)\, dy$

Inversion: $(\mathfrak{I}\, r)(x) = r(-x)$

Then the wavelet transform of a signal $s \in L^2$ with respect to the wavelet $g(t)$ is expressed as a set of convolutions, each of them labeled by the scale parameter a:

$$(2.1.1) \qquad S(.\,,a) = K_{\mathfrak{g}_a} s\,, \quad \text{with } \mathfrak{g}_a = \mathfrak{D}_a\, \mathfrak{I}\, \bar{g}\,.$$

In the following, we shall work with sequences $s \in l^2$, that is the space of sequences of complex numbers $s(n)$ ($n \in \mathbb{Z}$) of finite energy:

Energy: $\| s \| = \sum_n |s(n)|^2 < \infty.$

It is sometimes more convenient to use the z-transform of s which we denote s(z):

z-transform: $s(z) = \sum_n s(n).\, z^{-n}$

The following operators acting on sequences will be used constantly:

let $f,s \in l^2$, and $p \in \mathbb{N}$

Translations:

$(Ts)(n) = s(n-1)$

$(Ts)(z) = z^{-1}.\, s(z)$

Dilations :

$$(D_p s)(n) = \begin{cases} p^{-1/2}\, s(n/p) & \text{for } n = 0 \bmod p \\ 0 & \text{elsewhere} \end{cases}$$

$$(D_p s)(z) = p^{-1/2}\, s(z^p)$$

Convolutions:

$$(K_f s)(n) = \sum_m f(n-m)\, s(m)$$

$$(K_f s)(z) = s(z)\, f(z).$$

We shall denote by 1 the sequence which is zero everywhere except in 0 where it 1. So the identity can be written as:

Identity:

$$K_1\, s = s$$

The length of a sequence s (the number of non zero elements) will be denoted by $|s|$:

Length: $|s| = \sum_{s(n)\neq 0} 1.$

The most time consuming operations that we shall encounter are actually convolutions. In order to compare different algorithms we introduce the notion of complexity. For a convolution $K_f s$ it is quite reasonable to measure the complexity by the length $|f|$ of the filter f we convolute with:

Complexity: $|\, K_f \,| = |f|$

The reason for this is that this is exactly the number of operations - multiplication of two numbers and addition of the result to an accumulator - to realize this convolution. The complexity of the product of two convolutions is given by the sum of the respective complexities:

$$|\, K_f\ K_h \,| = |f| + |h| \qquad (2.1.2)$$

The passage from an everywhere defined function r(x) to a sequence is done by the perfect sampling operator P. For the sake of simplicity, let us suppose that the sampling time is unity, $T_s=1$.

Sampling: $(Pr)(n) = r(n) \qquad (n\in \mathbb{Z})$

In view of formula (1.6), we define the discrete wavelet transform of a sampled signal $s \in l^2$ with respect to the wavelet g(t) as a set of convolutions with filters g_a labeled by the scale parameter a:

$$(2.1.3) \qquad S_a = K_{g_a} s \, ; \qquad \text{with } g_a = P \, \mathfrak{D}_a \mathfrak{I} \; \bar{g} \, .$$

2.1 The need of an efficient algorithm

We now want to calculate the discrete wavelet transform for N octaves ; that is the scale parameter a takes the values $a=1,2,4,\dots,2^N$. Typically N is of the order of magnitude of 10 e.g as in audio-acoustic applications.

Obviously, there is a direct method of computing S_a just by using the definition (2.1.3), and evaluating the convolutions. So suppose now that $g = g_{a=1}$has finite length, $|g| < \infty$. In practical applications this will always be the case. Then the complexity of this algorithm to calculate the n-th octave, $S_{a=2^n}$, can be estimated as follows:

$$(2.1.1) \qquad \text{complexity} = | K_{g_{a=2^n}} | = | g_{a=2^n} | \sim |g| . \, 2^n$$

We see that the amount of calculation grows exponentially with the number of octaves, which is a serious problem when n~10.

2.3 A class of wavelets.

The algorithm we shall establish now will reduce the complexity of the convolution with the dilated wavelet, by factorizing it into convolutions with smaller filters (compare with formula 2.1.2). This will be possible under certain hypotheses on the wavelet .

In a first step, we shall construct an operator acting on sequences which shall be the analog of the dilation operator $\mathfrak{D}_2$ acting on functions. To be more precise, we are looking for an operator $O: l^2 \dashrightarrow l^2$ satisfying

$$(2.3.1) \qquad O^n \, P \, g = P \, (\mathfrak{D}_2)^n \, g \, , \qquad (n \in \mathbb{N})$$

for a sufficiently large class of functions g. In particular this class should contain some interesting wavelets. Equation (2.3.1) means that sampling the dilated versions of g can be replaced by the action of O on the original sampled sequence. Additionally we should require that O is numerically simple. The a priori choice $O = D_2$ is not satisfying since there are too few functions satisfying (2.3.1): the only continuous function satisfying (2.3.1) is g = 0 . This is due to the fact that $(D_2 g)(n) = 0$ whenever n is odd, independently of its neighbouring values. A better choice might be to obtain the values at the odd position by means of an interpolation procedure. Let us suppose that there is a filter $F \in l^2$ doing this job for us. We then define

$$(2.3.2) \qquad O = D_2 + T\, D_2\, K_F$$

To illustrate the action of O, let us give two examples.

a) Piecewise affine functions

Let F be given by : $F(-1) = F(0) = 1/2$, all other elements are zero, then O is doing a dilation by means of linear interpolation:

$$(Og)(n) = \begin{cases} 2^{-1/2}\, g(n/2) & \text{for n even} \\ 2^{-1/2}\, (1/2)\, [\, g(\,(n-1)/2\,) + g(\,(n+1)/2\,)] & \text{for n odd} \end{cases}$$

The class of continuous functions for which (2.3.1) holds are exactly the functions which are affine on each interval $[n,n+1[$.

b) Piecewise constant functions

Let F be defined by : $F(0) = 1$, all other elements are zero, then the action of O is :

$$(O\,g)(n) = (O\,g)(n+1) = g(n/2) \quad \text{for n even}$$

There is actually no continuous function satisfying 2.3.1 apart from the trivial one, g=0. However, the piecewise continuous function satisfying (2.3.1) are the functions that are constant on any interval $[n,n+1[$.

From these two examples, one might be tempted to guess that filters corresponding to higher order Lagrangian interpolation (quadratic, cubic, ...) might give rise to the corresponding spline functions. But this is not true. However for higher order interpolations, the functions satisfying 2.3.1 become more and more regular.[2]

However, in view of numerical applications, condition (2.3.1) is much too strong. Instead it should be sufficient to require that the difference , e.g. in norm, of the right and the left hand side are smaller than some given precision ε for all N octaves in consideration:

$$(2.3.3) \qquad \| O^n P g - P\, (\mathbb{D}_2)^n g \| < \varepsilon\, ; \quad 0 \le n \le N$$

This condition can easily be checked numerically for a given function g.

We now want to show, that convolutions $K_{O^n g}$, with dilated versions of a filter g, can be factorized into convolutions with smaller filters:

Lemma: let $g \in l^2$, and let $F \in l^2$ be a filter defining the pseudo-dilation operator O. Then the convolution operator $K_{O^n g}$ factorizes into simpler convolutions: ($\alpha = 1/\sqrt{2}$)

$$(2.3.4) \qquad K_{O^n g} = \alpha^n K_{g_n} K_{F_1} \ldots K_{F_n}$$

with $\qquad F_1 = 1 + T(\alpha^{-1} D_2) F, \qquad F_{i+1} = (\alpha^{-1} D_2) F_i, \qquad g_n = (\alpha^{-1} D_2)^n g.$

So with the help of this lemma we can realize the calculation of the n-th octave of the wavelet transform, which is a convolution with a filter of length $|O^n g| = 2^n |g|$, with the help of smaller convolutions, which correspond to an algorithm of complexity

$$(2.3.5) \qquad | K_{g_n} \; K_{F_1} \ldots K_{F_n} | = |g| + n (1 + |F|).$$

So, for wavelets satisfying (2.3.3), we have reduced the exponential growth in n of (2.1.1) to a linear one. More than that, as we shall see, the calculations for N consecutive octaves can be organized in a hierarchic way, yielding an additional gain of calculation time.

Proof of the lemma:

Let us write O in the z - representation:

$$(Og)(z) = 2^{-1/2} g(z^2) [1 + z^{-1}F(z^2)].$$

Iterating this identity yields:z^{2^n}

$$\begin{aligned}(O^n g)(z) &= 2^{-1/2} (O^{n-1} g)(z^2) \; [1 + z^{-1}F(z^2)] \\ &= 2^{-n/2} g (z^{2^n}) \; [1 + z^{-1}F(z^2)] \; [1 + z^{-2}F(z^4)] \ldots [1 + z^{-2^{n-1}} F(z^{2^n})].\end{aligned}$$

So using the z representation of the dilation D_2 and the convolution, we have proven the lemma.

3. The implementation of the algorithm

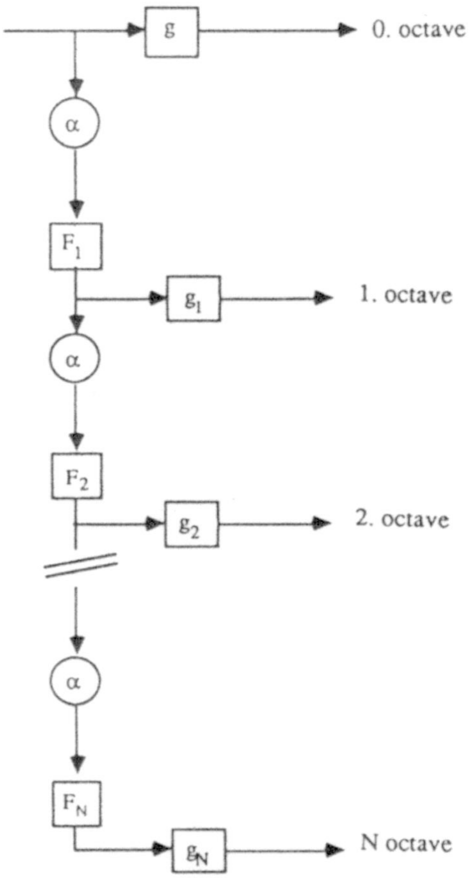
We now shall give two possible implementations using the algorithm presented above to calculate the wavelet transform for N octaves of the signal s with respect to the wavelet g. The hierarchic structure is clarified if one rewrites the necessary operations in the following way: ($\alpha = 1/\sqrt{2}$)

(3.1)

$$\begin{array}{lll}
 & S_{a=1} = K_g\, s; & \text{(0 octave)} \\
F_1 = 1 + T(\alpha^{-1} D_2)\, F & X_1 = \alpha\, K_{F_1}\, s; & \\
g_1 = \alpha^{-1} D_2\, g & S_{a=2} = K_{g_1}\, X_1; & \text{(1 octave)} \\
F_2 = \alpha^{-1} D_2\, F_1 & X_2 = \alpha\, K_{F_2}\, X_1; & \\
g_2 = \alpha^{-1} D_2\, g_1 & S_{a=22} = K_{g_2}\, X_2; & \text{(2 octaves)} \\
\quad \ldots\ldots & & \\
F_N = \alpha^{-1} D_2\, F_{N-1} & X_N = \alpha\, K_{F_N}\, X_{N-1}; & \\
g_N = \alpha^{-1} D_2\, g_{N-1} & S_{a=2N} = K_{g_N}\, X_N; & \text{(N octaves)}
\end{array}$$

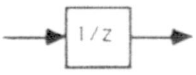
In the following we shall present two possible implementations of this algorithm. First we define some symbols that we shall encounter throughout this section.

(3.2)

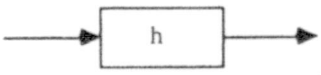


The convolutions with filters F_i , which are all the dilated versions of one fixed filter can be realized by an "algorithme à trous". We suppose that the non-zero elements of h are h(n), h(n+1) ... h(m).

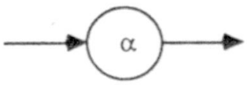
A delay shall be denoted by:



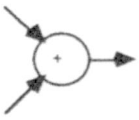
The convolution with a filter h shall be denoted by:



The multiplication by a (complex) number α shall be symbolized by:

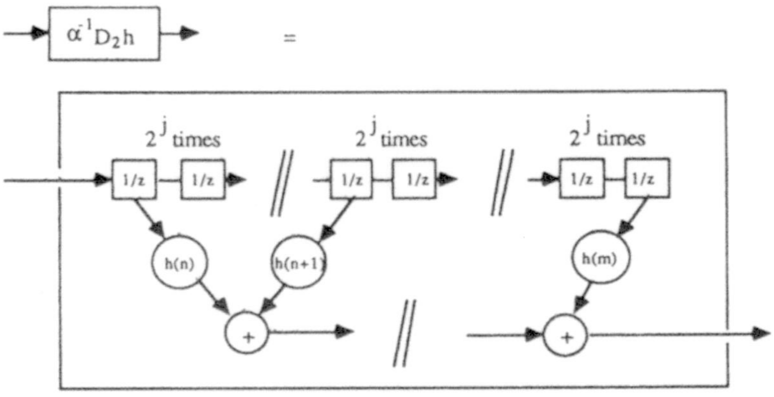


The addition of two numbers shall be symbolized by:

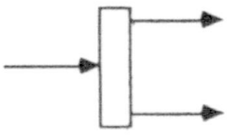


Then a first implementation of the algorithm is merely a direct translation of formula (3.1). It is given by the following diagram:

(3.3)

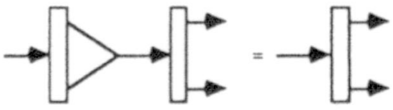


Another possible implementation makes use of a multiplexer:



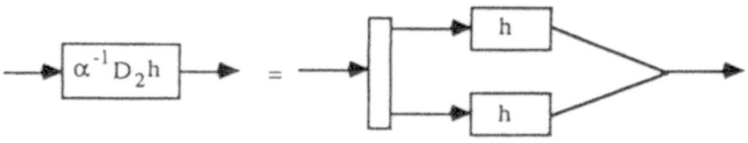
It separates a sequence s(n) into an even (s(2m)) and an odd (s(2m+1)) sequence. The following multiplexer identity is obvious:

(3.4)

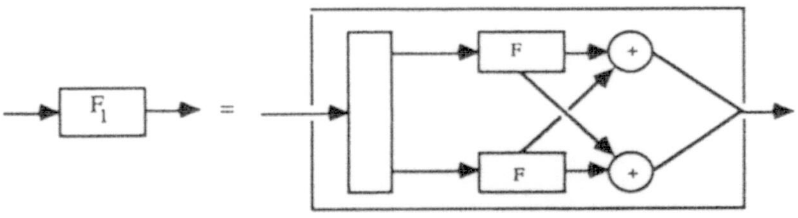


Then the convolution with a dilated filter $\alpha^{-1}D_2\,h$ is realized as: $(\alpha=1/\sqrt{2})$

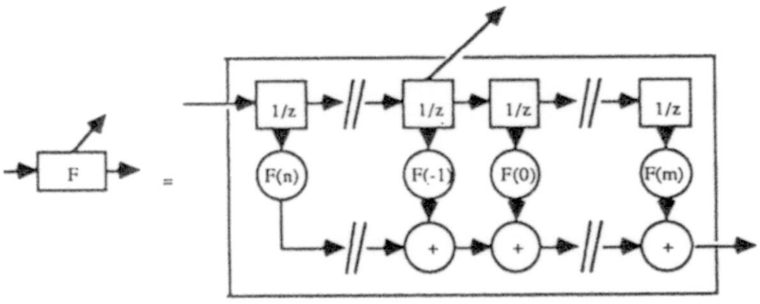
(3.5)



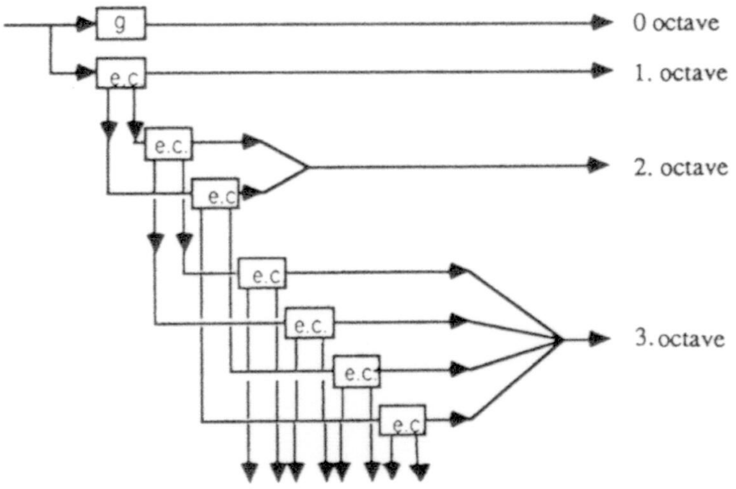
The convolution with $F_1 = 1 + T\,\alpha^{-1}D_2\,F$ is obtained by the following butterfly diagram:



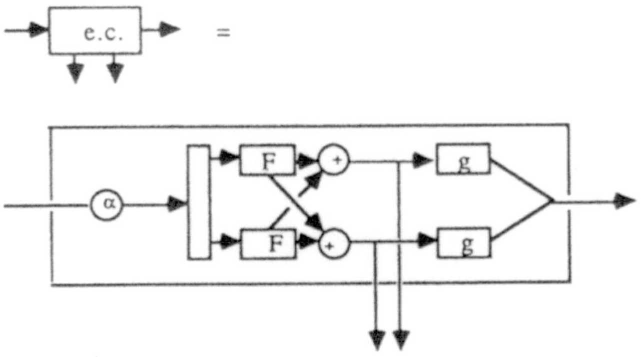
Here we have used the following symbol:



If we now replace all these identities in diagram (3.2) and do some graph algebra, then we see that the calculation of the wavelet transform on N octaves can be realized as follows: (N=3)



Here we have used the following abbreviation for the elementary cell:



4. The wavelet transform on N voices.

Up to now we only showed how to realize a real time algorithm to compute the wavelet transform on N octaves, which corresponds to a geometric progression in the scale variable a. It sometimes may be necessary to calculate the wavelet transform for dilation parameters which progress arithmetically: a = 1, 2, ... N.

In a first step we replace the dilation by 2 encountered in section 2 by any dilation by an integer number p. In complete analogy with (2.3.2) we define a dilation on sequences with the help of p-1 interpolation filters $F_1 \dots F_{p-1}$:

$$(4.1) \quad O_p = D_p + T D_p K_{F_1} + T^2 D_p K_{F_2} + \dots + T^{p-1} D_p K_{F_{p-1}} .$$

The following lemma is a generalization of the lemma of section 2. It shows how to decompose the convolution with a dilated filter $h_{p,q} = O_p\, O_q\, h$, by smaller convolutions:

Lemma: Let P_i, Q_i be the interpolation filters for O_p, O_q respectively. Then the convolution with the dilated filter $O_p\, O_q\, g$ factorizes as follows:

$$K_{O_{p,q}g} = K_{\mathfrak{g}}\ K_{\mathcal{Q}}\ K_{\mathcal{P}}$$

$$\text{with} \quad \mathcal{P} = \qquad 1 + T\,(\sqrt{p}D_p)\,P_1 + \ldots + T^{p-1}\,(\sqrt{p}D_p)\,P_{p-1}$$

$$\mathcal{Q} = (\sqrt{p}D_p)\ (1 + T\,(\sqrt{q}D_q)\,Q_1 + \ldots + T^{q-1}\,(\sqrt{q}D_q)\,Q_{q-1})$$

$$\mathfrak{g} = (p.q)^{-1/2}\,D_{p.q}\,g.$$

The proof of this lemma is as straight forward as for the lemma in the previous section.

Let us now suppose that for any prime number p we have chosen the interpolation filters. Then we can simulate the dilation by any integer N of the sampled wavelet g in the following way: we first factorize N into prime numbers, $N = p_1 \ldots p_m$, and then we define the dilated version g_N of g as:

$$(4.2) \qquad g_N = O_{p_1} \ldots O_{p_m}\, g.$$

Then the calculation of the voice corresponding to the convolution with g_N can be factorized into smaller convolutions if N itself is not prime. The complexity of this algorithm depends on some number theoretic properties of N.

There is an order problem in equation (4.2), since the continuous dilations commute whereas its discrete analogs do not in general. But for convenient wavelets g we may expect that the energies of the commutators applied to g are small. In particular for pseudo-dilation operators corresponding to linear interpolations, the commutators of these operators vanish on the affine wavelets.

References:

[1] Proceedings of the meeting "Ondelettes, méthodes temps-fréquence et espace des phases" and references in there. C.I.R.M Luminy Marseille France 14 -18 déc. 1987 to appear.

[2] M.Holschneider, R. Kronland-Martinet, J. Morlet, Ph. Tchamitchian, The "algorithme à trous", in preparation.