Articles sur les transformées en ondelettes soumis par le département ETP

- Centre de Recherches En Physique de L'Environnement Terrestre Et Planétaire

CENTRE NATIONAL D'ETUDES
DES TELECOMMUNICATIONS

CENTRE NATIONAL DE LA
RECHERCHE SCIENTIFIQUE

CENTRE DE
RECHERCHES
EN PHYSIQUE DE
L'ENVIRONNEMENT
TERRESTRE
ET PLANETAIRE

CRPE

NOTE TECHNIQUE

CRPE/192

ARTICLES SUR LES TRANSFORMEES EN ONDELETTES SOUMIS PAR LE DEPARTEMENT ETP

Par

COLLECTIF GROUPE "TRAITEMENT DU SIGNAL"

RPE/ETP
38-40, rue du Général Leclerc
92131 ISSY-LES-MOULINEAUX, FRANCE

CENTRE NATIONAL D'ETUDES
DES TELECOMMUNICATIONS
Centre Paris B

CENTRE NATIONAL DE LA
RECHERCHE SCIENTIFIQUE
Département TOAE

**CENTRE DE RECHERCHES EN PHYSIQUE DE
L'ENVIRONNEMENT TERRESTRE ET PLANETAIRE**

NOTE TECHNIQUE CRPE/192

ARTICLES SUR LES TRANSFORMEES EN ONDELETTES

SOUMIS PAR LE DEPARTEMENT ETP

par

COLLECTIF GROUPE "TRAITEMENT DU SIGNAL"
RPE/ETP
38-40 rue du Général Leclerc, 92131 Issy-les-Moulineaux

Le Directeur

G. SOMMERIA

Le Directeur Adjoint

J.L. BERTHELIER

Avril 1991

LISTE DE DIFFUSION SYSTEMATIQUE

CNET

MM.	FENEYROL	Directeur du CNET
	THABARD	Directeur Adjoint du CNET
	COLONNA	Adjoint Militaire au Directeur du CNET
	MEREUR	Directeur des Programmes
	BLOCH	DICET
	THUE	DICET
MME	HENAFF	DICET
MM.	PIGNAL	PAB
	RAMAT	PAB
	ZYLBERSZTEJN	PAB-BAG
	ABOUDARHAM	PAB-SHM
	HOCQUET	PAB-STC
	THEBAULT	PAB-STS
MME	PARIS	PAB-RPE
MM.	BAUDIN	PAB-RPE
	BERTHELIER	PAB-RPE
	BIC	PAB-RPE
	CERISIER	PAB-RPE
	GENDRIN	PAB-RPE
	LAVERGNAT	PAB-RPE
	ROBERT	PAB-RPE
	ROUX	PAB-RPE
	SOMMERIA	PAB-RPE
	TESTUD	PAB-RPE
	VIDAL-MADJAR	PAB-RPE

CNRS

MM.	BERROIR	TOAE
	CHARPENTIER	SPI
MME	SAHAL	TOAE
MM.	COUTURIER	INSU
	CADET	INSU

CNES

MMES	AMMAR
	DEBOUZY
MM.	BAUDOIN
	FELLOUS
	HERNANDEZ (Toulouse)

EERM

. ANDRE

MEN/DRED

M. MEGIE

Bibliothèques

CNET-SDI (3)
CNET-EDB
CNET-RPE (Issy) (5)
CNET-RPE (St Maur) (2)
Observatoire de Meudon
CNRS-SA
CNRS-INIST
CNRS-LPCE

LISTE COMPLEMENTAIRE

CCETT	DOCUMENTATION
CCETT/PFI/IPT	2 EX
CCETT/PFI/PVI	BOTREL
CCETT/SRL/DCS	EVAIN
CCETT/SRL/DCS	HELARD
CCETT/SRL/DCS	PALICOT
CCETT/TAV/CEP	MAU
CCETT/TAV/SVS	1 EX
CCETT/TAV/SVS	BRUSA
CNS/CCI/CAT	PRIVAT
CNS/CCI/CAT	SENN 2 EX
CNS/CCI/DCC	LARDY 2 EX
CNS/TCI/TMF	MARTIN
LAA/RSM/TTA	LE GUIGNER
LAA/SLC/EVP	GUILLEMIN
LAA/TSS/CMC	DIFFRANCESCO
LAA/TSS/CMC	GILLOIRE
LAA/TSS/RCP	HAFFNER
LAA/TSS/RCP	MICLET
LAA/TSS/CMC	LE GUYADER
LAA/TSS/RCP	JOUVET
LAA/TSS/RCP	WHITE
LAA/TSS/RCP	SORIN
LAB/IFE/CIO	TREGUIER
LAB/IFE/COD	LE MOIGNE 3 EX
LAB/OCM/FOG	MONERIE 2 EX
LAB/OCM/MPA	CLEROT
LAB/PTI/GER	LASSUDRIE
LAB/SMR	LECLERCT
LAB/SMR	VANDAMME
PAA/SRE	GIRARD
PAB/BAG/PMM	BENSOUSSAN
PAB/DIR	CAQUOT
PAB/RPE/EMI	FORGET
PAB/RPE/ETP	BARBOT
PAB/RPE/ETP	BLU
PAB/RPE/ETP	CIARLETTI
PAB/RPE/ETP	GLOAGUEN
PAB/RPE/ETP	GOULLAM
PAB/RPE/ETP	HELLION
PAB/RPE/ETP	HILAL
PAB/RPE/ETP	LEVY
PAB/RPE/ETP	MALLET
PAB/RPE/ETP	MAYRARGUE
PAB/RPE/ETP	ROSSI
PAB/RPE/ETP	YVON
PAB/RPE/OBT	TRESSENS
PAB/RPE/OPN	CHANTEUR
PAB/RPE/TID	ROBERT
PAB/STC/ECG	REBOURG 2 EX

EXTERIEUR

ENST/COM	VALLET
ENST/ELEC	MOU
ENST/IMA	MUGNIER
ENST/SIG	BARRAL
ENST/SIG	CHAIGNE
ENST/SIG	CHOLLET
ENST/SIG	FARAH
ENST/SIG	GRENIER
ENST/SIG	RAMDANE
ENST BR	DOCUMENTATION
ENST	DOCUMENTATION
INRA	DOCUMENTATION
TRT	DOCUMENTATION

INTRODUCTION

Le groupe "Traitement du Signal" du Département ETP publie régulièrement en note technique les articles soumis à des revues de comité de lecture, afin d'une part de diffuser l'information en interne au CNET, et d'autre part de disposer d'une référence interne permettant de dater le travail. En effet, le délai de parution des articles dans les revues à comité de lecture comme IEEE Transactions on Acoustics, Speech and Signal Processing est très long (2 à 3 ans d'attente de la soumission à la parution).

Cette note technique regroupe plus particulièrement les articles consacrés aux transformées en Ondelettes. La plupart de ces travaux ont été réalisés dans le cadre d'une thèse ENST suivie par O. Rioul et dirigée par P. Duhamel. L'article 1 est un article de vulgarisation destinés aux traiteurs de signaux. L'article 2 unifie la présentation de la version discète de la transformée en ondelettes et fait le lien avec d'autres transformées (pyramidales, banc de filtres). L'article 3 se consacre à la nouvelle propriété de "régularité" apportée par la théorie des Ondelettes. Enfin, l'article 4 traite les problèmes d'implantation et de réduction de charge de calcul.

SOMMAIRE

ARTICLE 1

"Wavelets Transforms in Signal Processing", par O. RIOUL et M. VETTERLI

IEEE ASSP Magazine, soumis en Déc. 1990

Dec. 1990. Submitted IEEE S.P. Magazine, Oct. 1991 issue.

Wavelet Transforms in Signal Processing

OLIVIER RIOUL[*] AND MARTIN VETTERLI[†]

[*]Centre National d'Etudes
des Télécommunications
CNET/PAB/RPE
38-40 rue du Général Leclerc
92131 Issy-Les-Moulineaux
France

[†]Department of Electrical Engineering
and Center for Telecommunications
Research
Columbia University
New York, NY 10027-6699
U.S.A.

INTRODUCTION

Wavelets have recently attracted attention in several fields of applied mathematics and engineering [WAV89]. In particular, they are of interest for the analysis of non-stationary signals, a central problem in signal processing; the Wavelet Transform (WT) provides an alternative to the Gabor or Short-Time Fourier Transform (STFT) [ALL77], [GAB46] and is related to the Wigner-Ville distribution [BOU85], [CLA80], [FLA90] as well.

Like these techniques, the WT is invertible and can be applied to general signals, since no apriori assumptions are made on them. In constrast to the STFT, the WT performs a constant relative bandwidth analysis, providing a different tiling of the time-frequency plane. In fact, the notion of scale is introduced as an alternative to Fourier frequency, leading to a so-called time-scale representation.

Wavelet analysis and synthesis may also be seen as a signal expansion into particular bases. The wavelets are defined as the corresponding basis functions; they are generated by time-shifts and dilations/contractions from a single prototype, which can be though of as a bandpass filter.

Similar to other transforms common in signal processing, the WT, which maps a signal into a time-scale plane, can be defined as a continuous [GRO89] or discrete [DAU88], [DAU90a], [MEY89] representation of continuous-time signals, or can be defined for discrete-time signals as well [DAU88], [RIO90b], [VET90a], [VET90b].

The idea of looking at a signal at various scales and analysing it with various resolutions has emerged independently in many different fields of mathematics, physics and engineering. Although similar ideas and constructions took place as soon as the beginning of the century [HAA10], [FRA28], [LIT37], [CAL64], [YOU78], wavelet theory has been developped as a unifying framework in the mid-eighties by researchers of the "French school," under the impulsion of a geophysicist, a theoretical physicist and a mathematician (namely, J. Morlet, A. Grossmann, and Y. Meyer.) They built strong mathematical foundations around the subject and labelled their work by the name of "*Ondelettes*" (Wavelets.) They also considerably interacted with other fields. In this context, I. Daubechies and S. Mallat soon caught the attention of the signal processing community on the subject of wavelets [DAU88], [MAL89a]. Since then, a number of theoretical, as well as practical contributions have been made on various aspects of WTs, and the subject of wavelets is growing rapidly. There remains, however, a large number of open questions and still-unclear points that are amenable to a specific decision, depending on the application (e.g. the choice of the "wavelet prototype".)

This review covers the main definitions and properties of wavelet transforms, while focusing on signal processing applications. Its purpose is to present a simple, synthetic view of the subject. Non-specialists interested in reading further on a particular advanced problem are referred to the bibliography.

There have been different ways of considering wavelets. Some see them as a very promising brand new theory [CIP90]. Others doubt that they could truly be a major breakthrough in signal processing, since depending on the field one works in, WTs can be seen either as constant-Q analysis [YOU78], wide-band cross-ambiguity functions [SPE67], affine coherent states' expansions [DAU90a], [PAU85], Frazier-Jawerth transform [FRA86], perfect reconstruction octave-band filter banks [EST77], [MIN85], [SMI86], [SMI87], or a variation of Laplacian pyramid decomposition [BUR83], [BUR89]! We think that the interest and merit of wavelet theory is to unify all this into a common framework, thereby allowing new ideas and developments.

I. NONSTATIONNARY SIGNAL ANALYSIS

The aim of signal analysis is to extract some kind of relevant information from a signal, by transforming it. "Parametric" methods include *a priori* assumptions on the signal in the analysis; this may yield sharp estimations if these assumptions are valid, but is obviously not of general applicability. In this paper we focus on "non-parametric" methods applicable to any general signal [FLA89]. In addition, we shall consider invertible analysis transformations: the analysis thus unambiguously represents the signal, and more involved functions such as parameters' estimation, coding, pattern recognition can be performed on the "transform side," where relevant properties have been exhibited.

Our approach is to consider a general signal as being a non-stationary signal; we shall therefore first give indications on which type of linear analysis is adapted for stationary signals and then extend it to non-stationary signals.

Intuitively, stationarity means that the properties of the signal do not evolve in time. This notion is formalized when e.g. the signal is modelized as a stationary statistical process. For such signals $x(t)$, the natural "stationary transform" is the well-known Fourier transform [FOU88]:

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-2j\pi ft}\, dt \tag{1}$$

The analysis coefficients $X(f)$ which define the notion of global frequency f in a signal, are computed as inner products of the signal with sinewave basis functions of infinite duration. As a result, Fourier analysis works well if the (deterministic) signal is composed of a few stationary (e.g., sinewaves) components. However, any abrupt change in time in a non-stationary signal is spread out over the whole frequency axis in $X(f)$; therefore an analysis adapted to non-stationary signals requires more than the Fourier Transform.

Two different views of the same extension of the Fourier Transform are here possible, with one common purpose: introduce time dependency in the analysis (while preserving linearity):

1) Define a local frequency parameter by adapting the Fourier Transform to occurences of limited duration (over which stationarity is assumed). This is the aim of the next section.

2) Modify the sinewave basis functions used in the Fourier Transform to functions more concentrated in time (at the expense of loss of Fourier frequency resolution.) We shall see that this approach leads to the same methods as in 1).

II. SCALE VS. FREQUENCY

II.1. The Short-Time Fourier Transform: Analysis with Fixed Resolution.

One way to introduce a frequency depending on time is to define the instantaneous frequency as proportional to the time-derivative of the phase of the analytic signal [FLA89]. If the signal is not narrow-band, however, the instantaneous frequency averages different spectral components in time. To become accurate in time, we therefore need one more dimension, that is a two-dimensional time-frequency representation $S(t,f)$ of the signal $x(t)$ composed of spectral characteristics depending on time. That kind of representation is similar to the notation used in a musical score. The local frequency f is defined through an appropriate definition of $S(t,f)$.

A straightforward adaptation of the Fourier Transform to define $S(t,f)$ was first used by Gabor [GAB46] and is as follows: consider a signal, and assume it is stationary when seen through a window g of limited extent, centered at time location τ:

$$x(t)g(t-\tau)$$

then do Fourier analysis: this yields the Short-Time Fourier Transform (STFT)

$$\mathrm{STFT}(\tau,f) = \int x(t)g(t-\tau)\, e^{-2j\pi ft}\, dt \tag{2}$$

which maps the signal into a two-dimensional function in a time-frequency plane (τ,f). Note that, although the local frequency f is still very close to the Fourier frequency (the Fourier Transform properties remain built in the definition of the STFT), it crucially depends on the choice of the window $g(t)$. The time-frequency plane is, by definition, filled "column by column", i.e. Fourier Transform after Fourier Transform when the window $g(t)$ "slides" in time through the signal evolutions. However, as shown in Fig. 1, the STFT may be interpreted "line by line" as well: for a fixed analysed frequency, as τ varies, the STFT is the output of some filter whose impulse response is the modulated window $g(t)\, e^{2i\pi ft}$ (within a phase factor). In other words, the STFT may also be seen as a modulated filter bank [ALL77].

From these two dual interpretations a fundamental drawback of the STFT, related to the so-called time and frequency resolution, can be shown. Frequency (or time) resolution means power of discrimination in the analysis. From what has just been said it is easy to find the conditions under which two pure sinwaves (two short bursts, respective.y) can be distinguished by the analysis in frequency (in time, respectively:) the difference of the two frequency (time, respectively) locations must be smaller than the extent of the Fourier Transform of $g(t)$: $G(f)$ ($g(t)$, respectively). These extents can be formally defined as

$$\Delta f^2 = \int f^2 |G(f)|^2 df \qquad \Delta t^2 = \int t^2 |g(t)|^2 dt \tag{3}$$

A small Δf (Δt, respectively) contributes for a good frequency (time, respectively) resolution in the analysis. The fundamental drawback concerning Δt and Δf is two-fold:

1) Increasing time resolution (i.e., narrowing the window) is done necessarily at the expense of decreasing frequency resolution, and *vice versa*. This is due to the so-called uncertainty principle, or Heisenberg's inequality

$$\text{Time - Bandwidth Product} = \Delta t \Delta f \geq \frac{1}{4\pi}$$

Gaussian windows are often used [GAB46], [ALL77], [DAU90a] since they reach the lower bound.

2) Fig. 2.(a) shows that once a window is chosen (i.e., once the STFT, or the analysis, is chosen), the resolution capabilities in time and frequency of the STFT remain fixed all over the time-frequency plane, i.e. for all analysed points (τ, f). Consequently, if a more accurate representation (in time of frequency) around a particular pattern in the time frequency plane is needed, the STFT has to be recomputed all over again.

Therefore, since the STFT has limited resolution capabilities, it is only adapted to signals with limited variations and dynamics in time and frequency. For example, if the signal is composed of small bursts associated to long, quasi-stationary components, each type of component can be analysed with good time or frequency resolution, but not both.

II.2. The Continuous Wavelet Transform: a Multiresolution Analysis.

To overcome the resolution limitation of the STFT (that is point 2) above), one can imagine to let the resolution Δt, Δf vary in the time-frequency plane, hence to obtain a *multiresolution* analysis. Intuitively, when the analysis is viewed "line by line" as a filter bank, the time resolution must increase with the central frequency of the analysis filters. We therefore impose that Δt is inversely proportional to f, or:

$$\frac{\Delta f}{f} = Cst. \tag{4}$$

The filter bank is then composed of band-pass filters with constant relative bandwidth (so-called "constant-Q" analysis). Another way to say this is that instead of being regularly spread over the frequency axis (as for the STFT case), the frequency responses of the analysis filters are regularly spread in a logarithmic scale over the frequency axis (see Fig. 3). This kind of filter bank is used for modelizing frequency responses of the cochlea situated in the inner ear and is therefore adapted to auditory perception, e.g. of music: filters satisfying (4) are naturally distributed into octaves.

Since (4) is satisfied, the analysis filters no longer limit the resolution in time or frequency. For example, two very close short bursts can always be eventually separated in the analysis by going up to higher analyzed frequencies to increase time resolution (see Fig.2.(b)). This kind of analysis of course works best if the signal is composed of high frequency components of short duration plus low frequency components of long duration.

The *Continuous Wavelet Transform* (CWT) exactly follows these ideas while adding a simplification: all impulse responses of the filter bank are defined as *scaled* (i.e. stretched or compressed) versions of the same prototype $h(t)$:

$$h_a(t) = \frac{1}{\sqrt{|a|}}\, h\!\left(\frac{t}{a}\right)$$

where a is a *scale* factor (the constant $\frac{1}{\sqrt{|a|}}$ is there for energy normalization.) This results in the definition of the CWT:

$$\mathrm{CWT}_x(\tau, a) = \frac{1}{\sqrt{|a|}} \int x(t)\, h\!\left(\frac{t-\tau}{a}\right) dt \tag{5}$$

Since the same prototype $h(t)$, called *basic wavelet*, is used for all filter impulse responses, no specific scale is privileged, i.e. wavelet analysis is self-similar at all scales. Moreover this simplification is useful to derive mathematical properties of the CWT.

In connection with the modulated window used in the STFT, the basic wavelet could be chosen as a modulated window [GOU84], [GRO89]:

$$h(t) = g(t)e^{-2j\pi f_0 t}$$

Then the frequency responses of the analysis filters indeed satisfy (4) with the identification

$$a = \frac{f_0}{f}$$

But more generally $h(t)$ can be any band-pass function and the scheme still works. In particular one can dispense with complex-valued transforms and deal only with a fully real-valued definition.

It is important to note that the notation f for local frequency in this section has nothing to do with that described for the STFT: indeed, it is associated to the scaling scheme (see Box 1.) As a result, this local frequency (whose definition depends on the basic wavelet) is no longer adapted to modulations (as was the case for the STFT) but is now well adapted to dilations/contractions, i.e., scaling a signal scales its CWT accordingly. This is the reason why the terminology "scale" is often preferred to "frequency" for the CWT, the word "frequency" being reserved for the STFT.

III. WAVELET ANALYSIS AND SYNTHESIS

Another way to introduce CWTs is to define *wavelets* as basis functions. Indeed basis functions already appear in the preceding definition (5) when one sees it as an inner product:

$$\mathrm{CWT}_x(\tau,a) = \int x(t)\, h_{a,\tau}^*(t)\, dt$$

which measures the "similarity" between the signal and the basis functions $h_{a,\tau}(t) = \frac{1}{\sqrt{a}} h^*\left(\frac{t-\tau}{a}\right)$ called *wavelets*. The wavelets are scaled and translated versions of the basic wavelet prototype $h(t)$ (see Fig.2.(c)-(d)). A similar point of view exists for the STFT; for both transforms, the sinewaves basis functions of the Fourier Transform are replaced by more localized reference signals (such as modulated windows or wavelets) labelled by time and frequency (or scale) parameters. In fact both transforms may be interpreted as ambiguity functions used in radar or sonar processing (see Box 2.)

Since wavelet analysis is defined as wavelet basis coefficients, we expect that any general signal can be represented as a decomposition into wavelets, i.e. that the original waveform is synthesized by adding elementary building blocks supported by the wavelets, of constant shape but different size and amplitude. Another way to say this is that we want the continuously labelled wavelets $h_{a,\tau}(t)$ to behave just like an *orthonormal basis* [MEY90]: the analysis is done by inner products, and the synthesis consists in summing up all the orthogonal projections of the signal onto the wavelets:

$$x(t) = Cst \iint_{a>0} \mathrm{CWT}(\tau,a)\, h_{a,\tau}(t)\, \frac{da\, d\tau}{a^2} \qquad (6)$$

The measure in this integration is formally equivalen : to *dt df* [GOU84]. We have assumed here that both signal and wavelets are either real-valued or complex analytic so that only positive dilations $a>0$ have to be taken into account. Otherwise (6) is more complicated.

Of course the $h_{a,\tau}(t)$ are certainly not orthogonal since they are very redundant (they are defined for continuously varying a and τ) but this reconstruction formula is indeed satisfied whenever $h(t)$ is of finite energy and *band pass* (which implies that it oscillates in time like a short wave, hence the name "wavelet"). More precisely, if $h(t)$ is assumed sufficiently regular, the reconstruction condition is $\int h(t)\, dt = 0$.

Note that the reconstruction takes place only in the energetic sense: for example, a signal may reconstructed only with zero mean since $\int h(t)\, dt = 0$. In fact the type of convergence of (6) may be straightened and is related to the numerical robustness of the reconstruction [DAU90a]. Another reconstruction formula, due to Morlet [GOU84], [GRO89]

$$x(t) = Cst \int \mathrm{CWT}(t,a) \frac{da}{a^{3/2}}$$

is often preferred in actual implementation since it requires only one integration.

Similar considerations can be done for the STFT, and the similarity is remarkable [DAU90a], [FLA90], [RIO90a]. However in the STFT case, the reconstruction condition is less restrictive: only finite energy of the window is required.

IV. SCALOGRAMS

Since the CWT behaves like an orthonormal basis decomposition, it can be shown that it is isometric [GRO84], i.e. it preserves energy:

$$\iint |\mathrm{CWT}(\tau,a)|^2 \frac{d\tau\, da}{a^2} = \text{Energy of signal } x = \int |x(t)|^2 dt \tag{7}$$

Therefore, the *scal‧ogram*, that is the squared modulus of the CWT, appears to be a distribution of the signal's energy in the time-scale plane (associated with measure $\frac{d\tau\, da}{a^2}$, and thus expressed in power per frequency unit.)

The main advantage of the energetic representation is that it produces an easily interpretable visual 2D representation of signals [GRO89]; each time-scale pattern in the time-scale plane contributes for the global energy of the signal. However an energetic representation has some disadvantages, too. The scalogram cannot be, in general, inverted: phase information is necessary to reconstruct the signal. Moreover it has been shown on some examples [GRO89] that the phase representation more accurately reveals isolated, local bursts in a signal than the energy representation does (see Box 3.) Also, since the scalogram is bilinear in the analysed signal, cross-terms appear as interferences between patterns in the time-scale plane and this may be undesirable. For example, the basic wavelet is generally chosen to be complex analytic so as to avoid interferences between negative and positive frequencies.

Of course similar derivations can be done for the STFT, leading to the well known *spectrogram* (square modulus of the STFT.) Note that in the wavelet case, the energy of the signal is distributed with different resolutions according to Fig. 2.(b).

In particular, Fig. 4.(a) shows that the influence of the signal's behavior around $t = t_0$ on the analysis is limited to a cone in the time-scale plane; it is therefore very "localized" around t_0 for small scales. In the STFT case, the corresponding region of influence would be as large as the extent of the analysing window, whatever the analyzed frequency (see Fig. 4.(b).) Moreover, the time-scale analysis being logarithmic in frequency, the area of influence of some pure frequency f_0 in the analyzed signal will narrow as f_0 increases (see Fig. 4.(c)-(d).)

Fig. 5 show some examples of spectrograms and scalograms for synthetic and speech signals (see Box 3.) Also, more involved energy representations can be developped for both time-frequency and time-scale (see Box 4.)

V. Wavelet Frames and Orthonormal Bases

V. 1. Discretization of Time-Scale Parameters.

In section III we have seen that the continuously labelled basis functions (wavelets) $h_{a,\tau}(t)$ behave in the wavelet analysis and synthesis just like an orthonormal basis. A natural question arises [MEY89]: if we suitably discretize the time-scale parameters a, τ, and for suitable choices of the basic wavelet $h(t)$, can we obtain a *true* orthonormal basis?

There is a natural way of discretizing the time-scale parameters a, τ [DAU90a]: since two scales $a_0 < a_1$ roughly correspond to two frequencies $f_0 > f_1$, the wavelet coefficients at scale a_1 can be subsampled at $(f_0/f_1)^{\text{th}}$ the rate of the coefficients at scale a_0, according to Nyquist's rule. We therefore choose to discretize the time-scale parameters on the sampling grid drawn in Fig. 7, that is, we have $a = a_0^j$ and $b = k\, a_0^j T$, where j and k are integers. The corresponding wavelets and wavelet coefficients are $h_{j,k}(t) = a_0^{-j/2} h(a_0^{-j} t - kT)$ and $c_{j,k} = \int x(t)\, h_{j,k}^*(t)\, dt$, respectively.

An analogy is the following: assume your wavelet analysis is like a microscope. First you choose the magnification, that is, a_0^{-j}. Then you move to the location of your choice. Now, obviously, if the magnification is large (that is, you are looking at small details), you want to move by small steps $a_0^j T$ in order to catch details.

The problem is to find a_0, T, and $h(t)$ such that the sum of projections

$$\sum_j \sum_k c_{j,k}\, h_{j,k}(t) \tag{8}$$

reasonably reconstructs the signal $x(t)$. Evidently when a_0 is close enough to 1 (and if T is small enough) the wavelet functions are overcomplete, signal reconstruction by (8) is still very close to (6) and will take place within non-restrictive conditions on $h(t)$. On the other hand, if the sampling is sparse, e.g. the computation is done octave by octave (a_0 = 2) a true orthonormal basis will be obtained only for very special $h(t)$'s [DAU90a], [MEY90].

V. 2. Wavelet Frames.

The theory of wavelet frames was derived by Daubechies [DAU90a]. It provides a general framework which covers the two extreme situations just mentionned. It therefore permits to balance 1) *redundancy*, i.e. sampling density in Fig. 7, and 2) restrictions on $h(t)$ for the reconstruction scheme to work. The less redundancy, the tighter these restrictions but the fewer computed wavelet coefficents, which is more adapted to coding schemes.

The theory of frames [DUF52] is based on the assumption that the linear correspondance $x(t) \rightarrow c_{j,k}$ is bounded, with bounded inverse. The family of wavelet functions is then called a *frame* and is such that the energy of the wavelet coefficients (sum of the square modules) relative to that of the signal lies between two positive "frames bounds" A and B. These frame bounds can be computed from a_0, T and $h(t)$ using Daubechies' formulae [DAU90a]. What is interesting is that they govern the accurateness of signal reconstruction by (8). More precisely, we have

$$x(t) \approx \frac{2}{A+B} \sum_j \sum_k c_{j,k}\; h_{j,k}(t)$$

with relative SNR greater than $(B/A+1)/(B/A-1)$ (see Fig.8.) The closer B/A is to one, the more accurate the reconstruction. It may happen that A=B ("tight frame"), in which case the wavelets behaves exactly like an orthonormal basis, although they may not even be linearly independent! The reconstruction formula can also be made exact in the general case if one uses different synthesis functions h'_{jk} (which constitute the *dual frame* of the h_{jk}'s [DAU90a].)

V. 3. Introduction to orthogonal wavelet bases.

If a tight frame is such that all wavelets $h_{j,k}$ are necessary to reconstruct the signal (i.e., if one is removed, a general signal cannot be synthesized), then the wavelets form an orthonormal basis of the space of signals with finite energy [HEI90]. Orthonormality, of course, means

$$\int h_{j,k}(t)\, h^{*}_{j',k'}(t)\, dt = \begin{cases} 1 \text{ if } j = j', k = k' \\ 0 \text{ otherwise} \end{cases}$$

Again this can be extended to synthesis functions $h'_{jk} \neq h_{jk}$, leading to so-called *biorthogonal* wavelet bases [COH90], [VET 90a], [VET90b] Such schemes are best understood when one turns to the discrete-time cas:. This is the subject of the next section, which focuses on the orthogonal case.

VI. The Discrete Time Case

In the discrete time case, two methods were developed independently in the late seventies and early eighties which lead naturally to discrete wavelet transforms, namely subband coding [CRI76], [CRO76], [EST77] and pyramidal coding or multiresolution signal analysis [BUR83]. The methods were proposed for coding, and thus, the notion of critical sampling (of requiring a minimum number of samples) was of importance. Pyramid coding actually uses some oversampling, but because it has an easier intuitive explanation, we will describe it first.

VI.1 The Multiresolution Pyramid

Given an original sequence $x(n)$, $n \in \mathbf{Z}$, we derive a lower resolution approximation at a lower scale $y(n)$ by lowpass filtering (with a filter having impulse response $h(n)$ and subsampling by two (that is, dropping every other sample):

$$y(k) = \sum_{k=-\infty}^{+\infty} g(k)\, x(n-2k)$$

The resolution change is obtained by the lowpass filter (loss of high frequency detail). The scale change is due to the subsampling by two, since a shift by two in the original signal $x(n)$ results in a shift by one in $y(n)$.

Now, based on this lowpass and subsampled version of $x(n)$, we try to find an approximation $a(n)$ to the original. This is done by first upsampling $y(n)$ by two (that is, inserting a zero between every sample) since we need a signal at the original scale for comparison.

$$y'(2n) = y(n), \quad y'(2n+1) = 0$$

Then, $y'(n)$ is interpolated with a filter with impulse response $g'(n)$ to obtain the approximation $a(n)$:

$$a(n) = \sum_{k=-\infty}^{\infty} g'(k)\, y'(n-k)$$

Note that if $g(n)$ and $g'(n)$ were perfect halfband filters (having transmission 1 from $-\pi/2$ to $\pi/2$ and 0 elsewhere), then the Fourier transform of $a(n)$ would be equal to the one of $x(n)$ over the range $(-\pi/2, \pi/2)$ while being zero elsewhere. That is, $a(n)$ would be a perfect halfband lowpass approximation to $x(n)$.

Of course, in general $a(n)$ is not going to be equal to $x(n)$ (in the previous example, $x(n)$ would have to be a halfband signal). Therefore, we compute the difference between $a(n)$ (our approximation based on $y(n)$) and $x(n)$:

$$d(n) = x(n) - a(n)$$

It is obvious that $x(n)$ can be reconstructed based on $d(n)$ and $a(n)$ (that is, $y(n)$) by simply adding them, and the whole process is shown in Fig. 9. However, there has to be some redundancy, since a signal with sampling rate f_s is mapped into two signals $d(n)$ and $y(n)$ with sampling rates f_s and $f_s/2$, respectively.

However, the separation of the original signal $x(n)$ into a coarse approximation $a(n)$ plus some additional detail contained in $d(n)$ is conceptually important. Because of the resolution change involved (lowpass filtering followed by subsampling by two produces a signal with half the resolution at half the scale of the original), the above method and related ones are part of what is called multiresolution signal analysis [ROS84] in computer vision.

The scheme can be iterated on $y(n)$, creating a hierarchy of lower resolution signals at lower scales. Because of that hierarchy and the fact that signals become shorter and shorter (or images smaller and smaller), such schemes are called signal or image pyramids [BUR83].

VI.2 Subband Coding Schemes

We have seen that the above system creates a redundant set of samples. This redundancy can be removed if the filters $g(n)$ and $g'(n)$ meet certain conditions (see Box 5), but we will look at a different scheme instead, where no such redundancy appears. It is the so-called subband coding scheme first popularized in speech compression. The lowpass, subsampled approximation is obtained exactly as explained above, but, instead of a difference signal, we compute the "added detail" as a highpass filtered version of $x(n)$ (using a filter with impulse response $h(n)$), followed by subsampling by two. Intuitively, it is clear that the "added detail" to the lowpass approximation has to be a highpass signal, and it is obvious that if $g(n)$ is an ideal halfband lowpass filter, then an ideal halfband highpass filter $h(n)$ will lead to a perfect representation of the original signal into two subsampled versions.

This is exactly one step of a wavelet decomposition using sin(x)/x filters, since the original signal is mapped into a lowpass approximation (at half the scale) and an added detail signal (also at half the scale.) In particular, using these ideal filters, the discrete version is identical to the continuous wavelet transform.

What is more interesting is that it is not necessary to use ideal (that is, impractical) filters, and yet $x(n)$ can be recovered from its two filtered and subsampled versions which we now call $y_0(n)$ and $y_1(n)$. To do so, both are upsampled and filtered by $g'(n)$ and $h'(n)$ respectively, and finally added together, as shown in Fig. 10. Now, unlike the pyramid case, the reconstructed signal (which we now call $\hat{x}(n)$) is not identical to $x(n)$, unless the filters meet some specific constraints, which are referred to as the perfect reconstruction property.

The easiest case to analyze appears when the analysis and synthesis filters in Fig. 10 are identical (within time-reversal) and that perfect reconstruction is achieved (that is, $\hat{x}(n) = x(n)$, maybe within a shift.) Then it can be shown that the subband analysis/synthesis corresponds to a decomposition onto an orthonormal basis, followed by a reconstruction which amounts to summing up the orthogonal projections. We will assume FIR filters in the following. Then, it turns out that the highpass and lowpass filters are related by:

$$h(L-1-n) = (-1)^n g(n) \tag{9}$$

where L is the filter length (which has to be even). Note that the modulation by $(-1)^n$ transforms indeed the lowpass filter into a highpass one.

Now, the filter bank in Fig. 10, which computes convolutions followed by subsampling by two, evaluates inner products of the sequence $x(n)$ and the sequences $\{g(-n+2k), h(-n+2l)\}$ (the time reversal comes from the convolution, which reverses one of the sequences). Thus:

$$y_0(k) = \langle g(-n+2k) \mid x(n) \rangle$$

$$y_1(k) = \langle h(-n+2l) \mid x(n) \rangle$$

Because the filter impulse responses form an orthonormal set, it is very simple to reconstruct $x(n)$ as:

$$x(n) = \sum_{k=-\infty}^{\infty} \left[y_0(k)\, h(-n+2k) + y_1(k)\, g(-n+2k) \right] \qquad (10)$$

that is, as a weighted sum of the orthogonal impulse responses, where the weights are the inner products of the signal with the impulse responses. This is of course the standard expansion of a signal into an orthonormal basis, where the resynthesis is the sum of the orthogonal projections.

From (10), it is also clear that the synthesis filters are identical to the analysis filters within time reversal.

Such perfect reconstruction filter banks have been studied in the digital signal processing literature, and the orthonormal decomposition we just indicated is usually referred to as a "paraunitary" or "lossless" filter bank [VAI89]. An interesting property of such filter banks is that they can be written in lattice form [VAI88], and that the structure and properties can be extended to more than two channels [VAI89], [VET89].

VI.3 The Discrete Wavelet Transform

We have shown how to decompose a sequence $x(n)$ into two subsequences at half rate, or half resolution, and this by means of "orthogonal" filters (orthogonal with respect to even shifts). Obviously, this process can be iterated on either or both subsequences. In particular, to achieve finer frequency resolution at lower frequencies (as obtained in the continuous wavelet transform), we iterate the scheme on the lower band only. If $g(n)$ is a good halfband lowpass, $h(n)$ is a good halfband highpass (by (9)), and one iteration of the scheme creates a lowband that corresponds to the lower quarter of the frequency spectrum. Each further iteration halves the width of the lowband (increases its frequency resolution by two), but due to the subsampling by two, its time resolution is halved as well. At each iteration, the current high band corresponds to the difference between the previous lowband and the current one, that is a passband. Schematically, this is equivalent to Fig. 11, and the frequency resolution is as in Fig. 2.(b).

An important feature of this discrete algorithm is its relatively low complexity. Actually, the following somewhat surprising result holds: independently of the depth of the tree in Fig. 11, the complexity is linear in the number of input samples, with a constant factor that depends on the length of the filter. The proof is straightforward. Assume the computation of the first filter bank requires C_0 operations per input sample (C_0 is typically of the order of L). Then, the second stage requires also C_0 operations per sample of its input, but, because of the subsampling by two, this amounts to $C_0/2$ operations per sample of the input si.jnal. Therefore, the total complexity is bounded by:

$$C_{total} = C_0 + \frac{C_0}{2} + \frac{C_0}{4} + \ldots \quad < 2C_0$$

which demonstrates the efficiency of the discrete wavelet transform algorithm and shows that it is independent of the number of octaves that one computes [RIO90c].

VI.4 Iterated Filters and Regularity

There is a major difference between the discrete scheme we just saw and the continuous time wavelet transform. In the discrete time case, the role of the wavelet is played by the highpass filter $h(n)$ and the cascade of subsampled lowpass filters followed by a highpass filter (which amounts to a bandpass filter). These filters, which correspond roughly to octave band filters, unlike in the continuous wavelet transform, are not exact scaled versions of each other. In particular, since we are in discrete time, scaling is not as easily defined, since it involves interpolation as well as time expansion.

Nonetheless, under certain conditions, the discrete system converges (after a certain number of iterations) to a system were subsequent filters are scaled versions of each other. Actually, this convergence is the basis for the construction of continuous time compactly supported wavelet bases [DAU88].

It will be convenient to use z-transforms of filters, e.g. $G(z) = \sum_n g(n)\, z^{-n}$ in the following.

Now, we would like to find the equivalent filter that corresponds to the lower branch in Fig. 11, that is the iterated lowpass filter. It can be easily checked that subsampling by two followed by filtering with $G(z)$ is equivalent to filtering with $G(z^2)$ followed by the subsampling (z^2 inserts zeros between samples ot the impulse response, which are removed by the subsampling). That is, the first two steps of lowpass filtering can be replaced by a filter with z-tranform $G(z).G(z^2)$, followed by subsampling by 4. More generally, calling $G^i(z)$ the equivalent filter to i stages of lowpass filtering and subsampling by two (that is, a total subsampling by 2^i), we get:

$$G^i(z) = \prod_{l=0}^{i-1} G(z^{2^l}) \tag{11}$$

Call its impulse response $g^i(n)$.

As i infinitely increases, this filter becomes infinitely long. Instead, consider a function $f^i(x)$ which is piecewise constant on intervals of length $1/2^i$ and has value $2^{i/2}\, g^i(n)$ in the interval $[n/2^i, (n+1)/2^i]$. That is, $f^i(x)$ is a staircase function with the value given by the samples of $g^i(n)$ and intervals which decrease as 2^{-i}. It can be verified that the function is supported on the interval $[0, L\text{-}1]$. Now, for i going to infinity, $f^i(x)$ can converge to a continuous function, or a function with finitely many discontinuities or even a fractal function. Fig. 12 shows two examples, one where $f^\infty(x)$ is continuous and another where $f^\infty(x)$ has fractal behavior.

I. Daubechies gave a sufficient condition so that the iterated function converges to a continuous function. In essence, the filter $G(z)$ must have a sufficient number of zeros at $z = -1$, or half sampling frequency, so as to attenuate repeat spectras. Based on this sufficient condition, one can construct filters which are both orthogonal and converge to a continuous function. Such filters are called regular, and examples can be found in [DAU88], [DAU90b], [RIO90b], [VET90] (see Box 6.)

VI.5 Scaling Functions and Wavelets Obtained from Iterated Filters

Call the final function to which $f^i(x)$ converges $g_c(x)$. Because it is the product of lowpass filters, it is itself lowpass and is called a "scaling function" because it is used to go from a fine scale to a coarser scale. Because of the product (11) from which the scaling function is derived, $g_c(x)$ satisfies the following two scale difference equation:

$$g_c(x) = \sum_{n=-\infty}^{\infty} g(n)\, g_c(2x-n) \tag{12}$$

and Fig. 13 shows two such examples. The second one is based on a filter which is regular and orthogonal to its even translates and was designed by I. Daubechies [DAU88].

So far, we have only discussed the iterated lowpass and its associated scaling function. However, from Fig. 10, it is clear that a bandpass filter is obtained in the same way, except for a final highpass filter. Therefore, similarly to (12), the wavelet $h_c(x)$ is obtained as:

$$h_c(x) = \sum_{n=-\infty}^{\infty} h(n)\, g_c(2x-n)$$

that is, it satisfies also a two scale equation. Now, if the filters $h(n)$ and $g(n)$ form an orthonormal set with respect to even shifts, then the functions $g_c(x\text{-}l)$ and $h_c(x\text{-}k)$ form an orthonormal set (see Box 7.)

Because they also satisfy two scale difference equations, it can be shown [DAU88] that the set $h_c(2^{-i}x\text{-}k)$, $i,k \in \mathbf{Z}$, forms an orthonormal basis for the set of square integrable functions $L^2(\mathbf{R})$. We have shown how regular filters can be used to generate continuous wavelet bases. The converse is also true, that is, orthonormal sets of scaling functions and wavelets can be used to generate perfect reconstruction filter banks.

CONCLUSION

We have seen that Short-Time Fourier Transforms and Wavelet Transforms represent alternative slicings of the time-frequency (or time-scale) plane. Two major advantages of the Wavelet Transforms are that they can zoom in to time discontinuities and that orthonormal bases can be constructed.

In the discrete case, the Wavelet Transform is equivalent to a logarithmic filter bank, with the added constraint of regularity on the lowpass filter.

The theory of wavelets can be seen as a common framework for techniques that had been developped independently in various fields. This conceptual unification furthers the understanding of the mechanisms involved, quantifies trade-offs, and points to new potential applications.

BOX 1: THE SCALE AND RESOLUTION NOTIONS

In order to understand what the CWT does in connection with the scale and time-resolution notions, consider the following dilation/contraction operation acting on the analysed signal $x(t)$:

$$x(t) \longrightarrow x(at)$$

When a decreases, the signal is progressively dilated so that we are able to see more and more details in its temporal waveform. This means that the *time-resolution* of the signal we see *increases* as a decreases (because the eye basically low-passes the signal $x(at)$.)

As a result, the *new* information that comes into the eye when resolution increases is something like a *band-pass* filtering of $x(at)$. But by making a change of variables in the definition (5) of the CWT, we indeed obtain a band-pass filtering of $x(at)$:

$$\mathrm{CWT}\{\tau, a\} = \sqrt{|a|} \int x(at)\ h(t - \frac{\tau}{a})\ dt$$

The CWT, as a function of τ, may therefore be interpreted as the infinitesimal *novelty of information* between two (very close) time-resolutions.

When the signal is dilated enough (for small a), no further new information comes into the eye and dilating is no more useful: the highest time-resolution of the signal itself has then been attained, and that corresponds to the lowest scale a of interest for analysing the signal.

Note that here the scale parameter a is not associated to the analysed signal , but rather to the wavelet $h(t/a)$ which analyses the signal at scale a in equation (5). The scale parameter is of course similar to the notion of scale used for maps: decreasing the scale means compressing the wavelet, which thereby analyses thin details on the signal. On the contrary, for very large scales, the wavelet is so stretched that it can see a large portion of the signal. In the above description, the proper scale of the signal $x(at)$ is in fact $1/a$.

BOX 2: STFTS AND CWTS AS AMBIGUITY FUNCTIONS

The inner product is often used as a similarity measurement, and because both STFT's and CWT's are inner products, they appear in several detection/estimation problem. Consider, for example, the problem of estimating the location and velocity of some target in radar or sonar applications. The estimation procedure consists in first emetting a known signal $h(t)$. In presence of a target, this signal will return to the source (received signal $x(t)$) with a certain delay τ, due to the target's location, and a certain distortion (Doppler effect) , due to the target's velocity.

For narrow-band signals, the Doppler effects amounts in a single frequency shift f_0 and the characteristics of the target will be determined by maximizing the cross correlation function (called "narrow-band ambiguity function") [WOO53]

$$\int x(t)h(t-\tau)\, e^{-2j\pi f_0 t}\, dt = \mathrm{STFT}(\tau,f)$$

For wide-band signals, however, the Doppler frequency shift varies in the signal's spectrum, causing a streching or a compression in the signal. The estimator thus becomes the "wide-band ambiguity function" [SPE67], [AUS90]

$$\frac{1}{\sqrt{a}}\int x(t)\, h\!\left(\frac{t-\tau}{a}\right) dt = \mathrm{CWT}_x(\tau,a)$$

As a result, in both cases, the "maximum likehood" estimator takes the form of a STFT or a CWT, i.e. of an inner product between the received signal and either STFT or wavelet basis functions. The basis function which best fits the signal is used to estimate the parameters.

Note that, although the wide-band ambiguity function is a CWT, for physical reasons, the dilation parameter a stays in the order of magnitude of 1, whereas it may cover several octaves when used in signal analysis.

BOX 3: SPECTROGRAMS AND SCALOGRAMS

We present spectrograms and scalograms for some synthetic and a real signal. The signals are of length 384, the STFT uses a Gaussian-like window of length $L = 128$ and the scalogram is obtained with a Morlet wavelet (a complex sinusoid windowed with a Gaussian) of length from 23 to 363. First, Fig. 5.1 shows the analysis of two Diracs and two sinusoids with Short-Time Fourier Transform and Wavelet Transform. Note how the Diracs are well time-localized at high frequencies in the scalogram. Fig. 5.2 shows the analysis of three starting sinusoids, and Fig. 5.3. shows the transforms of a chirp signal. Again, the transitions are well resolved at high frequencies in the scalogram. Finally, Fig. 5.4 shows the analysis of a stretch of speech signal, where the onset of voicing is clearly seen in both representations.

BOX 4: INCORPORATING THE STFT, CWT AND WIGNER-VILLE DISTRIBUTION INTO A COMMON CLASS OF TIMEFREQUENCY OR TIME-SCALE ENERGY REPRESENTATIONS

There has been considerable work in extending the spectrogram into more general time-frequency energy distributions TF(τ, f), which all have the basic property of distributing the signal's energy all over the time-frequency plane:

$$\iint \mathrm{TF}(\tau,f)\,d\tau\,df=\int |x(t)|^2\,dt$$

Among them, an alternative to the spectrogram for nonstationary signal analysis is the Wigner-Ville distribution [CLA80], [BOU85]:

$$W_x(\tau,f)=\int x(\tau+\frac{t}{2})\,x^*(\tau-\frac{t}{2})\,e^{-2j\pi ft}\,dt$$

which has been widely adressed in the literature. More generally, the whole class of time-frequency energy distributions have been fully described by Cohen [COH66]: they can all be seen as smoothed (or, more precisely, correlated) versions of the Wigner-Ville distribution. The spectrogram is itself recovered when the "smoothing" function is the Wigner-Ville distribution of the analysing window!

A similar situation appears for time-scale energy distributions. For example, the scalogram can be written as [FLA90], [RIO90a]

$$|CWT(\tau,a)|^2=\iint W_x(t,\nu)\,W_h(\frac{t-\tau}{a},a\nu)\,dt\,d\nu$$

i.e. as some 2D "affine" correlation (some say correlation on the affine group [GRO84]) between the signal and "basic" wavelet 's Wigner-Ville distributions. This remarkable formula tells us that there exists strong links between Wavelet transforms and Wigner-Ville distributions. And, as a matter of fact, it can be generalized to define the most general class of time-scale energy distributions [FLA90], [RIO90a], just as in the time-frequency case.

By progressively controlling gaussian smoothing functions, Fig. 6 shows that it is even possible to go continuously from the spectrogram of a given signal to its scalogram, provided that one passes by its Wigner-Ville distribution at the middle [FLA90], [RIO90a]. This property may allow us to decide whether or not we should choose time-scale analysis tools, rather than time-frequency ones for a given problem.

BOX 5: CRITICALLY SAMPLED PYRAMID

Because one stage of a pyramid decomposition leads to both a half rate low resolution signal and a full rate difference signal, we have an increase in the number of samples by 50%. This oversampling can be avoided, in which case the systern reduces to a subband coding system.

To see this, note that the upper branc ı of the pyramid system of Fig. 9 is equivalent to the lowpass branch (analysis and synthesis) of the subband system of Fig. 10. If the lowpass filter is part of a perfect reconstruction filter bank, the difference signal is equal to the highpass branch of a subband system (since added to the lowpass brarıch, it would yield perfect reconstruction). That is, using the notation H (resp. G) to denote the operator corresponding to lowpass filtering with $h(n)$ (resp. $g(n)$) followed by subsampling by two, and $\mathbf{H}^*$, $\mathbf{G}^*$ their conjugate transposes (which correspond to upsampling by two and interpolating with the conjugate filter), we have (see Fig. 10):

$$d = (\mathbf{I} - \mathbf{G}^*\mathbf{G})(x) = \mathbf{H}^*\mathbf{H}(x) \tag{1}$$

Because the filter impulse responses form an orthonormal set with respect to even shifts when we have an orthogonal filter bank, we have:

$$\mathbf{H}\mathbf{H}^* = \mathbf{I}$$

Thus, filtering d by $h_1(n)$ and subsampling by two leads to:

$$\mathbf{H}(d) = \mathbf{H}\mathbf{H}^*\mathbf{H}(x) = \mathbf{H}(x)$$

that is, exactly equal to the highpass analysis output of a subband coder. From this subsampled version of x, we can recover d by (1) and thus perfectly reconstruct the input, showing that a critically sampled pyramid scheme is just a subband coding system. Note that we assume linear processing throughout. If non-linear processing is involved (like quantization), the oversampled nature of the pyramid can actually lead to greater robustness.

BOX 6: REGULAR SCALING FILTERS

The regularity property of wavelet filters (see section IV. 4) has recently attracted attention to the digital signal processing community, especially for image coding applications [ANI90], [MAL89b]. In fact, it is well known that the structure of computations in a Discrete Wavelet Transform and in an octave-band filter bank are identical. Therefore, besides the different views and interpretations that have been given on them, the main difference lies in the filter design: wavelet filters are chosen so as to be regular. Recall that this means, with the notation of section IV.4 and 5, that the piecewise constant function associated to the discrete wavelet sequence $h_j(n)$ of z-transform $G^j(z)\ H(z^{2^j})$ converges (e.g. pointwise), as j indefinitely increases, to a regular limit function $h_c(x)$. By "regular" we mean that $h_c(x)$ is at least continuous, or better, once or twice continuously differentiable. The regularity order is the number of times $h_c(x)$ is continuously differentiable.

In practise the convergence is very fast (the discrete wavelets are almost undistinguishable from their analog counterparts after 3 to 7 iterations - it can be shown that the more regular the limit function, the faster the convergence to this limit [DAU90], [RIO90d]). This justifies the study of the limit $h_c(x)$ which is, in practice, attained after a few octaves. Since an error in a wavelet coefficient results, after reconstruction, in an overall error proportional to a discrete wavelet $h_j(n)$, regularity seems a nice property e.g. to avoid visible distortion on a reconstructed image [ANI90].

From equation (9) and (11), the knowledge of $g(n)$ suffices to determine the limit $h_c(x)$. Several methods have been developped to estimate the regularity order of the limit knowing the coefficients $g(n)$, most of which being based on Fourier transform techniques. Recently, a time-domain technique has been developped which provides optimal estimates [DAU90c], [RIO90d].

It can be shown that in order the limit to be N times continuously differentiable, the z-transform $G(z)$ must have at least $N+1$ zeros at $z=-1$. This can be interpretated as a flatness condition on the spectrum at half sampling frequency. As a result, maximally regular wavelet filters are very close to, but different from [DAU90b] maximally flat filters [HER71]. It is still not clear whether maximally flat or maximally regular filters are most adapted to coding schemes [AKA90], [ANI90]. The minimal regularity order necessary for good coding performance of discrete wavelet transform schemes, if needed at all, is also not known and remains a topic for future investigation.

BOX 7: MULTIRESOLUTION ANALYSIS

The concept of multiresolution approximation of functions was introduced by Y.Meyer and S.Mallat and provides a powerful framework to understand wavelet decompositions. The basic idea is that of successive approximation, together with that of "added detail" as one goes from one approximation to the next, finer one. A formal definition can be found in [MAL89a], [MAL89c], [MEY90], and we will only give the intuition behind the construction.

Assume we have a ladder of spaces such that:

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots$$

with the property that if $f(x) \in V_i$, then $f(x-2^{-i}k) \in V_i$, $k \in \mathbf{Z}$, and $f(2x) \in V_{i-1}$. Call W_i the orthogonal complement of V_i in V_{i-1}. Therefore:

$$V_{i-1} = V_i \overset{\perp}{\oplus} W_i \tag{1}$$

Thus, W_i contains the "added detail" necessary to go from the resolution V_i to V_{i-1}. Iterating (1), it is clear that:

$$V_{i-1} = W_i \overset{\perp}{\oplus} W_{i+1} \overset{\perp}{\oplus} W_{i+2} \overset{\perp}{\oplus} W_{i+3} \overset{\perp}{\oplus} \cdots \tag{2}$$

that is, a given resolution can be attained by a sum of added details.

Now, assume we have an orthonormal basis for V_0 made up of a function $g_e(x)$ and its integer translates. Because $V_0 \in V_{-1}$, $g_e(x)$ can be written in terms of the basis in V_{-1}:

$$g_e(x) = \sum_n c_n \; g_e(2x-n)$$

Then it can be verified that the function:

$$h_e(x) = \sum_n (-1)^n c_{n+1} \; g_e(2x-n)$$

and its integer translates form an orthonormal basis for W_0. Because of (2), it can be shown that $h_e(x)$ and its scaled and translated versions form a wavelet basis of $L^2(\mathrm{R})$ [MAL89a], [MAL89c], [MEY90].

The multiresolution idea is now very intuitive. Assume we have an approximation of a signal at a resolution corresponding to V_0. Then a better approximation is obtained by adding the details corresponding to W_0, that is, the projection of the signal in W_0 which amounts to a weighted sum of wavelets at that scale. Thus, by iterating this idea, a signal in $L^2(\mathbf{R})$ can be seen as the successive approximation or weighted sum of wavelets at finer and finer scale.

REFERENCES

[AKA90] A.N. Akansu, R.A. Haddad, and H. Caglar, "The Binomial QMF-Wavelet Transform for Multiresolution Signal Decomposition," submitted *IEEE Trans. Signal Proc.*, 1990.

[ALL77] J.B. Allen and L.R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, Vol. 65, No. 11, pp. 1558-1564, 1977.

[ANI90] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Vector Quantization in the Wavelet Transform Domain," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, Apr.3-6, 1990, pp. 2297-2300.

[AUS90] L. Auslander and I. Gertner, "Wide-Band Ambiguity Function and *a.x+b* group," in *Signal Processing, Part I: Signal Processing Theory*, L. Auslander, T. Kailath, S. Mitter eds., Institute for Mathematics and its Applications, Vol. 22, Springer Verlag, New York, pp.1-12, 1990.

[BOU85] G. F. Boudreaux-Bartels, "Time-Varying Signal Processing Using the Wigner Distribution Time-Frequency Signal Representation," in *Adv. in Geophysical Data Proc.*, Vol. 2, pp. 33-79, Jai Press Inc., 1985.

[BUR83] P.J.Burt and E.H.Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. on Com.*, Vol. 31, No.4, pp.532-540, April 1983.

[BUR89] P.J. Burt, "Multiresolution Techniques for Image Representation, Analysis, and 'Smart' Transmission," *Proc. SPIE Conf. on Visual Communication and Image Processing.*, pp. 2-15, Philadelphia, PA, Nov. 1989.

[CAL64] A. Calderón, "Intermediate Spaces and Interpolation, the Complex Method," *Studia Math.*, Vol. 24, pp. 113-190, 1964.

[CIP90] B.A. Cipra, "A New Wave in Applied Mathematics," *Science*, Research News, Vol. 249, August 24, 1990.

[CLA80] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis. Part I: Continuous-Time Signals," *Philips J. Res.*, Vol. 35, No. 3, pp.217-250, 1980.

[COH66] L. Cohen, "Generalized Phase-Space Distribution Functions," *J. Math. Phys.*, Vol. 7, No. 5, pp. 781-786, 1966.

[COH90] A. Cohen, I. Daubechies and J.C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets," preprint, 1990.

[CRI76] A.Croisier, D.Esteban, and C.Galand, "Perfect Channel Splitting by Use of Interpolation, Decimation, Tree Decomposition Techniques," Int. Conf. on Information Sciences/Systems, Patras, pp. 443-446, Aug. 1976.

[CRO76] R.E.Crochiere, S.A.Weber, and J.L.Flanagan, "Digital Coding of Speech in Subbands," Bell Syst. Tech. J., Vol.55, pp.1069-1085, Oct.1976.

[CRO83] R.E.Crochiere, and L.R.Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[DAU88] I.Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. in Pure and Applied Math.*, Vol.41, No.7, pp.909-996, 1988.

[DAU90a] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. on Info. Theory*, Vol. 36, No.5, pp.961-1005, Sept. 1990.

[DAU90b] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme," preprint AT&T, 1990.

[DAU90c] I. Daubechies and J.C. Lagarias, "Two-Scale Difference Equations II. Local Regularity, Infinite Products of Matrices and Fractals," submitted *SIAM J. Math. Anal*, 1990.

[DUF52] R.J. Duffin and A.C. Schaeffer, "A Class of Nonharmonic Fourier Series," *Trans. Am. Math. Soc.*, Vol. 72, pp. 341-366, 1952.

[EST77] D. Esteban and C. Galand, "Application of Quadrature Mirror Filters to Split-Band Voice Codir g Schemes," *Int. Conf. Acoust., Speech, Signal Proc.*, Hartford, Connecticut, pp. 191-195, May 1977.

[FLA89] P.Flandrin, "Some Aspects of Non-Stationary Signal Processing with Emphasis on Time-Frequency and Time-Scale Methods," in [WAV89], pp.68-98, 1989.

[FLA90] P. Flandrin and O. Rioul, "Wavelets and Affine Smoothing of the Wigner-Ville Distribution," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, April 3-6, 1990, pp. 2455-2458.

[FOU88] J.Fourier, "Théorie Analytique de la Chaleur," [in French], in *Oeuvres de Fourier*, tome premier, G.Darboux, Ed., Paris: Gauthiers-Villars, 1888.

[FRA28] P.Franklin, "A Set of Continuous Orthogonal Functions," *Math. Annal.*, Vol.100, pp.522-529, 1928.

[FRA86] M. Frazier and B. Jawerth, "The ϕ-Transform and Decomposition of Distributions," *Proc. Conf. Function Spaces and Appl.*, Lund 1986, Lect. Notes Math., Springer.

[GAB46] D.Gabor, "Theory of Communication," J. of the IEE, Vol.93, pp.429-457, 1946.

[GOU84] P.Goupillaud, A.Grossmann and J.Morlet, "Cycle-Octave and Related Transforms in Seismic Signal Analysis," *Geoexploration*, Vol.23, pp.85-102, Elsevier Science Publishers, B.V. Amsterdam, Netherlands, 1984/85.

[GRO84] 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.

[GRO89] A. Grossmann, R. Kronland-Martinet, and J. Morlet, "Reading and Understanding Continuous Wavelet Transforms," in [WAV89], pp.2-20, 1989.

[HAA10] A.Haar, "Zur Theorie der Orthogonalen Funktionensysteme," [in German] *Math. Ann.*, Vol. 69, pp. 331-371, 1910.

[HEI90] C.E. Heil, "Wavelets and Frames," in *Signal Processing, Part I: Signal Processing Theory*, L. Auslander, T. Kailath, S. Mitter eds., Institute for Mathematics and its Applications, Vol. 22, Springer Verlag, New York, pp.147-160, 1990.

[HER71] O. Herrmann, "On the Approximation Probem in NonRecursive Digital Filter Design," *IEEE Trans. Circuit Theory*, Vol. CT-18, No. 3, pp. 411-413, May 1971.

[LIT37] J. Littlewood and R. Paley, "Theorems on Fourier Series and Power Series," *Proc. London Math. Soc.*, Vol.42, pp. 52-89, 1937.

[MAL89a] S.Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Trans. on Pattern Analysis and Machine Intell*, Vol.11, No. 7, pp.674-693, July 1989.

[MAL89b] S.Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. 37, No.12, pp.2091-2110, December 1989.

[MAL89c] S.Mallat, "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$,", *Trans. Amer. Math. Soc.*, Vol.315, No.1, pp.69-87, September 1989.

[MEY89] Y. Meyer, "Orthonormal Wavelets," in [WAV89], pp. 21-37, 1989.

[MEY90] Y.Meyer, *Ondelettes et Opérateurs, Tome I*, [in French] Herrmann ed., Paris, 1990.

[MIN85] F.Mintzer, "Filters for Distortion-Free Two-Band Multirate Filter Banks," *IEEE Trans. on Acoust., Speech, Signal Proc.*, Vol.33, pp.626-630, June 1985.

[PAU85] T.Paul, "Affine Coherent States and the Radial Schrödinger Equation I. Radial Harmonic Oscillator and the Hydrogen Atom," to appear in *Ann. Inst. H.Poincaré*.

[RIO90a] O. Rioul and P. Flandrin, "Time-Scale Energy Distributions: A General Class Extending Wavelet Transforms," submitted to *IEEE Trans. Signal Proc.*, May 1990. Rapport ICPI TS-8910, URA 346 CNRS, 25, rue du Plat, 69288 Lyon Cedex 02.

[RIO90b] O.Rioul, "A Unifying Multiresolution Theory for the Discrete Wavelet Transform, Regular Filter Banks and Pyramid Transforms," submitted to *IEEE Trans. Signal Proc.*, June 1990.

[RIO90c] O. Rioul and P. Duhamel, "Structures and Fast Algorithms for Implementing Wavelet Transforms," preprint.

[RIO90d] O. Rioul, "A Simple, Optimal Regularity Estimate for Wavelets," preprint.

[ROS84] A.Rosenfeld, Ed, *Multiresolution Techniques in Computer Vision*, Springer-Verlag, New York 1984.

[SMI86] M.J.T.Smith and T.P.Barnwell, "Exact Reconstruction for Tree-Structured Subband Coders," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. ASSP-34, pp. 434-441, June 1986.

[SMI87] M.J.T.Smith and T.P.Barnwell, "A New Filter Bank Theory for Time-Frequency Representation," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. ASSP-35, No.3, March 1987, pp. 314-327.

[SPE67] J.M. Speiser, "Wide-Band Ambiguity Functions," *IEEE Trans. on Info. Theory*, pp. 122-123, 1967.

[VAI88] P.P.Vaidyanathan and P.-Q. Hoang, "Lattice Structures for Optimal Design and Robust Implementation of Two-Band Perfect Reconstruction QMF Banks," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. ASSP-36, No. 1, pp.81-94, Jan. 1988.

[VAI89] P. P. Vaidyanathan and Z. Doganata, "The Role of Lossless Systems in Modern Digital Signal Processing," *IEEE Trans. Education, Special issue on Circuits and Systems*, Vol. 32, No.3, Aug. 1989, pp.181-197.

[VET89] M.Vetterli and D. Le Gall, "Perfect Reconstruction FIR Filter Banks: Some Properties and Factorizations," *IEEE Trans. on Acoust., Speech Signal Prpc.*, Vol.37, No.7, pp.1057-1071, July 1989.

[VET90a] M.Vetterli and C.Herley "Wavelets and Filter Banks: Relationships and New Results," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, pp. 1723-1726, Apr. 3-6, 1990.

[VET90b] M. Vetterli and C. Herley, "Wavelets and Filter Banks: Theory and Design," submitted *IEEE Trans. on Signal Proc.*, Aug. 1990.

[WAV89] *Wavelets, Time-Frequency Methods and Phase Space*, Proc. Int. Conf. Marseille, France, Dec. 14-18, 1987, J.M. Combes, A. Grossmann, P. Tchamitchian eds., Inverse Problems and Theoretical Imaging, Springer Verlag Berlin Heidelberg, 315 p., 1989.

[WOO53] P.M. Woodward, *Probability and Information Theory with Application to Radar*, Pergamon Press, London, 1953.

[YOU78] J.E. Younberg, S.F. Boll, "Constant-Q Signal Analysis and Synthesis," *IEEE Int. Conf. on Acoust., Speech, and Signal Proc., ICASSP-78*, Tulsa, OK, pp. 375-378, 1978.

EXTENDED REFERENCES

History of Wavelets: see [HAA09], [FRA28], [LIT37], [CAL64], [YOU78], [GOU84], [GRO84].

Books on Wavelets: see [WAV89], [MEY90] and

I. Daubechies, *Wavelets*, Lecture Notes, Lowell University, CBMS, SIAM publ., to appear.

Wavelets and their Applications, R.R. Coifman, I. Daubechies, S. Mallat, Y. Meyer scientific eds., L.A. Raphael, M.B. Ruskai managing eds., Academic Press, to appear.

Tutorials on Wavelets (see also [FLA89], [GRO89], [MAL89b])

R.R. Coifman, "Wavelet Analysis and Signal Processing," in *Signal Processing, Part I: Signal Processing Theory*, L. Auslander, T. Kailath, S. Mitter eds., Institute for Mathematics and its Applications, Vol. 22, Springer Verlag, New York, 1990.

C.E. Heil and D.F. Walnut, "Continuous and Discrete Wavelet Transforms," *SIAM Review*, Vol. 31, No. 4, pp 628-666, Dec. 1989.

Y. Meyer, S. Jaffard, O. Rioul, "L'Analyse par Ondelettes," [in French]*Pour La Science*, No.119, pp.28-37, Sept 1987.

G.Strang, "Wavelets and Dilation Equations: A Brief Introduction," *SIAM Review*, Vol. 31, No. 4, pp. 614-627, Dec. 1989.

Mathematics, Mathematical Physics (see also [GRO84], [DAU88], [MEY90])

P.G. Lemarié and Y. Meyer, "Ondelettes et Bases Hilbertiennes," [in French] *Revista Matematica Iberoamericana*, Vol.2, No.1&2, pp.1-18, 1986.

G.Battle, "A Block Spin Construction of Ondelettes, II. The Quantum Field Theory (QFT) Connection," *Comm. Math. Phys.*, Vol. 114, pp. 93-102, 1988.

Regular Scaling Filters (see also [DAU88], [DAU90b], [DAU90c], [RIO90c])

S.Dubuc, "Interpolation Through an Iterative Scheme," *J. Math. Analysis Appl.*, Vol.114, pp.185-204, 1986.

Numerical Analysis

G.Beylkin, R.Coifman and V.Rokhlin, "Fast Wavelet Transforms and Numerical Algorithms. I," submitted, Dec. 1989.

R.R. Coifman, "Multiresolution Analysis in Nonhomogeneous Media," in [WAV89], pp. 259-262, 1989.

V.Perrier, "Toward a Method to Solve Partial Differential Equations Using Wavelet Bases," in [WAV89], pp. 269-283, 1989.

Quantum Mechanics: see e.g. [PAU85]

Fractals, Turbulance

A.Arnéodo, G.Grasseau, and M.Holschneider, "Wavelet Transform of Multifractals," *Phys. Review Letters*, Vol.61, No.20, pp.2281-2284, 1988.

F.Argoul, A.Arnéodo, G.Grasseau, Y.Gagne, E.J.Hopfinger, and U.Frisch, "Wavelet Analysis of Turbulence Reveals the Multifractal Nature of the Richardson Cascade," *Nature*, Vol.338, pp.51-53, March 1989.

Radar/Sonar, Ambuguity Functions: see e.g. [WOO53], [SPE67], [AUS90]

Time-Scale Representations: see [FLA89], [FLA90], [RIO90a].

Filter Bank Theory (see also [CRI76], [CRO76], [EST77], [MIN85], [SMI86], [SMI87], [VAI88], [VAI89], [VET89])

J.D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," Proc. ICASSP-80, pp.291-294, April 1980.

T.Q.Nguyen and P.P.Vaidyanathan, "Two-Channel Perfect-Reconstruction FIR QMF Structures Which Yield Linear-Phase Analysis and Synthesis Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-37, No. 5, pp.676-690, May 1989.

P.P.Vaidyanathan, "Quadrature Mirror Filter Banks, M-band Extensions and Perfect-Reconstruction Techniques," *IEEE ASSP Magazine*, Vol. 4, No. 3, pp.4-20, July 1987.

M.Vetterli, "Multi-Dimensional Subband Coding: Some Theory and Algorithms," *Signal Processing*, Vol. 6, No.2, pp. 97-112, Feb. 1984.

M.Vetterli, "Filter Banks Allowing Perfect Reconstruction," *Signal Processing*, Vol.10, No.3, April 1986, pp.219-244.

One-Dimensional Signal Analysis

C.D'Alessandro and J.S.Liénard, "Decomposition of the Speech Signal into Short-Time Waveforms Using Spectral Segmentation," in *Proc. 1988 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, New York, Apr.11-14, 1988, pp.351-354.

J.L.Larsonneur and J.Morlet, "Wavelets and Seismic Interpretation," in [WAV89], pp.126-131, 1989.

F. B. Tuteur, "Wavelet Transformations in Signal Detection," in *Proc. 1988 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, New York, NY, Apr. 11-14, 1988, pp.1435-1438. Also in [WAV89], pp. 132-138, 1989.

M.V. Wickerhauser, "Acoustic Signal Compression with Wave Packets," preprint Yale University, 1989.

Pyramid Transforms: see [BUR83], [BUR89], [ROS84].

Wavelet Image Coding and Processing (see also [ANI90], [MAL89a], [MAL89b], [MAL89c].)

J.C. Feauveau, "Analyse Multirésolution pour les Images avec un Facteur de Résolution √2," [in French], *Traitement du Signal*, Vol. 7, No. 2, pp. 117-128, July 1990.

S.Mallat and S.Zhong, "Complete Signal Representation with Multiscale Edges," submitted to *IEEE Trans. Pattern Analysis and Machine Intell*, 1989.

FIGURES CAPTIONS

Fig. 1. Time-frequency plane corresponding to the Short-Time Fourier Trransform. It can either be seen as successive Fo urier Transforms (column by column) or as a modulated filter bank (line by line).

Fig. 2. Basis functions and time-frequency resolution of the Short-Time Fourier Transform (STFT) and the Wavelet Transform (WT). (a) Coverage of the time-frequency plane for the STFT, (b) for the WT. (c) Corresponding basis functions for the STFT, (d) for the WT ("wavelets".)

Fig. 3. Tiling of the frquency domain (a) for the STFT (uniform coverage) and (b) for the WT (logarithmic coverage.)

Fig. 4. Regions of influence in time (a),(b) or frequency (c),(d) drawn as shaded areas in the time-scale (a), (c) and time-frequency (b), (d) plane (see text.)

Fig. 5. Spectrograms and scalograms. Analysis of the sum of two Dirac pulses and two sinusoids (Fig. 5.1), of three starting sinusoids (Fig. 5.2.), of a chirp signal (Fig. 5.3.), and of a stretch of speech signal (Fig. 5.4.)

(a) Amplitude of the STFT. (b) Phase of the STFT. (c) Amplitude of the WT (d) Phase of the WT.

Fig. 6. From spectrograms to scalograms *via* Wigner-Ville. By controlling only one prameter μ, it is possible to make a full transition between time-scale and time-frequency analyses. Here seven analyses of the same signal (composed of three Gaissian packets) are shown. Note that the best joint tim-frequency resolution is attained for the Wigner-Ville distribution, while both spectrogram and scalogram (which can be thought of smoothed versions of Wigner-Ville) provide reduces cross-term effects compared to WIgner-Ville. (after [FLA90], [RIO90a].)

Fig. 7. Dyadic sampling grid in the time-scale plane. Each node corresponds to a wavelet basis function $h_{j,k}(t)$ (see text.)

Fig. 8. Reconstruction Signal/Noise Ratio (SNR) error after frame decomposition for different sampling densities $a_0 = 2^{1/N}$ (N = number of voices per octave), $b=a_0^j \, k \, b_0$. The basic wavelet is the Morlet wavelet (modulated Gaussian) used in [GRO89]. The accurateness of the reconstruction

grows as N increases and b_0 decreases, i.e. as the density of the sampling grid of Fig. 7 increases. In other words, redundancy refines reconstruction. (after [DAU90a].)

Fig. 9. Pyaramid Scheme. Derivation of a lowpass, subsampled approximation $y(n)$, from which an approximation $a(n)$ to $x(n)$ is derived by up-sampling and interpolation.

Fig. 10. Subband Coding Scheme. Two subsampled approximations, one corresponding to low and the other to high frequencies, are computed. The reconstructed signa. is obtained by re-interpolating the approximations and summing them. The filters on the left form an analysis filter bank, while on the right is a synthesis filter bank.

Fig. 11. Block diagram of the Discrete Wavelet Transform implemented with discrete-time filters and subsampling by two.

Fig. 12. Iterated low-pass filter. (a) $h(n) = (1,3,3,1)$ converges to a regular, smoothed function. (b) $h(n) = (-1,3,3,-1)$ converges to a fractal function (see text.)

Fig. 13. Scaling functions satisfying two-scale difference equations. (a) the hat function. (b) the D_4 wavelet obtained from a length-4 regular filter by Daubechies.

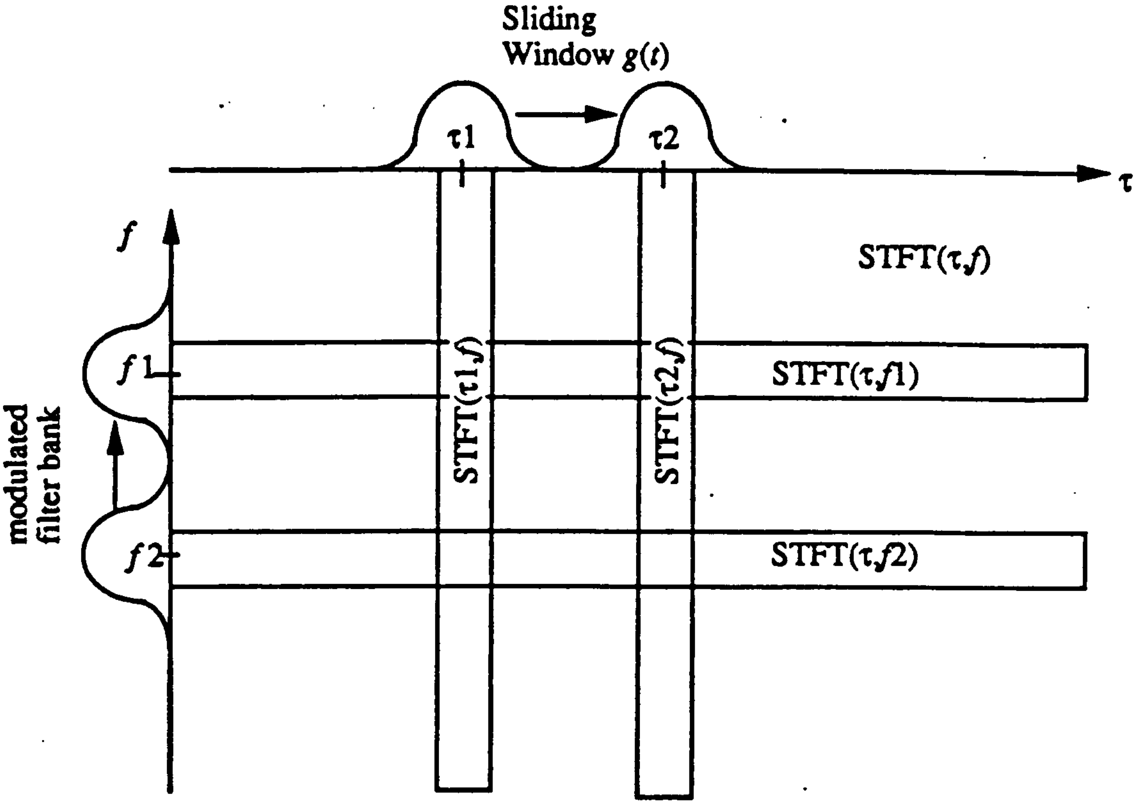


Fig. 1. Time-frequency plane corresponding to the Short-Time Fourier Trransform. It can either be seen as successive Fourier Transforms (column by column) or as a modulated filter bank (line by line).

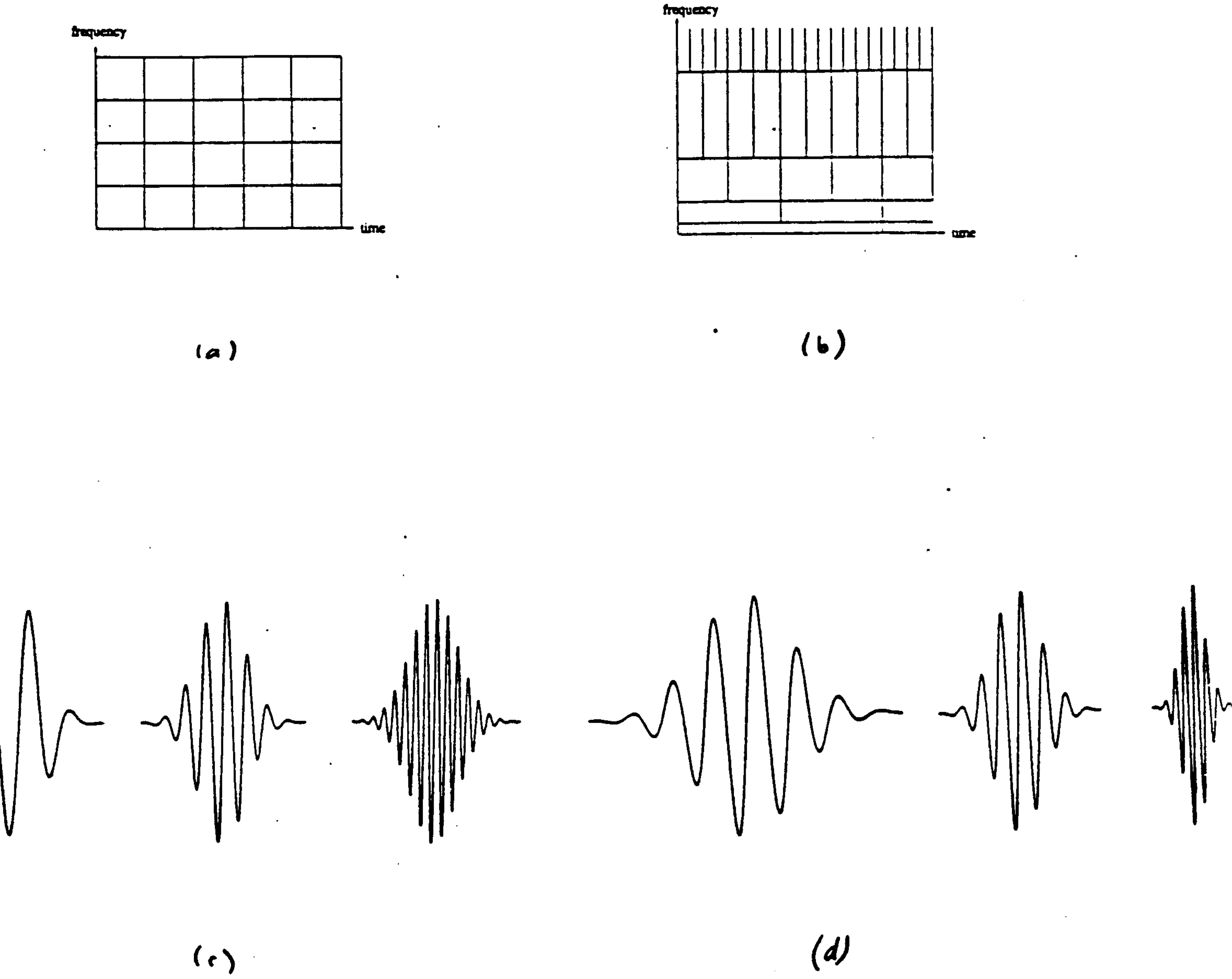


Fig. 2. Basis functions and time-frequency resolution of the Short-Time Fourier Transform (STFT) and the Wavelet Transform (WT). (a) Coverage of the time-frequency plane for the STFT, (b) for the WT. (c) Corresponding basis functions for the STFT, (d) for the WT ("wavelets".)

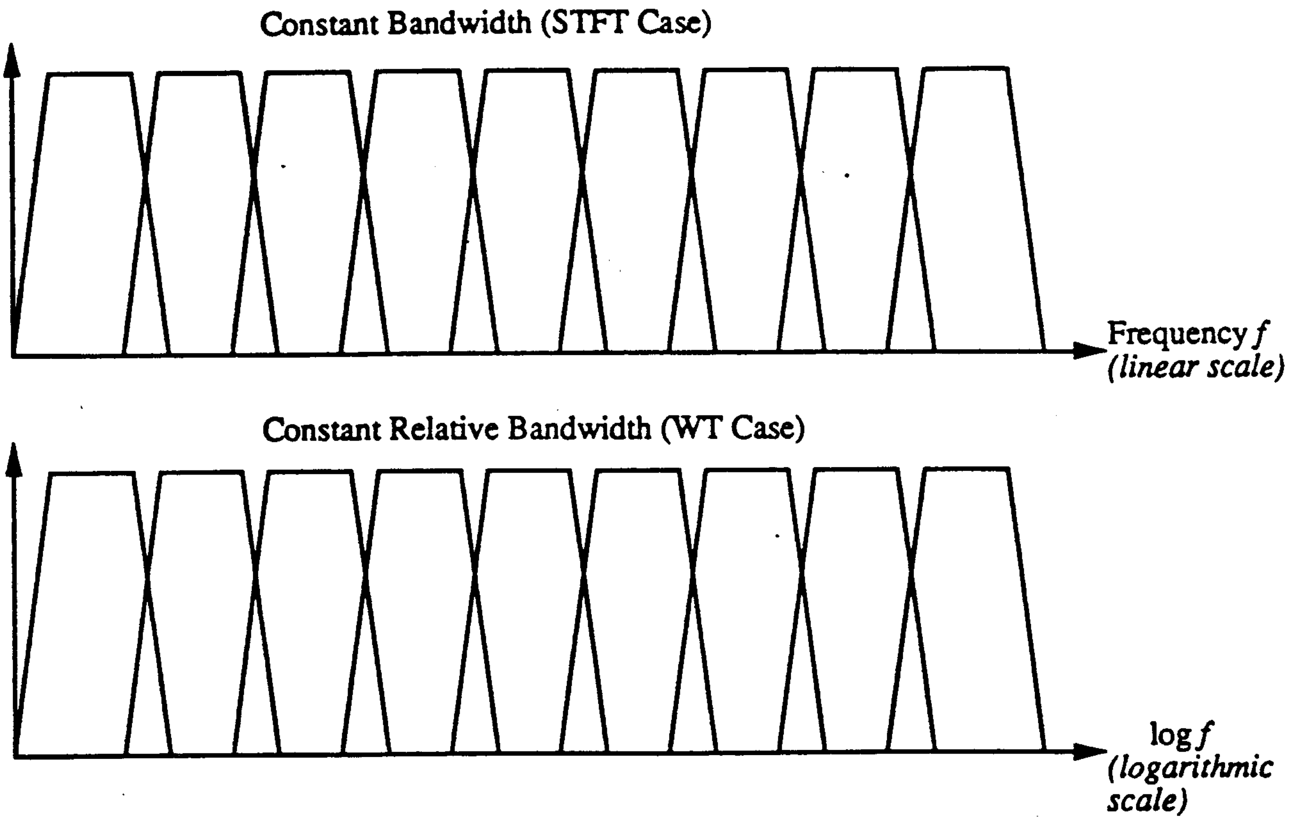

Constant Bandwidth (STFT Case)
Frequency *f*
(linear scale)
Constant Relative Bandwidth (WT Case)
log *f*
(logarithmic scale)

Fig. 3. Tiling of the frquency domain (a) for the STFT (uniform coverage) and (b) for the WT (logarithmic coverage.)

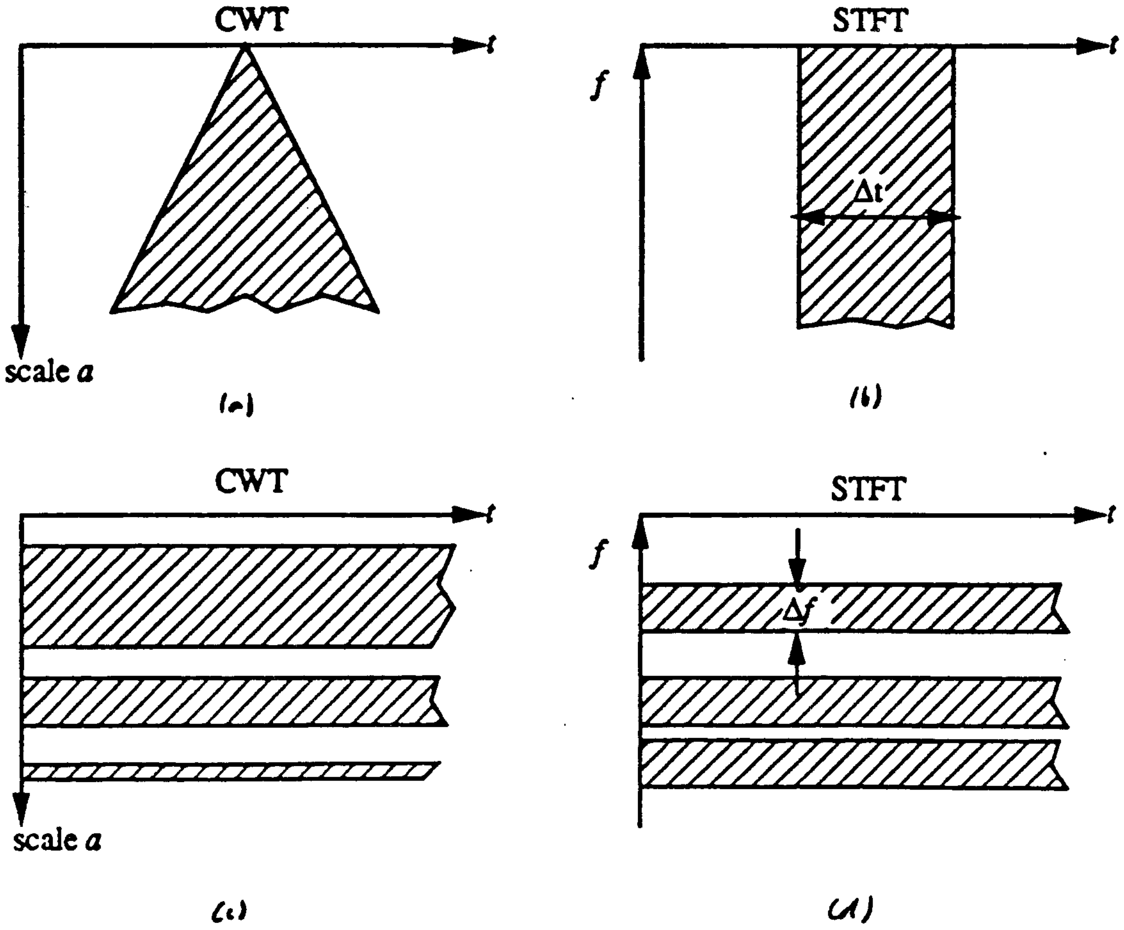


Fig. 4. Regions of influence in time (a),(b) or frequency (c),(d) drawn as shaded areas in the time-scale (a), (c) and time-frequency (b), (d) plane (see text.)

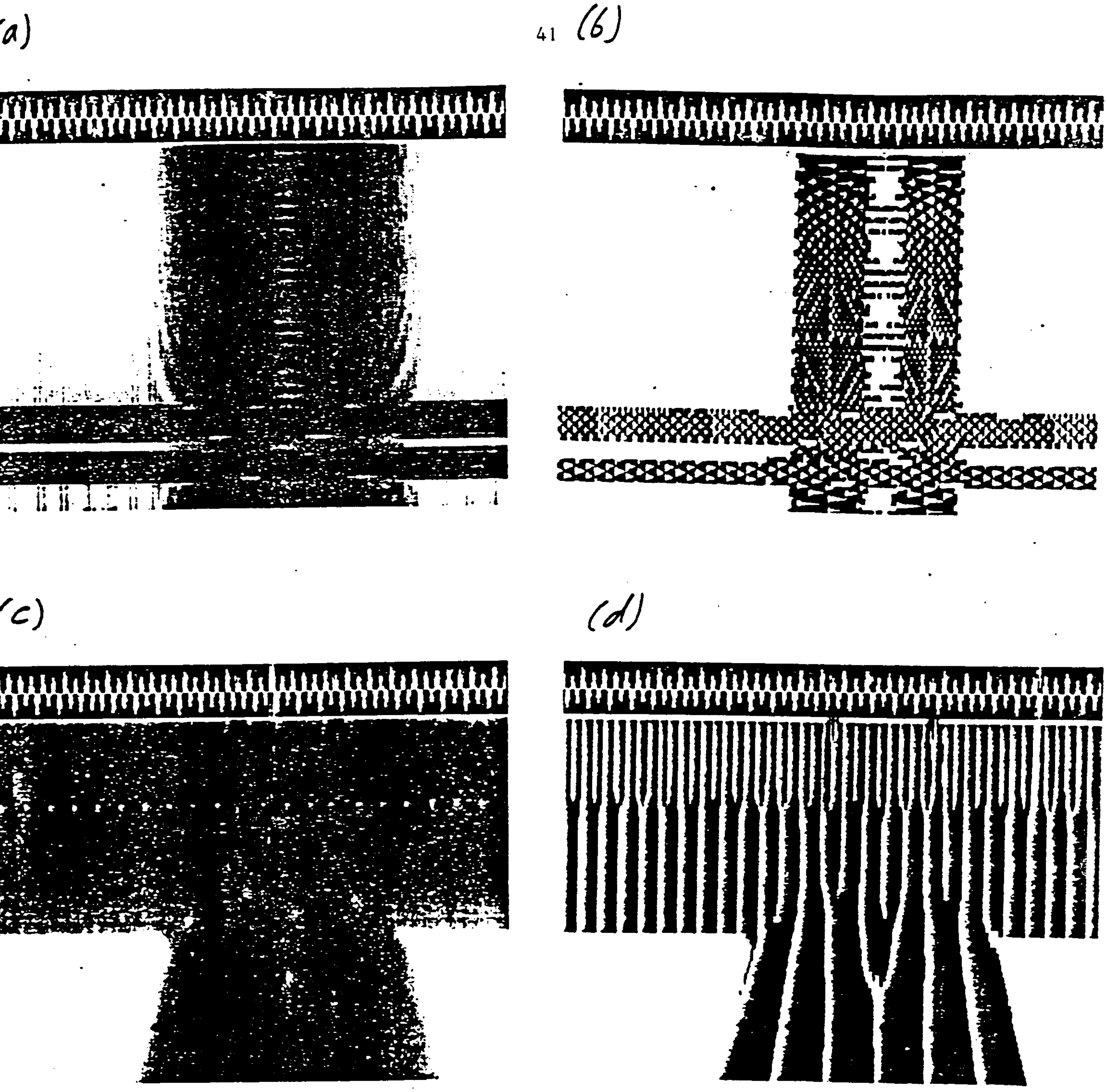


Fig. 5. Spectrograms and scalograms. Analysis of the sum of two Dirac pulses and two sinusoids (Fig. 5.1), of three starting sinusoids (Fig. 5.2.), of a chirp signal (Fig. 5.3.), and of a stretch of speech signal (Fig. 5.4.)

(a) Amplitude of the STFT. (b) Phase of the STFT. (c) Amplitude of the WT (d) Phase of the WT.

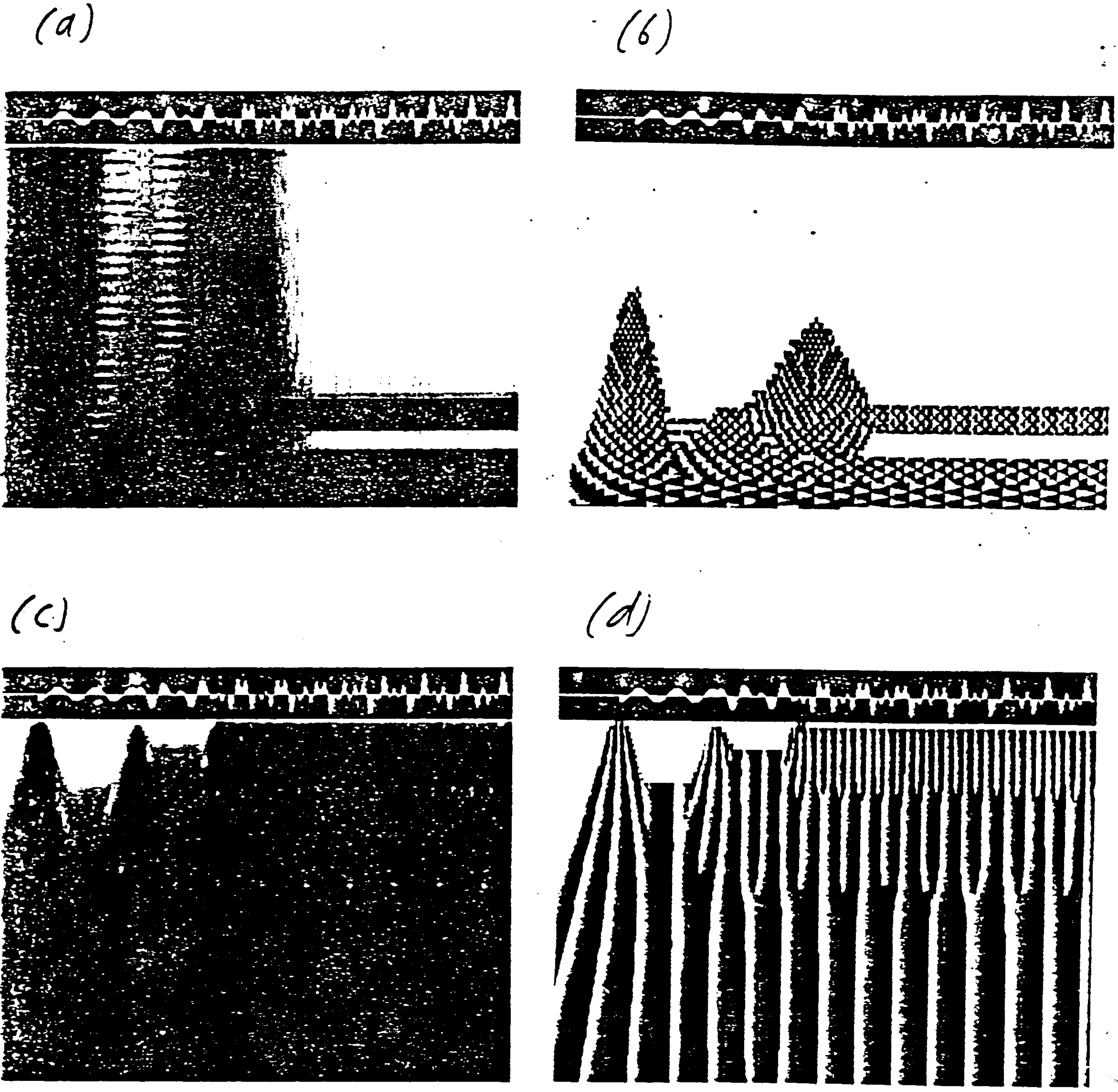


Figure 5.2

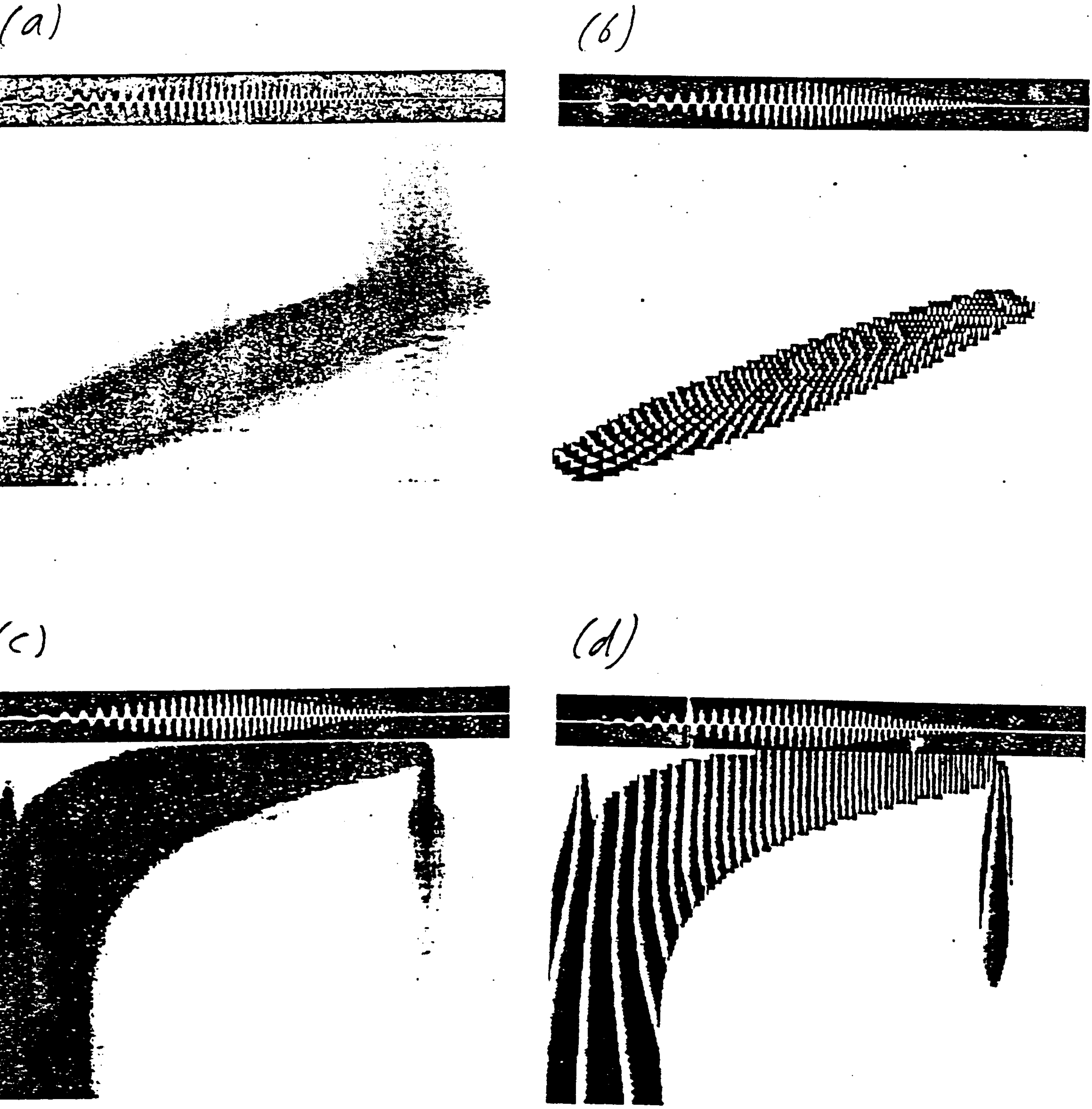


Figure 5.3

Note: Figure 5.4 is missing (speech)

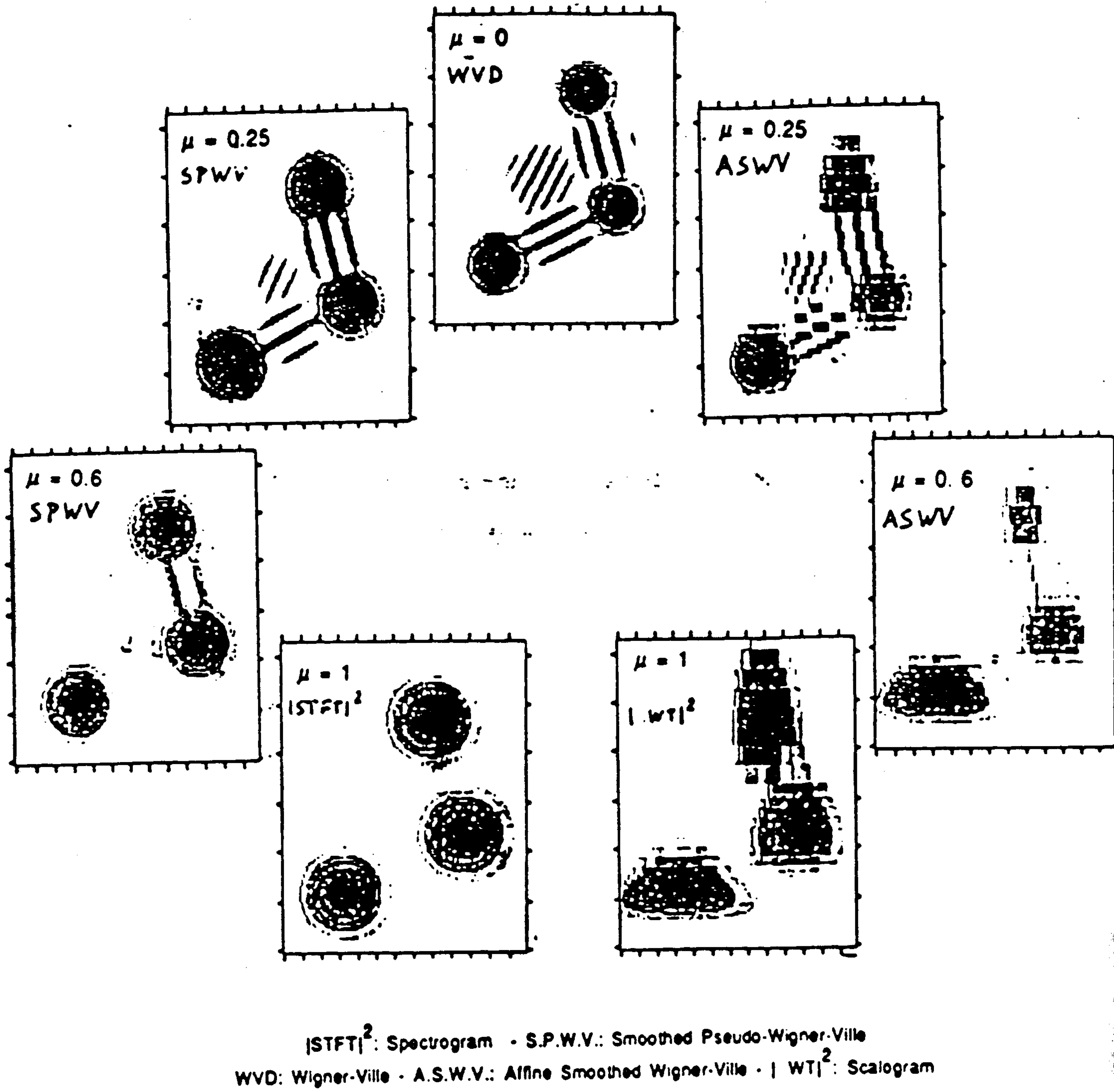


Fig. 6. From spectrograms to scalograms *via* Wigner-Ville. By controlling only one prameter μ, it is possible to make a full transition between time-scale and time-frequency analyses. Here seven analyses of the same signal (composed of three Gaissian packets) are shown. Note that the best joint tim-frequency resolution is attained for the Wigner-Ville distribution, while both spectrogram and scalogram (which can be thought of smoothed versions of Wigner-Ville) provide reduces cross-term effects compared to WIgner-Ville. (after [FLA90], [RIO90a].)

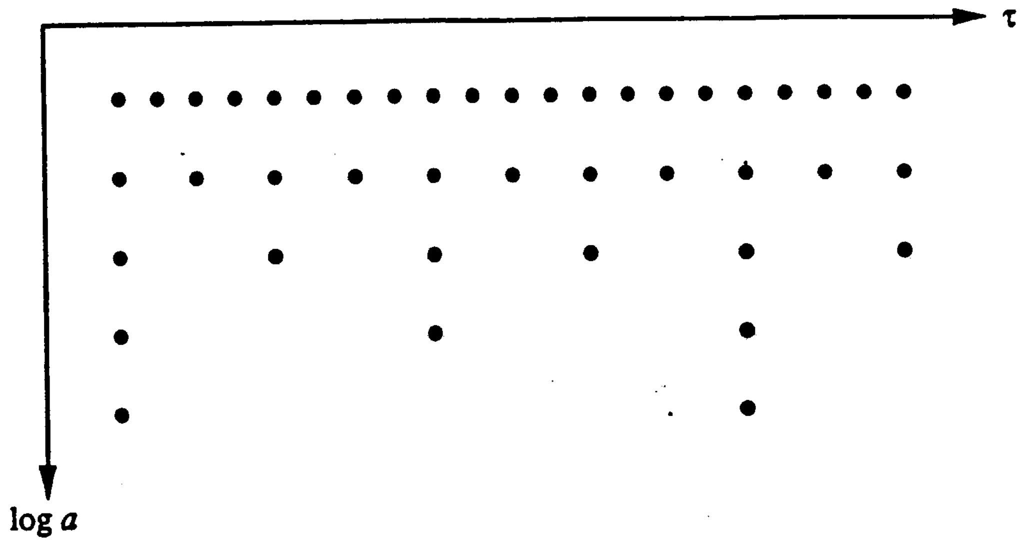


Fig. 7. Dyadic sampling grid in the time-scale plane. Each node corresponds to a wavelet basis function $h_{j,k}(t)$ (see text.) .

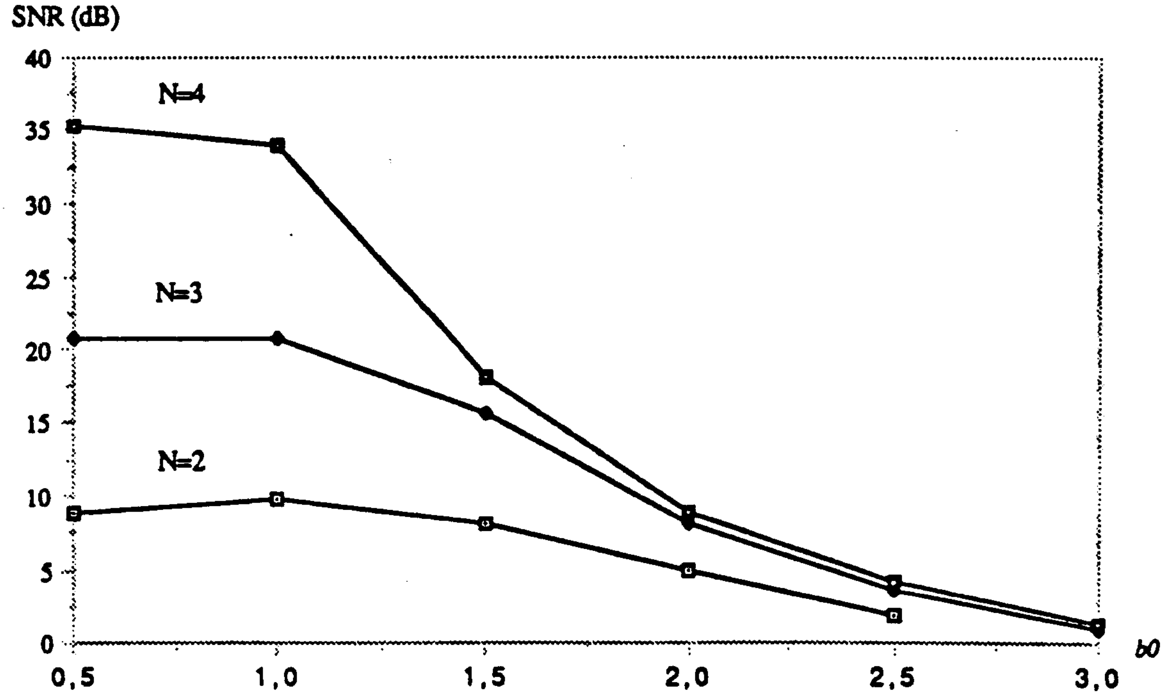


Fig. 8. Reconstruction Signal/Noise Ratio (SNR) error after frame decomposition for different sampling densities $a_0 = 2^{1/N}$ (N = number of voices per octave), $b=a_0^j\, k\, b_0$. The basic wavelet is the Morlet wavelet (modulated Gaussian) used in [GRO89]. The accurateness of the reconstruction grows as N increases and b_0 decreases, i.e. as the density of the sampling grid of Fig. 7 increases. In other words, redundancy refines reconstruction. (after [DAU90a].)

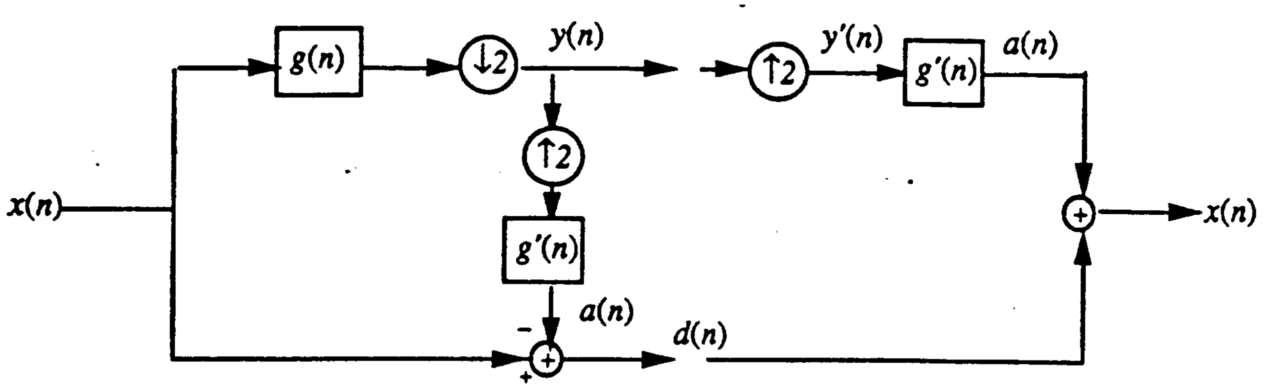


Fig. 9. Pyaramid Scheme. Derivation of a lowpass, subsampled approximation $y(n)$, from which an approximation $a(n)$ to $x(n)$ is derived by up-sampling and interpolation.

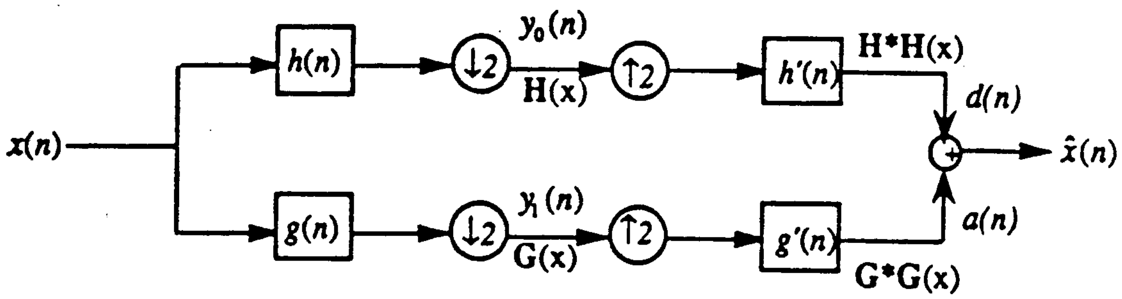


Fig. 10. Subband Coding Scheme. Two subsampled approximations, one corresponding to low and the other to high frequencies, are computed. The reconstructed signal is obtained by re-interpolating the approximations and summing them. The filters on the left form an analysis filter bank, while on the right is a synthesis filter bank.

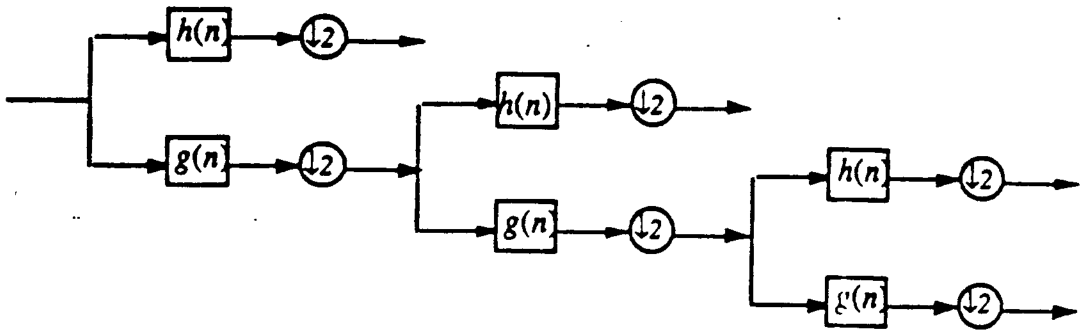


Fig. 11. Block diagram of the Discrete Wavelet Transform implemented with discrete-time filters and subsampling by two.

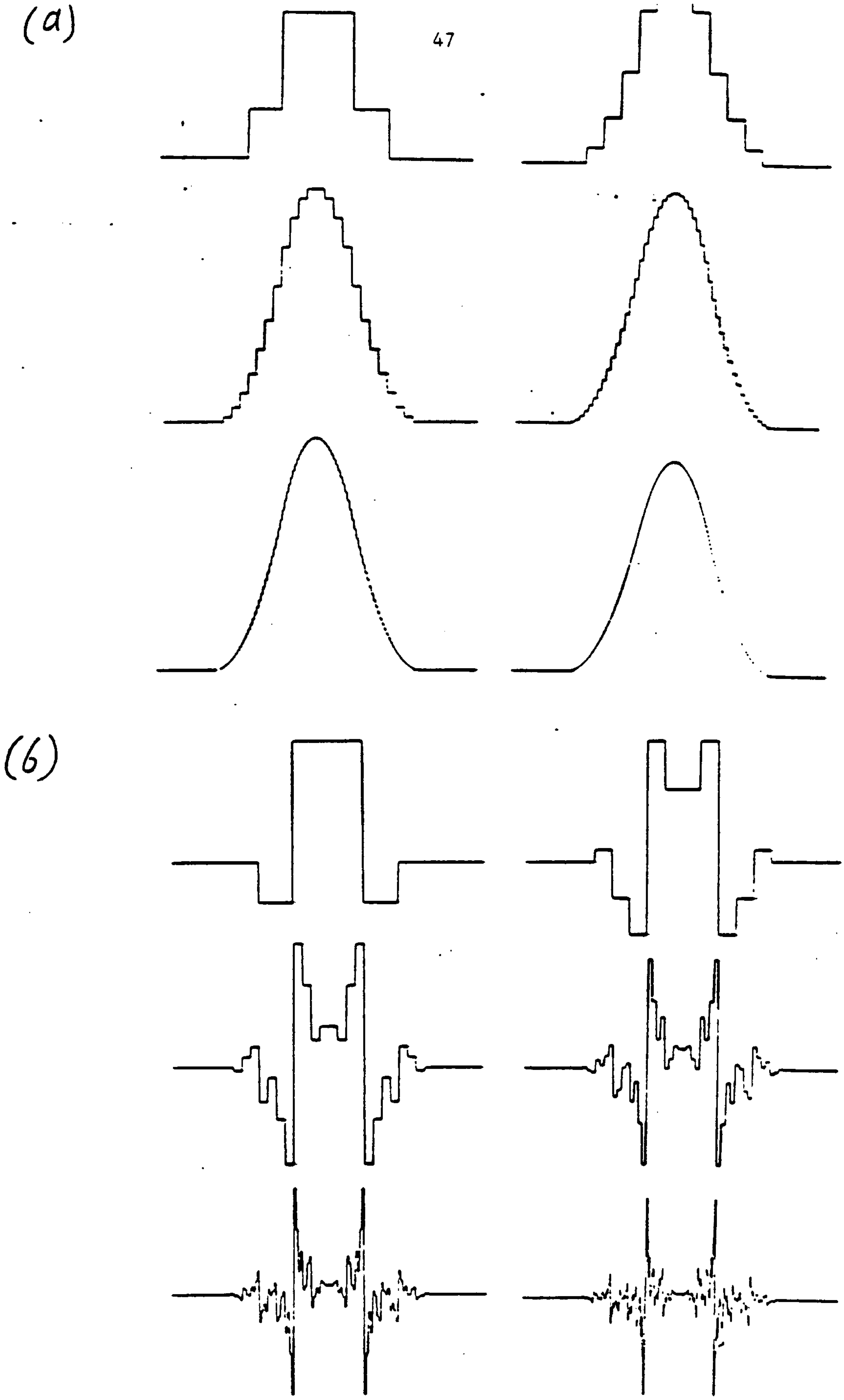


Fig. 12. Iterated low-pass filter. (a) $h(n) = (1,3,3,1)$ converges to a regular, smoothed function. (b) $h(n) = (-1,3,3,-1)$ converges to a fractal function (see text.)

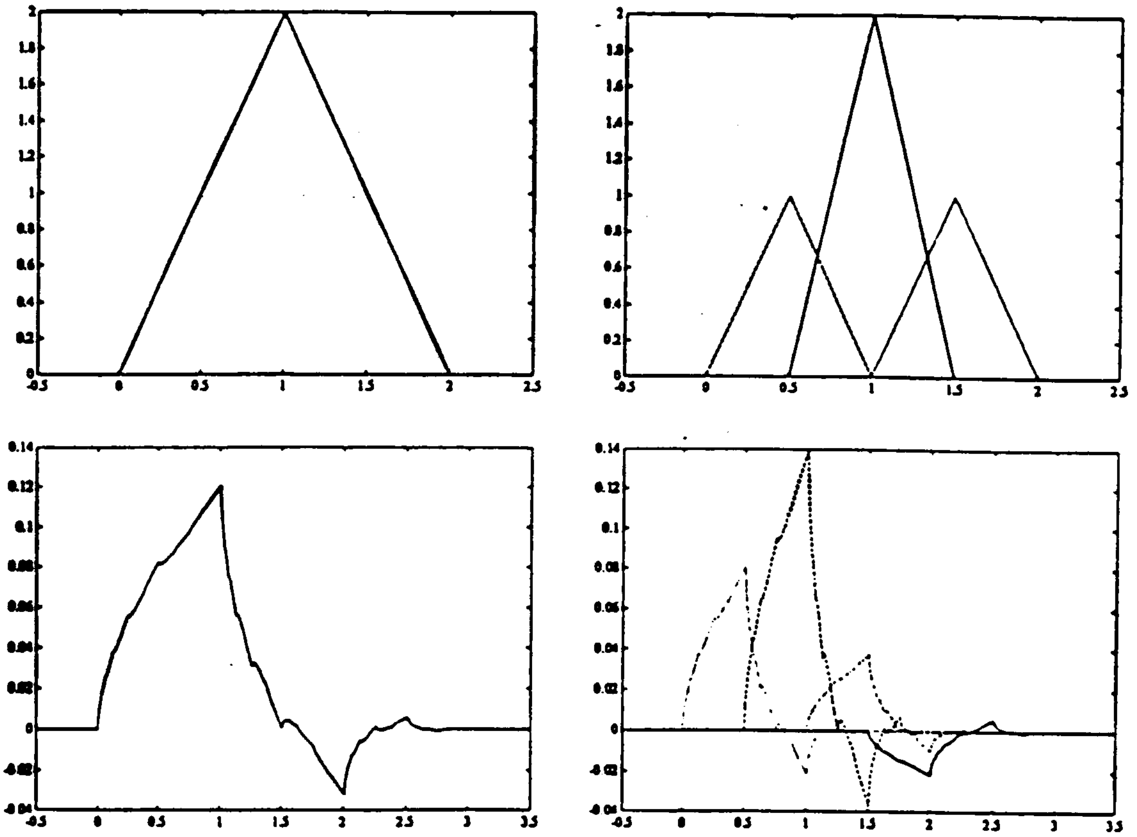


Fig. 13. Scaling functions satisfying two-scale difference equations. (a) the hat function. (b) the D_4 wavelet obtained from a length-4 regular filter by Daubechies.

ARTICLE 2

"A Discrete-Time Multiresolution Theory Unifying Octave-Band Filter Banks, Pyramid and Wavelet Transforms", par O. RIOUL

IEEE Trans. on ASSP, soumis en Juin. 1990, révisé en Avril 1991

A Discrete-Time Multiresolution Theory Unifying Octave-Band Filter Banks, Pyramid and Wavelet Transforms

OLIVIER RIOUL

Centre National d'Etudes des Télécommunications
C.N.E.T/PAB/RPE
38-40 rue du Général Leclerc
92131 Issy-Les-Moulineaux
France

Abstract — A global formalism for the analysis and synthesis of discrete-time signals at different resolutions is presented in this paper. It uses precise definitions of scale and resolution to unify the Discrete Wavelet Transform (DWT), octave-band perfect reconstruction Filter Banks, and Pyramid Transforms in a common framework. These schemes are described as a decomposition of an original signal into multiresolution "details": in a DWT, these details are supported by discrete wavelets.

Although close to analog models such as wavelet series decomposition, this approach is purely discrete-time-based, and is therefore readily applicable to practical Digital Signal Processing tasks. In addition, it has the same mathematical richness than the corresponding analog models, including several properties or filter design criteria such as biorthonormality, orthonormality and regularity.

INTRODUCTION.

Wavelet Transforms, Octave-Band Filter Banks and Pyramid Transforms, which have been used for different purposes in various fields of Electrical Engineering and Mathematics, have been recognized recently as different views of a common theory. This theory is hereafter referred to as *Multiresolution Analysis and Synthesis*.

The aim of this paper is not to provide a complete state-of-the-art in Multiresolution Theory or Wavelet Theory (see, e.g., [6], [8], [29]) — nor is this paper intended to paraphrase mathematical concepts [22], [23]. Rather, this paper intends to enlighten various links between Filter Banks [31], Pyramid Transforms [5] and Wavelets [8], while deliberately focusing on discrete-time signals.

Before stating in more detail the motivations that lead to this paper, it is necessary to briefly outline the basic properties of Wavelet Transforms and review their connections to Filter Banks.

The *Continuous Wavelet Transform* has been first introduced by Goupillaud, Morlet and Grossmann [15], [16]. It maps a one-dimensional signal $x(t)$ into a two-dimensional representation in a time-scale plane (b, a), as follows.

$$\mathrm{CWT}(a, b) = a^{-1/2} \int x(t)\psi^{*}(\tfrac{t-b}{a})\, dt \qquad (1)$$

The functions of time: $\psi(\frac{t-b}{a})$, called *wavelets*, are used to band-pass filter the signal. This is a sort of time-varying spectral analysis, where the *scale* parameter a plays the role of a local frequency: for large a, stretched wavelets analyze low frequencies; for small a, short, contracted wavelets analyze high frequencies. It is a *multiresolution* analysis, because the time-frequency extent of wavelets varies in the time-scale plane (b, a) according to a constant-Q analysis [36] — this feature is not provided by e.g. Short-Time Fourier Transforms [3]. As one would expect, the Continuous Wavelet Transform is mostly used in Signal Analysis.

For coding or compression purposes, it is wise to economically represent the signal in terms of wavelet coefficients (1), hence to discretize time-scale parameters (b,a). Scale is often discretized octave by octave [8], i.e., $a = 2^j, j \in \mathbb{Z}$, and wavelets are shifted in direct proportion to their extent, i.e., $b = k\,2^j$. Then, wavelet coefficients become inner products of the signal against analyzing wavelets $\psi_{j,k}(t)$:

$$w^j_k = \int x(t)\psi^{*}_{j,k}(t)\, dt, \qquad (2)$$

where

$$\psi_{j,k}(t) = 2^{-j/2}\,\psi(2^{-j}\,t - k). \qquad (3)$$

This model still applies to analog (continuous-time) signals. A rich, mathematical theory has been developped by several mathematicians (including Meyer [23], Mallat [21], [22]), which is commonly referred to as "Multiresolution Analysis." It is based on a description of approximated versions of signals at different resolutions using so-called "Multiresolution Spaces." Some of these ideas previously occured in image coding [5], [1].

Meyer's theory of Multiresolution Analysis allowed several authors [7], [11], [20], [23], to construct special wavelet prototypes $\psi(t)$ for which perfect reconstruction of the signal $x(t)$ is possible *via* a *Wavelet Series,*

$$x(t) = \sum_{j \in Z} \sum_{k \in Z} w^j_k \, \tilde{\psi}_{j,k}(t), \tag{4}$$

where synthesis wavelets $\tilde{\psi}_{j,k}(t)$ are defined similarly as (3), possibly with a different prototype $\tilde{\psi}(t) \neq \psi(t)$, closely related to $\psi(t)$ [7].

We now turn to the connection with Filter Banks. In order that wavelet series coefficients (2) be computed efficiently, Mallat [21] has derived a recursive algorithm that iteratively computes (2) octave by octave. This algorithm can easily be transposed to provide an inverse algorithm for the reconstruction part. Mallat's direct and inverse algorithms turn out to be exactly an analysis and synthesis *Octave-Band Filter Bank* [30].

Wavelet theory is therefore closely related to Filter Bank theory. The latter, which has been previously developped for applications such as subband coding of speech [10], has greatly influenced the former; known FIR filter design techniques [17], [30] were independently used by Daubechies [11] to construct orthonormal wavelets of finite length satisfying (2), (4) with $\tilde{\psi}(t) = \psi(t)$. Other FIR filter designs were recently used [7], [34], [35] to construct linear phase, "biorthonormal" wavelets for which $\tilde{\psi}(t) \neq \psi(t)$.

The connection between wavelet series and filter banks has thus been extensively studied already. However, the "discrete-time side" of wavelet series has not been fully exploited yet. It has been mainly used as a technical necessity to either derive fast algorithms (which simply rely on the filter bank structure itself — for further developments see [26], [27]) or construct analog wavelet bases (*via* filter design). Very little multiresolution wavelet theory has been developped, if not at all, in the sole framework of discrete-time signals: according to what can be found in the literature, analog wavelets are the interesting objects that underly discrete-time computational tools.

This situation is similar to the one formerly observed in Signal Processing for linear filters and Fourier transforms. Discrete filters, as well as the Discrete Fourier Transform, were once exclusively considered as discretized versions of their analog counterparts, rather than interesting objects of their own. Since the mid-sixties, however, Discrete-Time Signal Processing has emerged as an independent theory [25], that has subsequently allowed new developments.

Very likely, a fully discrete-time presentation of Multiresolution Theory would be more appropriate for most practical situations in Digital Signal Processing; as a matter of fact, one often processes digital signals as the interesting objects. For example, an image is generally thought of as a nxm-pixel 8-bit data array, not as basis coefficients of a two-dimensional square integrable function of $\mathbf{L}^2(\mathbb{R}^2)$. Various image compression schemes have been derived, based on the wavelet model for analog signals [4], [37], but they are essentially discrete in nature.

For this reason, this paper focuses on discrete-time signals. Its primary purpose is to develop a discrete-time multiresolution theory that e.g. permits to define a *Discrete Wavelet Transform* (DWT) in its own right — that acts on one-dimensional discrete-time signals. In a DWT, the interesting objects are the *discrete* wavelets. Extension to two-dimensional signals can be straightforward: a wavelet decomposition of an image can be simply obtained by processing rows and columns separately [4], [21]. Other methods exist [33], but the underlying ideas are the same as in one dimension, which is the framework considered in this paper.

This paper is for most part self-contained: in writing the paper, care has been taken that the reader need not previous knowledge on wavelet basics (although some rudimentary knowledge of z-transforms and vector spaces is assumed). As a result, since the discrete-time and continuous-time approaches have many things in common, most developments and ideas presented in this paper are known to wavelet experts — yet they may not appear as presented in the literature.

NOTATIONS

x or $\{x_n\}$	Original signal: a discrete, complex-valued sequence.
x_n	Its nth sample.
δ	Impulse signal: $\delta_n = 0$ if $n\neq 0$, $\delta_0 = 1$.
$\mathbf{I}$	Identity operator: $\mathbf{I}x = x$.
$\mathbf{G}$, $\mathbf{H}$	Low-pass (high-pass) filtering operator, with impulse response $\{g_n\}$ ($\{h_n\}$).
$\mathbf{G}x$	Low-pass filtered signal, i.e., the discrete convolution $g * x$.
$(\mathbf{G}x)_n$	Its nth sample. When needed, the input sample index is shown, as in $(\mathbf{G}x_k)$.
$\uparrow x$, $\downarrow x$	Up-sampled and down-sampled signal: $(\uparrow x)_{2n}=x_n$, $(\uparrow x)_{2n+1}=0$, and $(\downarrow x)_n = x_{2n}$.
$\mathbf{G}\uparrow x$, $\downarrow \mathbf{G}'x$	Up-scaled and down-scaled signal (see section I).
$\langle x; y\rangle$ or $\langle x_n; y_n\rangle$	Inner product of signals $\{x_n\}$ and $\{y_n\}$: $\langle x; y\rangle = \sum x_n y^*_n$.
$\lVert x \rVert$	Norm of signal $\{x_n\}$: $\lVert x \rVert^2 = \langle x; x\rangle = \sum_n \lvert x_n \rvert^2$."

I. Discrete-Time Scaling

The aim of this section is to define precisely the concept of *scale* for discrete-time signals. This notion is inspired from the commonly used scale of road maps: given a "real" object, the *scale* of a representation of this object is the ratio of the length unit of the representation to the corresponding real length. Here the "real object" is an original discrete-time signal $\{x_n\}$, which is at scale 1 by definition. In this paper, scale refers to discrete time: a scaled version of $\{x_n\}$ is either

- *up-scaled*: a discrete-time signal similar to $\{x_n\}$, but sampled at a higher rate, or
- *down-scaled*: a discrete-time signal similar to $\{x_n\}$, but sampled at a lower rate.

Since scale is a relative notion, we focus on the description of *operators* that change the scale, i.e., map a signal into a scaled version of it. Throughout the paper we restrict to changes of scale by an integer power of two (but other scale factors are sometimes used [18]). We therefore study two basic scaling operators:

- *up-scaling operator* (by a factor 2): a discrete equivalent to the dilation $x(t) \to x(t/2)$.
- *down-scaling operator* (by a factor 2): a discrete equivalent to the contraction $x(t) \to x(2t)$.

Examples. Obviously the scale notion is related to multirate systems. For example, a down-scaled version of $\{x_n\}$ could be $\{x_n\}$ itself, *down-sampled*:

$$(\downarrow x)_n = x_{2n}. \tag{5}$$

However, *up-sampling*:

$$(\uparrow x)_n = \begin{cases} x_{n/2}, & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \tag{6}$$

is not a good candidate for up-scaling, since it inserts a zero between every other sample of $\{x_n\}$; hence x and $\uparrow x$ do not have similar evolutions in time. This can be corrected by further interpolating the samples, as in the following example, illustrated in Fig. 1.

$$\text{Up-scaled version of } x_n = \begin{cases} x_{n/2}, & n \text{ even} \\ \frac{1}{2}(x_{(n-1)/2} + x_{(n+1)/2}), & n \text{ odd} \end{cases} \tag{7}$$

We now determine general expression for up and down-scaling operators. We need some basic assumptions:

(a) Scaling operators are linear.

(b) When scaling a signal, *time-shifts* are scaled accordingly:

If y_n = up-scaled version of x_n, then y_{n-2k} = up-scaled version of x_{n-k}.

If y_n = down-scaled version of x_n, then y_{n-k} = down-scaled version of x_{n-2k}.

(c) *Shape preservation*: Scaled versions have similar time evolutions (similar shapes).

The third point is difficult to be properly expressed. Therefore it is not considered until section VIII, where it is connected to the "regularity" property.

Assumptions (a) and (b) result in the following characterizations, proven in Appendix A:

The up-scaled version of x is of the form $\mathrm{G}\!\uparrow\! x$, where $\mathrm{G}\!\uparrow$ denotes up-sampling (6) followed by filtering with some impulse response $\{g_n\}$:

$$(\mathrm{G}\!\uparrow\! x)_n = \sum_k g_{n-2k}\, x_k\,. \tag{8}$$

The down-scaled version of x is of the form $\downarrow\!\mathrm{G}'x$, where $\downarrow\!\mathrm{G}'$ denotes filtering with some impulse response $\{g'_n\}$, followed by down-sampling (5):

$$(\downarrow\!\mathrm{G}'x)_n = \sum_k g'_{2n-k}\, x_k\,. \tag{9}$$

The impulse responses $\{g_n\}$ and $\{g'_n\}$ are not necessarily equal. They are assumed *low-pass* in the sequel; intuitively this is required by assumption (c) above — section VIII gives a theoretical justification. The example (7) corresponds to a 3-tap up-scaling filter $g_{\pm 1}= .5$, $g_0=1$.

The corresponding flow graphs of up and down scaling operators are shown in Fig. 2. They turn out to be usual building blocks of analysis and synthesis Filter Banks [31]! Note that the operator notation used here is very easily connected to flow-graph implementation: for example, $\mathrm{G}\!\uparrow\downarrow\!\mathrm{G}'x$ means that the input x successively encounters a filter of impulse response g', a down-sampler, an up-sampler, and finally a filter of impulse response g.

Using only two operators (8), (9), one can compute scaled versions of the original signal $\{x_n\}$ at *all* dyadic scales $s = 2^{-i}$, where $i \in \mathbb{Z}$. Simply, the up-scaling operator (8) doubles the scale, while the down-scaling operator (9) halves it. *Scaling filters* g_n, g'_n remain fixed at all scales, i.e., all scale transitions are performed the same way, and no particular scale is privileged.

However, a scaled version of a given signal $\{x_n\}$ at a given scale s is not unique. For example, x, $(\mathrm{G}\!\uparrow)^N(\downarrow\!\mathrm{G}')^N x$, and $(\mathrm{G}\!\uparrow)^N(\downarrow\!\mathrm{G}')^M(\mathrm{G}\!\uparrow)^M(\downarrow\!\mathrm{G}')^N x$ are *all* "scaled" versions of x at scale 1! To characterize scaled versions of x we need another parameter than scale, namely, *resolution*. This will be discussed in section III.

II. Hermitian Transposition and Inner Products

Before introducing the notion of resolution, it is convenient to say a few words about Hermitian transposition of operators, and inner products.

In matrix notation, the up-scaling operator is

$$\mathrm{G}\uparrow x = \begin{pmatrix} & & & \vdots & & & \\ \cdots & g_4 & g_2 & g_0 & g_{-2} & g_{-4} & \cdots \\ \cdots & g_5 & g_3 & g_1 & g_{-1} & g_{-3} & \cdots \\ \cdots & g_6 & g_4 & g_2 & g_0 & g_{-2} & \cdots \\ & & & \vdots & & & \end{pmatrix} \begin{pmatrix} \vdots \\ x_{-1} \\ x_0 \\ x_1 \\ \vdots \end{pmatrix} \tag{10}$$

Its Hermitian transpose is, by definition, the operator obtained by transposing and complex conjugating the matrix (10), i.e.,

$$\begin{pmatrix} & & & \vdots & & & \\ \cdots & g_1^* & g_2^* & g_3^* & g_4^* & g_5^* & \cdots \\ \cdots & g_{-1}^* & g_0^* & g_1^* & g_2^* & g_3^* & \cdots \\ \cdots & g_{-3}^* & g_{-2}^* & g_{-1}^* & g_0^* & g_1^* & \cdots \\ & & & \vdots & & & \end{pmatrix}$$

This is exactly the matrix form of a *down-scaling* operator (9), whose scaling filter is $\{g^*_{-n}\}$. Following the Filter Bank terminology [31], $\{g^*_{-n}\}$ is called the *paraconjugate sequence* of $\{g_n\}$ (the connection with filter banks will be explained in section VI). Paraconjugation will be denoted in this paper with a tilde symbol:

$$\tilde{g}_n = \text{paraconjugate of } g_n = g^*_{-n} \tag{11}$$

We have thus shown that *the Hermitian Transpose of the up-scaling operator* $\mathrm{G}\uparrow$ *is the paraconjugate down-scaling operator* $\downarrow\tilde{\mathrm{G}}$. Similarly, *the Hermitian Transpose of the down-scaling operator* $\downarrow\mathrm{G}'$ *is the paraconjugate up-scaling operator* $\tilde{\mathrm{G}}'\uparrow$.

Hermitian transposition is useful for several reasons. First, it has a flow-graph interpretation. In fact, it is well known [9], [25] that transposing a linear operator amounts to transposing its flow graph: the Hermitian transposed flow graph is obtained by reversing the directions of all arrows — hence summing nodes become branching nodes and *vice versa* — and by complex conjugating the multipliers' coefficients. From the discussion above, it follows that the flow graphs of Fig. 2 are eachother's Hermitian transpose if and only if the scaling filters $\{g_n\}$ and $\{g'_n\}$ are eachother's paraconjugate (11). In any case, the flow-graph computational.structures of up and down-scaling operators are always tranpose of eachother. This has useful implications for e.g. deriving algorithms [26], [27]: for example, once an algorithm has been derived to compute down-scaling, the transposed algorithm computes up-scaling (and *vice versa*), at the same computational cost if both scaling filters have same length. The situation is the same for direct and inverse Wavelet Transforms (see section VI).

Another characterization of Hermitian transposition uses inner products and will be useful in the sequel. By definition, the *inner product* of two discrete-time signals $\{x_n\}$ and $\{y_n\}$ is the number

$$\langle x; y\rangle = \sum_n x_n y^*_n = \begin{pmatrix} \cdots & x_{-1} & x_0 & x_1 & \cdots \end{pmatrix} \begin{pmatrix} \vdots \\ y^*_{-1} \\ y^*_0 \\ y^*_1 \\ \vdots \end{pmatrix} \tag{12}$$

This definition requires that signals have finite energy, i.e.,

$$\| x \|^2 = \langle x; x\rangle = \sum_n | x_n |^2 < +\infty, \tag{13}$$

which is assumed throughout this paper. The inner product $\langle x; y\rangle$ measures the "similarity" between x and y. It also permits to interpret signals as geometrical vectors: for example, the signals x and y are said to be *orthogonal* if $\langle x; y\rangle = 0$. They are *orthonormal* if they moreover have norm unity, i.e., $\| x \| = \| y \| = 1$.

Using the definition (12), it is easy to show that the *Hermitian transpose* $O^\dagger$ of some operator O can be alternatively defined as satisfying

$$\langle x ;\ Oy \rangle = \langle O^\dagger x ;\ y \rangle, \tag{14a}$$

or, equivalently,

$$\langle x ;\ O^\dagger y \rangle = \langle Ox ;\ y \rangle, \tag{14b}$$

for any signals x and y. These equations do not introduce a new concept: only notations are new. They are useful when dealing with inner products because of their conciseness: any operator on the left side of an inner product can be brought to the right side after Hermitian transposition, and *vice versa*. For scaling operators we can thus write

$$\langle x\,;\ \mathrm{G}{\uparrow}y\rangle = \langle{\downarrow}\tilde{\mathrm{G}}\,x\,;\ y\rangle\ . \tag{15a}$$

$$\langle x\,;\ {\downarrow}\mathrm{G}'y\rangle = \langle\,\tilde{\mathrm{G}}'{\uparrow}x\,;\ y\rangle, \tag{15b}$$

which stand for

$$\sum_n x_n\left(\sum_k g_{n-2k}\,y_k\right)^* = \sum_n\left(\sum_k \tilde{g}_{2n-k}\,x_k\right)y_n^*$$

and

$$\sum_n x_n\left(\sum_k g'_{2n-k}\,y_k\right)^* = \sum_n\left(\sum_k \tilde{g}'_{n-2k}\,x_k\right)y_n^*\ .$$

III. Discrete-Time Resolution and Biorthonormality

In this section we define the *resolution* notion for characterizing different versions of an original signal $\{x_n\}$ at the same scale. Intuitively, the more information is present in a scaled version of $\{x_n\}$, the higher the resolution. More precisely, we define the *resolution* parameter r as follows.

A scaled version of $\{x_n\}$, obtained by action of up and/or down-scaling operators on $\{x_n\}$, is *at resolution* $r=2^{-j}$ ($j\geq 0$) if it is characterized by one sample every other $r^{-1}=2^j$ sampling periods of $\{x_n\}$.

For example, x itself is at resolution 1. Its down-scaled version ${\downarrow}\mathrm{G}'x$ (9), which is at scale $s=1/2$, is also at resolution 1/2, because the down-scaling operator throws away half of the samples. Note that in general, the resolution of a signal cannot exceed its scale, otherwise it would be characterized by more samples than actually present in the signal! Therefore, we always have

$$\textit{resolution} \leq \textit{scale}, \tag{16}$$

and because we start from an original signal $\{x_n\}$ at resolution 1, all resolutions considered in this paper are negative powers of two: $r=2^{-j}, j\geq 0$.

How is affected resolution by up and down-scaling operators? We have seen that down-scaling, when applied to the original signal $\{x_n\}$, cuts both scale and resolution by two. As for up-scaling, since an up-scaled signal is computed directly from the signal coefficients, no resolution is added. That is, up-scaling "magnifies" a signal but does not add new details. In addition, despite low-pass filtering present in up-scaling (8), it is not necessary that information is lost when up-scaling a signal. This is clear from the example (7). We therefore assume in this paper that the up-scaling operator $\mathrm{G}{\uparrow}$ is *one-to-one*:

$$\text{If } x \neq y, \text{ then } \mathrm{G}{\uparrow}x \neq \mathrm{G}{\uparrow}y \tag{17}$$

This can be proven if e.g. the scaling filter$\{g_n\}$ is FIR, by noting that $\mathrm{G}{\uparrow}x = 0$ implies, from (8), $g_{2n} * x_n = g_{2n+1} * x_n = 0$, hence $x = 0$.

· To summarize, there are two important rules concerning changes of scale and resolution:

— *When replacing the original signal x by its down-scaled version $\downarrow \mathbf{G}'x$, both scale and resolution are halved.* (18a)

— *Up-scaling any signal via $\mathbf{G}\uparrow$ doubles its scale but leaves resolution unchanged.* (18b)

Another example permits to halve the resolution while leaving the scale unchanged. It plays a central role in the following. Consider

$$Ax = \mathbf{G}\uparrow\downarrow\mathbf{G}'x, \tag{19}$$

obtained by first down-scaling x, then up-scaling the result. From (18), this signal is at scale 1 and resolution 1/2. The operator A is therefore called the *approximation operator at half the resolution.* Note that A does not reduce to a filter because it is not shift-invariant. Its flow graph is depicted in Fig. 3.

Characterization of scaled signals. It remains to be found under which conditions the scale and resolution parameter *uniquely* determine the scaled versions of the original signal $\{x_n\}$. That is, among all possible scaled versions of $\{x_n\}$ obtained using up and/or down-scaling operators, there should be only one scaled version at a given scale s and resolution r. We give several equivalent conditions.

It is proven in Appendix B that each of the following statements implies the others:

(i) A scaled version of $\{x_n\}$ at scale $s = 2^{-i}$ and resolution $r = 2^{-j}$ is unique. It is given by

$$(\mathbf{G}\uparrow)^{j-i}\,(\downarrow\mathbf{G}')^{j}\,x. \tag{20}$$

(ii) The approximation operator at half the resolution (19) is a *projector*, i.e.,

$$A^2 = A. \tag{21}$$

(iii) Up-scaling followed by down-scaling leaves the signal unchanged, i.e.,

$$\downarrow\mathbf{G}'\,\mathbf{G}\uparrow = \mathbf{I}. \tag{22}$$

(iv) The two families of shifted scaling impulse responses $\{g_{n-2k}\}$ and $\{\tilde{g}'_{n-2k}\}$, indexed by k, are mutually *biorthonormal.* (Recall that $\tilde{g}'$ is the paraconjugate (11) of g'.) This means that

$$\langle g_{n-2k}\,;\,\tilde{g}'_{n-2l}\rangle = \sum_n g_{n-2k}\,\tilde{g}'^{*}_{n-2l} = \begin{cases}1 & \text{if } k=l\\ 0 & \text{if } k\neq l\end{cases} \tag{23}$$

Besides being equivalent to (i), conditions (ii)-(iv) can be interpreted as follows. First, (21) states that there is no use in re-approximating the approximation Ax of x at resolution 1/2. Equation (22) means that $A^2 = \mathbf{G}\uparrow\downarrow\mathbf{G}'\ \mathbf{G}\uparrow\downarrow\mathbf{G}'$ indeed reduces to $\mathbf{G}\uparrow\downarrow\mathbf{G}' = A$ by simplification of the middle term $\downarrow\mathbf{G}'\ \mathbf{G}\uparrow = \mathbf{I}$. This is illustrated using flow graphs in Fig. 3 (b).

In addition, (20), (22) implies that the down-scaling operator ↓G' halves both scale and resolution only if it applies to signals such as x itself (18a), which have same scale and resolution; however, for "over-scaled" signals —such as Ax— which have higher scales than resolutions, the operator ↓G' only halves scale and leaves resolution unchanged.

The biorthonormality property (23) deserves attention. First, note that "biorthonormality" is just a short-hand for: "scaled versions of the original signal $\{x_n\}$ are uniquely determined by their scale and resolution parameters." Therefore, biorthonormality is not something optional: it should always hold in "coherent" multiresolution systems, for which multiresolution approximations are unique. It will be therefore generally assumed in this paper. What is optional is *orthonormality*, a special case of biorthonormality for which one further imposes

$$g_n = \tilde{g}'_n \,, \tag{24}$$

so that the family of shifted signals $\{g_{n-2k}\}$ form an *orthonormal* set (see (23))

$$\langle g_{n-2k} ; g_{n-2l} \rangle = \sum_n g_{n-2k}\, g^*_{n-2l} = \begin{cases} 1 & \text{if } k=l \\ 0 & \text{if } k \neq l \end{cases} \tag{25}$$

Note that from (24) and the discussion of section II, orthogonality is again a shorthand for the combination of two properties: biorthogonality, plus "up and down scaling operators are Hermitian transpose of eachother."

IV. Multiresolution Residue Signals and Pyramid Transforms

The aim of this section is to give a precise idea of what a *mutiresolution signal decomposition* can be, based on the definitions and properties of scale and resolution discussed in the preceding sections. We assume that scale and resolution characterizes scaled signals as discussed in the last section (conditions (21)-(23)). Intuitively, in a *multiresolution analysis*, the original signal $\{x_n\}$ is decomposed into several multiresolution components associated to different resolutions. During *synthesis*, the signal is reconstructed from its multiresolution components.

From (20), (22), a signal at some resolution r contains all the necessary information to obtain versions at lower resolutions $\leq r$. Therefore, to avoid redundancy of information in a multiresolution analysis, the signal is decomposed into *residue signals* that catch "details" from one resolution to the next finer one. These residue signals are defined by difference as follows.

Assuming that

$$s \geq 2r, \tag{26}$$

the *residue signal* of $\{x_n\}$ at scale s and resolution r is the signal that doubles the resolution (i.e., increases resolution from r to $2r$) when added to the scaled version of $\{x_n\}$ at scale s and resolution r.

$$\begin{array}{ccccc} \text{scaled signal} & + & \text{residue signal} & = & \text{scaled signal} \\ (\text{scale } s, \text{ resolution } r) & & (\text{scale } s, \text{ resolution } r) & & (\text{scale } s, \text{ resolution } 2r) \end{array} \tag{27}$$

Note that, from (26), the right-hand side of (27) is well defined according to (16).

Therefore, a *multiresolution decomposition* of an original signal $\{x_n\}$ is a collection of residue signals at sucessive resolutions 1/2, 1/4, 1/8, ... These residue signals are computed during *multiresolution analysis*. During *multiresolution synthesis*, the signal $\{x_n\}$ is reconstructed starting from a low resolution scaled version of $\{x_n\}$, by applying (27) iteratively to increase resolution until resolution 1 is reached. Note that with this definition, a multiresolution decomposition differs from another only by the *scales* of multiresolution components.

The *Pyramid Transform* is a direct application of these ideas. It was first introduced by Burt and Adelson [5] for image coding purposes; we describe here pyramid decompositions for one-dimensional signals in the framework of this paper. A *Pyramid Transform* on J "octaves" decomposes the signal $\{x_n\}$ into the collection of residues signals at scale $2^{-(j-1)}$ and resolution 2^{-j}, where j=1, ..., J, plus a low resolution version of $\{x_n\}$, namely the scaled signal at scale and resolution 2^{-J}. This description is sufficient to fully describe a Pyramid Transform.

It is easy to connect this to the well-known, original description of Burt and Adleson [5], by deriving a computational tree-structure for its implementation: Start with one step of decomposition, i.e., let J=1. From the above definition, the original signal x is decomposed into two components: its residue signal at scale 1 and resolution 1/2, which from (19), (27) is

$$x - Ax = x - \mathrm{G}\uparrow\downarrow\mathrm{G}'x, \tag{28}$$

plus its scaled version at scale and resolution 1/2, i.e., $\downarrow\mathrm{G}'x$. The corresponding flow graph is depicted in Fig. 4 (a). To reconstruct x, $\downarrow\mathrm{G}'x$ is brought back to scale 1 and (27) is applied, i.e.,

$$\mathrm{G}\uparrow(\downarrow\mathrm{G}'x) + (x - \mathrm{G}\uparrow\downarrow\mathrm{G}'x) = x. \tag{29}$$

In other words, one simply adds what has been previously substracted! The corresponding flow graph is depicted in Fig. 4 (b). This decomposition readily extends to a full computation of a Pyramid Transform on J octaves. Simply note that from the definition above, both scale and resolution parameters of multiresolution residue signals are halved at each stage (octave). Now, since any residue signal is a difference of *scaled* versions of x, the rules (18) apply for residue signals as well. Therefore, scale and resolution parameters of a multiresolution residue signal are halved when replacing x by $\downarrow\mathrm{G}'x$. This amounts to iterate the basic computational structure (one step decomposition) of Fig. 4 (a), (b) on $(\downarrow G')^j x$ at each step (octave) j=0, ..., J-1. This gives the flow graph of Fig. 4 (c).

In Burt and Adelson's terminology [5], the above multiresolution *residue* signals form a *Laplacian Pyramid*, while the set of versions of $\{x_n\}$ at scales 2^{-i} (i=0, ..., J) is called a *Gaussian Pyramid.* The terminology "pyramid" comes from the fact that multiresolution components are computed at successive *scales*, from scale 1 ("base" of pyramid") to scale $2^{-(J-1)}$ ("top" of pyramid). "Gaussian" and "Laplacian" were named after the type of scaling filters used in [5].

Note that from the discussion above, perfect reconstruction is always vouched since one adds backs what has been substracted (see (29)). Therefore there is no constraint at all on scaling filters g_n and g'_n. Even the basic biorthonormality constraint (23), which ensures uniqueness of scaled versions of x at a given scale, is not necessary for the scheme to work !

There is a price to pay, however: the multiresolution residue signals live at scales $s=2^{-i}$ (i=0, ..., J-1) that are always *twice* their resolution. Therefore the transform is overcomplete: starting from an original signal sampled at rate $1/T$, the multiresolution components of a Pyramid Transform are sampled at rate $1/T\left(\sum_{i=0}^{J-1} 2^{-i} + 2^{-J}\right) \approx 2/T$. This means that there are about twice as many transform coefficients than the original signal samples! (In two dimensions this factor becomes 4/3 [1], [6], [11].)

In contrast with the Pyramid Transform, the Discrete Wavelet Transform, presented next, is not overcomplete (there are as many wavelet coefficients as signal samples) but requires design constraints on scaling filters.

V. The Discrete Wavelet Transform and Perfect Reconstruction Filter Banks

We have seen in the preceding section that a potential drawback of the Pyramid Transform is its overcompleteness, due to the fact that residue signals involved are "over-scaled", i.e., their scale parameter is always twice the resolution parameter. In a *Discrete Wavelet Transform* (DWT), each residue signal is "critically sampled," i.e., its resulting scale and resolution parameters are equal. To describe the DWT we therefore need to extend the basic definition of residue signals (27) to the case where scale and resolution parameters are equal (thus violating the previous restriction (26)).

· This can be easily done by considering a perfect reconstruction two-band filter bank [24], [30], [32] depicted in Fig. 5 (a), in which the approximation operator at half the resolution (see Fig. 3 (a)) explicitly appears. The perfect reconstruction condition can be rewritten in operator's notation as

$$x - \mathbf{G}\!\uparrow\, \downarrow\!\mathbf{G}'x = \mathbf{H}\!\uparrow\, \downarrow\!\mathbf{H}'x, \tag{30}$$

where $\mathbf{H}\!\uparrow$ ($\downarrow\!\mathbf{H}'$, respectively) is defined similarly as $\mathbf{G}\!\uparrow$ ($\downarrow\!\mathbf{G}'$, respectively), but with different filter impulse responses h_n (h'_n, respectively). It is a well-known fact [24], [30], [32] that these filters should be high-pass, since the scaling filters g_n and g'_n are assumed low-pass.

Now, (30) is recognized as the residue signal of $\{x_n\}$ at scale 1 and resolution 1/2 (28). We can therefore define $\downarrow\!\mathbf{H}'x$ as the residue signal of $\{x_n\}$ at scale 1/2 and resolution 1/2. It is brought back to scale 1 by applying $\mathbf{H}\!\uparrow$ to give (30). This definition is immediately extended to residue signals at other common values of scale and resolution by application of the rules (18) as in the preceding section. This gives

$$\downarrow\!\mathbf{H}'(\downarrow\!\mathbf{G}')^{j-1}x \tag{31}$$

as the residue signal at scale and resolution 2^{-j} (j>0).

Using this extended definition of multiresolution residue signals, we are now ready to define the Discrete Wavelet Transform (DWT): A DWT of a signal $\{x_n\}$ on J "octaves" decomposes it into "wavelet coefficients" $\{w^j_n\}$, which are precisely the residue signals (31) at scale and resolution 2^{-j}, for j=1, ..., J, *plus* a low resolution version of $\{x_n\}$, namely, the scaled signal at scale and resolution 2^{-J} called $\{v^J_n\}$. To reconstruct the signal $\{x_n\}$, residue signals are first up-scaled by means of $\mathbf{H}\!\uparrow$ (as in (30)), then the basic definition (27) is applied to iteratively increase the resolution until resolution 1 is reached, i.e., until the original signal $\{x_n\}$ is recovered. This is performed by the Inverse DWT (IDWT).

This definition easily recognized as an octave-band filter bank [30] of Fig. 5(b), since in order to halve both resolution and scale parameters of residues signals at each step j, one just iterates the basic computational cell of Fig. 5 (a) to the scaled version $(\downarrow\!\mathbf{G}')^j x$, according to the rule (18a). In Fig. 5 (b), the analysis filter bank computes the DWT, whereas the synthesis filter bank computes the IDWT. Note that since this filter bank is critically sampled, there are as many computed wavelet coefficients as the signal samples. This can be considered as an improvement compared to the situation encountered in the last section for the Pyramid Transform. However, the four low-pass $\{g_n\}$, $\{g'_n\}$, and high-pass $\{h_n\}$, $\{h'_n\}$ filters are constrained to satisfy the perfect reconstruction property (30).

Since a DWT and an octave-band perfect reconstruction Filter Bank share the same computational structure, it can be argued that nothing is new with the DWT. However, one interest of the DWT is that it provides an alternative formalism, which focuses on temporal multiresolution decomposition rather than on subband frequency decomposition. This formalism can be developped as a signal decomposition using temporal *basis functions* called *wavelets*, as follows.

Define the *basic analysis* (*synthesis*) *scaling sequence* to be $\tilde{g}'_n$ (g_n, respectively). Also define the *basic analysis* (*synthesis*) *wavelet* to be the corresponding high pass impulse responses, i.e., $\tilde{h}'_n$ (h_n, respectively). The whole set of *scaling sequences* and *wavelets* is obtained from the basic scaling sequences and wavelets by successive up-scalings:

$$\begin{aligned} &\textit{analysis scaling sequences: } \tilde{g}'^j = (\tilde{G}'\uparrow)^{j-1}\,\tilde{g}' \\ &\textit{analysis wavelets: } \tilde{h}'^j = (\tilde{G}'\uparrow)^{j-1}\,\tilde{h}' \\[1em] &\textit{synthesis scaling sequences: } g^j = (G\uparrow)^{j-1}\,g \\ &\textit{synthesis wavelets: } h^j = (G\uparrow)^{j-1}\,h \end{aligned} \tag{32}$$

Then we have the following, proven in Appendix C: The wavelet coefficients w^j_k of the original signal $\{x_n\}$ at octave j (j=1, ..., J), that is the residue signal at scale and resolution 2^{-j}, are *inner products* (12) of x with the corresponding *analysis wavelets*:

$$w^j_k = \langle x_n\,;\,\tilde{h}'^j_{n-2^jk}\rangle = \sum_n x_n\,(\tilde{h}'^j_{n-2^jk})^*, \quad j=1, \ldots, J \tag{33a}$$

Similarly the low resolution component is

$$v^J_k = \langle x_n\,;\,\tilde{g}'^J_{n-2^Jk}\rangle \tag{33b}$$

Thus, the DWT on J octaves computes the inner products (33). Then, the Inverse DWT (IDWT) reconstructs the signal as a linear combination of shifted *synthesis wavelets* weighted by the corresponding wavelet coefficients, plus a very low resolution approximation of x, which is obtained similarly:

$$x_n = \sum_{j=1}^{J}\sum_k w^j_k\,h^j_{n-2^jk} + \sum_k v^J_k\,g^J_{n-2^Jk}. \tag{34}$$

Equations (33)-(34), proven in Appendix C, are simply a rewriting of the previous definition of the DWT/IDWT. They show that wavelet basis sequences underly a multiresolution decomposition: for example, each wavelet $h^j_{n-2^jk}$ — shifted in proportion to their scale, as is usual [8] — contributes to represent x at a given resolution 2^{-j}, around a given location $k2^{-j}$ in the signal.

VI. Biorthonormal and Orthonormal Discrete Wavelets

The biorthonormality property (23) derived in section III ensures that scale and resolution parameters uniquely determine a scaled or residue version of an original signal $\{x_n\}$. In a DWT, we moreover have the constraint of perfect reconstruction (without delay) (30) of the two-band filter bank of Fig. 5 (a). Because of perfect reconstruction, the biorthonormality property can be extended to *wavelets*: in fact, restricting to FIR filters for convenience, it is shown in Appendix D that the four filters g, g', h, h' are related by

$$\begin{cases} h_n = (-1)^n g'_{n-1} \\ h'_n = (-1)^n g_{n+1} \end{cases} \tag{35}$$

(When the filters are moreover assumed causal, this set of equations is slightly affected and induces a delay on the output of the filter bank.) Therefore, to actually design a DWT it is sufficient to find two low-pass filters satisfying biorthonormality (23) and to assign high-pass filters by (35). Now, using (35) and biorthonormality (23), it turns out that the two families of analysis and synthesis *wavelets* are mutually *biorthonormal* not only across time-shifts at a given scale, but also *across scales*:

$$\langle h^j_{n-2^j k} \; ; \; \tilde{h}'^i_{n-2^i l} \rangle = \sum_n h^j_{n-2^j k} \, (\tilde{h}'^i_{n-2^i l})^* = \begin{cases} 1 & \text{if } k=l \text{ and } i=j \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

A proof is given in Appendix D. In fact Appendix D shows more, namely that biorthonormality (23), hence (36) is already implied by perfect reconstruction. Thus, biorthonormality always occurs in a DWT because it uses a perfect reconstruction filter bank.

Biorthonormality of wavelets (36) is perhaps easier to undertand if we note that, in (34), the term contributing to resolution 2^{-j} is

$$\sum_k \langle x_n ; \; \tilde{h}'^j_{n-2^j k} \rangle \, h^j_{n-2^j k}. \tag{37}$$

Now (36) allows (37) to be a *projection* of x onto the subspace of signals spanned by the wavelets $\{h^j_{n-2^j k}\}$, since if x belongs to this subspace, then using (36) one finds that (37) reduces to x. In other words, owing to biorthonormality, the DWT decomposes a signal into projections of this signal onto subspaces corresponding to different resolutions 2^{-j} (j=1, ..., J). This point of view is close to the original "Multiresolution Analysis" theory of Meyer [23] and Mallat [22].

The Orthonormal Case.

We have seen in section III that the orthonormal case is a special case of biorthonormality for which one further imposes $g_n = \tilde{g}'_n$ .From (35), this implies

$$h_n = \tilde{h}'_n \; . \tag{38}$$

Thus, analysis and synthesis wavelets are equal. From the discussion of section II, it follows that the analysis and synthesis filter banks of Fig. 5 (b) which perform the DWT and the IDWT respectively, are Hermitian transpose of eachother. In other words the DWT is an orthogonal transform: it is composed of *lossless* or *paraunitary* two-band filter banks [31] for which analysis and synthesis filters are paraconjugate of eachother.

From (38), (36), the wavelets form an *orthonormal basis*:

$$\langle h^j_{n-2^jk} \; ; \; h^i_{n-2^il} \rangle = \sum_n h^j_{n-2^jk} \, (h^i_{n-2^il})^* = \begin{cases} 1 & \text{if } k=l \text{ and } i=j \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

hence (37) is now an *orthogonal* projection of x. This means that among all signals y belonging to the subspace spanned by the $\{h^j_{n-2^jk}\}$, (37) is the one that minimizes the quadratic distance $\|x - y\|^2$. Therefore, orthonormality can also be thought of as the condition for which multiresolution components are *most similar* to the original signal [21].

Orthonormality is often considered as an essential property for coding applications (see e.g., [21]), where the coding strategy is based on an LMS error criteria. However it is well known that orthonormal filters cannot be *linear phase*, except for a trivial choice [11], [35]. Note that orthonormality also simplifies the design: scaling and wavelet sequences are easily deduced from each other since (35) reduces to $h_n = (-1)^n g^*_{1-n}$.

VII. Comparison with Wavelet Series

There is a remarkable parallelism between the DWT, presented in the preceding section, and its countinuous-time counterpart, which has been developped for functions of a continuously varying parameter by Mallat [21], [22], Meyer [23], Daubechies [11], [7] and other authors. We refer to the latter as the *orthonormal* or *biorthonormal wavelet series decomposition*. Of course, this analog model uses a continuous version of the inner product (12), namely

$$\langle x(t)\,;y(t)\rangle = \int x(t)y^{*}(t)dt.. \tag{40}$$

and applies to analog signals $x(t)$ which have finite energy $\langle x(t)\,;x(t)\rangle$. Also, up-scaling is the simple dilation $x(t) \rightarrow \frac{1}{\sqrt{2}}\, x(t/2)$ (in the presentation of the DWT above, the constant $\sqrt{2}$ has been integrated into the discrete scaling sequences). As for the DWT, one defines analysis and synthesis basic scaling functions, denoted by $\tilde{\phi}(t)$ and $\phi(t)$, and analysis and synthesis basic wavelets $\tilde{\psi}(t)$ and $\psi(t)$. The whole set of scaling functions and wavelets are deduced as in (32), by successive up-scalings. For example, the synthesis wavelets are $\psi^{j}(t) = 2^{-j/2}\,\psi(2^{-j}t)$. The other basis functions involved, namely $\tilde{\phi}^{j}(t)$, $\tilde{\psi}^{j}(t)$, $\phi^{j}(t)$ are defined similarly. The wavelet series coefficients are (compare with (33a), (33b))

$$\begin{aligned} W^{j}_{k} &= \langle x(t)\,;\tilde{\psi}^{j}(t\text{-}2^{j}k)\rangle \\ V^{J}_{k} &= \langle x(t)\,;\tilde{\phi}^{J}(t\text{-}2^{J}k)\rangle. \end{aligned} \tag{41}$$

The reconstruction part uses a wavelet series (compare with (34))

$$x(t) = \sum_{j=1}^{J}\sum_{k} W^{j}_{k}\ \psi^{j}(t\text{-}2^{j}k) + \sum_{k} V^{J}_{k}\ \phi^{J}(t\text{-}2^{J}k). \tag{42}$$

Analog wavelets are biorthonormal (compare with (36))

$$\langle \psi^{j}(t\text{-}2^{j}k)\,;\tilde{\psi}^{i}(t\text{-}2^{i}l)\rangle = \begin{cases} 1 & \text{if } k=l \text{ and } i=j \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

and, of course, orthonormality occurs when analysis and synthesis wavelets are equal. Other properties such as symmetry or linear phase [7], [34], [35], finite length or "compact support" [7], [11], [35] are also expressed equivalently as they would be in the discrete-time case.

Therefore, the sole ability of the wavelet transform to do multiresolution signal decomposition, using orthonormal or biorthonormal bases, should not be determining to decide whether to choose the discrete-time model or the continuous-time one, because *both models share the same properties.*

It has been argued, however, that the *regularity* property is an exception to this rule: continuous-time wavelets are said to be *regular* if they are at least continuous, possibly with several continuous derivatives. Evidently this cannot be expressed directly on discrete-time signals. However we show in the next section that the regularity property *can* be expressed for discrete-time wavelets, and that the effect on the multiresolution analysis and synthesis are equivalent to what happens in the continuous-time case. Therefore, even the regularity property is no exception.

Because of the strong parallelism that exists between the discrete-time and the continuous-time formalisms, it can be shown [7] that a DWT can always be deduced from a biorthonormal analog wavelet basis. More precisely, the discrete counterparts are defined by the relations [7]

$$\begin{aligned} \widehat{\phi}(t) &= \sqrt{2}\sum_n \tilde{g}'_n\, \widehat{\phi}(2t-n), & \widehat{\psi}(t) &= \sqrt{2}\sum_n \tilde{h}'_n\, \widehat{\psi}(2t-n), \\ \phi(t) &= \sqrt{2}\sum_n g_n\, \phi(2t-n), & \psi(t) &= \sqrt{2}\sum_n h_n\, \psi(2t-n). \end{aligned} \tag{44}$$

Furthermore, additional properties satisfied by continuous-time wavelets, such as orthonormality, symmetry, and finite length are automatically satisfied by the associated discrete-time wavelets [11], [7]. Thus, a continuous wavelet series decomposition scheme induces an associated DWT, the latter being implemented as an octave-band Filter Bank. Moreover, fed by the discrete input $\{V^0_k\}$, this filter bank provides the continuous-time wavelet series coefficients $\{W^j_k\}$ (j=1, ..., J) and $\{V^J_k\}$ [7]. The algorithm was first derived by Mallat [21] in the context of orthonormal wavelets (see also [26], [27]). However, any arbitrary DWT cannot always be deduced from a Wavelet Series decomposition scheme, because this association implies that the obtained discrete wavelets satisfy other constraints than (23), (35). For example, they must satisfy the relations [7]

$$\sum_n h_n = \sum_n \tilde{h}'_n = 0, \tag{45}$$

which are not always met in perfect reconstruction filter banks [24], [30].

This has motivated several researchers [7], [19] to determine the minimal conditions under which the converse deduction holds, i.e. under which continuous-time wavelet bases can be deduced from discrete-time wavelet bases. Necessary and sufficient conditions were recently derived, which turn out to be quite technical [7], [19]. It was previously shown [11], however, that a sufficient condition for the equivalence between the continuous-time case and the discrete-time case is the regularity property, which is by itself interesting; it is discussed in the next section.

VIII. Filter Regularity

The *regularity* property was first required on continuous-time wavelets in order that analysis and synthesis be performed with "smooth" basis functions [8], [23]. Necessary conditions were then derived on discrete FIR filters involved in a DWT, so that they generate regular, compactly supported continuous-time wavelets [11], [7]. Therefore, the regularity property, brought by wavelet theory, was soon recognized as a new design constraint for perfect reconstruction filter banks. The regularity property has also been observed in Pyramid Transforms [5].

The aim of this section is to show that, in multiresolution schemes, regularity can be properly expressed only in terms of discrete signals. We then make the connection with continuous-time basis functions and show how regularity can be taken into account in a filter design procedure. The results listed in this section can be found in [11], [13], [14], [28].

Intuitively, a discrete-time signal is "regular," or "smooth" if its samples vary smoothly in time. In the framework of multiresolution decomposition schemes, we give below a definition of "regularity," or "smoothness" for basis sequences obtained by successives *up-scalings*: the goal is to find the conditions on an up-scaling operator $\mathrm{G}\!\uparrow$ (8) under which signals like

$$(\mathrm{G}\!\uparrow)^j x \tag{46}$$

vary smoothly, even for large j. We shall therefore say that the underlying scaling filter $\{g_n\}$ is *regular* if the iterated signals (46) are. In a DWT, for example, regularity of the scaling filter $\{g_n\}$ implies regularity for all *synthesis* discrete wavelets and scaling sequences, since they are defined by successive up-scalings associated to $\{g_n\}$ (32). On the other hand, regularity of *analysis* discrete wavelets and scaling sequences (32) holds if the scaling filter $\{\tilde{g}'_n\}$ is regular. Note that regularity can also be studied for Pyramid Transforms because their equivalent basis functions [5] are also of the form (46).

Regularity is believed to be useful in multiresolution decomposition schemes for several reasons [4], [2]. In a DWT, for example, any error ocurring in a wavelet coefficient w^j_k results, from (34), in a perturbation in the synthesized signal which is equal to the corresponding synthesis wavelet sequence $\{h^j_{n-2^jk}\}$. It is therefore natural to require that this perturbation is smooth, rather than discontinuous, or even fractal-like, as in the example shown in Fig. 6 (a). This may be useful in e.g. image coding applications where a "fractal" perturbation is likely to "strike the eye" much more than a smooth one, for the same SNR level [4]. On the other hand, requiring that the signal is analyzed by smooth basis functions $\{\tilde{h}'^j_n\}$ ensures that no "artificial" discontinuity (i.e., not due to the signal itself) appears in the wavelet coefficients $w^j_k = \langle x_n ; \tilde{h}'^j_{n-2^jk} \rangle$. This should be of interest for compression purposes.

Definition of Regularity.

We study regularity for iterated signals like (46), i.e., for reponses to the operator $(\mathrm{G}\!\uparrow)^j$. We therefore consider the impulse responses

$$g^j_n = ((\mathrm{G}\!\uparrow)^j\,\delta)_n \tag{47}$$

The following definition of regularity for g^j_n is inspired from *Hölder regularity* of continuous-time functions [14], [28]

— For $0<\alpha\leq 1$, g^j_n is said to be regular of order $r=\alpha$ if it satisfies

$$| g^j_{n+1} - g^j_n | \leq c\,(2^{-j})^\alpha \tag{48}$$

where c is a constant independent of j and n. In other words, the *slopes*

$$\Delta g^j_n = (g^j_{n+1} - g^j_n)/2^{-j} \tag{49}$$

of the curve g^j_n plotted against $n2^{-j}$, are constrained to increase less than $2^{j(1-\alpha)}$ as j increases: for $\alpha<1$ they may indefinitely increase (Fig. 6 (b) shows an example which satisfies (48) for α=0.550...[12], [14]); for $\alpha=1$ they are always bounded (see Fig. 6 (c)). Note that the regularity condition (48) is stronger as α increases, thereby imposing smoother time evolutions of the sequence g^j_n.

— In order to extend this definition to higher regularity orders, we impose (48) on the "discrete derivatives" of g^j_n. The first "discrete derivative" of g^j_n is the slopes' sequence (49). The Nth discrete derivative $\Delta^N g^j_n$ is defined by applying N times the difference operator Δ. Now, g^j_n is said to be regular of order $r=N+\alpha$ ($0<\alpha\leq 1$) if its Nth discrete derivative $\Delta^N g^j_n$ is regular of order α, i.e., if

$$| \Delta^N g^j_{n+1} - \Delta^N g^j_n | \leq c\,(2^{-j})^\alpha \tag{50}$$

The example of Fig. 6 (c) is regular of order $r > 2.102$ [28]. Again this definition imposes a stronger condition as r increases; the condition (50) is here imposed on "slopes of slopes" and therefore requires very smooth evolutions of g^j_n.

Connection with Wavelet Series.

It can be shown [28] that regular discrete wavelet and scaling sequences *converge* to continuous-time functions as $j\rightarrow+\infty$. More precisely:

the discrete curves $\{g^j_n\}$ $(\{\tilde{g}^{\prime j}_n\}, \{h^j_n\}, \{\tilde{h}^{\prime j}_n\})$, *when plotted against* $n2^{-j}$, *uniformly converge to* $\phi(t)$ $(\hat{\phi}(t), \psi(t), \hat{\psi}(t)$, *respectively*). (51)

(see [11], [28], for a more rigorous statement).

In addition, these limit functions have the same *Hölder* regularity order r as their discrete counterparts [28], Hölder regularity $r=N+\alpha$ ($0<\alpha\leq 1$) being defined similarly as for discrete-time sequences:

$$| \phi^{(N)}(t+h) - \phi^{(N)}(t) | \leq c\,|h|^\alpha, \tag{52}$$

where $\phi^{(N)}(t)$ is the Nth derivative of $\phi(t)$ (compare with (50)). Note that a Hölder regularity order $r>N$ implies that N derivatives are continuous.

Now, the continuous-time limits $\phi(t)$, $\hat{\phi}(t)$, $\psi(t)$, and $\hat{\psi}(t)$ can be used to define a Wavelet Series decomposition as in section VII. Compactly supported continuous-time wavelets have been designed by this method [11], [7].

Therefore, owing to regularity, the identification between discrete-time and continuous-time wavelet schemes is complete: it is only a matter of preference to say that continuous-time wavelets underly discrete-time ones (44) or vice versa (51). Likewise, the regularity property can be seen either on continuous-time functions (52) or on discrete-time sequences (50).

A First Necessary Condition for Regularity.

Regularity of the scaling filter $\{g_n\}$ also implies *shape preservation* by the up-scaling operator $\mathrm{G}\uparrow$ (see assumption (c) of section I). In other words, up-scaling dilates the sequence $\{g^j_n\}$ but does not affect its shape: we have $g^j_n \cong g^{j+1}_{2n} \cong g^{j+1}_{2n+1}$, the approximations being sharper as j increases. Now since $g^{j+1} = \mathrm{G}\uparrow g^j$, using (8), we have $g^{j+1}_{2n} = \sum_{k=0}^{L-1} g_{2k}\, g^j_{n-k} \cong (\sum_{k=0}^{L-1} g_{2k})\, g^j_n$, where L is the length of the scaling filter $\{g_n\}$. Therefore $\sum_k g_{2k} = 1$. Similarly one proves $\sum_k g_{2k+1} = 1$, starting from g^{j+1}_{2n+1} ([28] contains a more rigorous proof). These conditions are necessary conditions for shape preservation and for regularity. They can be written as

$$\sum_n g_n = 2 \text{ and } \sum_n (-1)^n g_n = 0 \tag{53}$$

or as

$$G(e^{i\omega})\,|_{\omega=0} = 2 \text{ and } G(e^{i\omega})\,|_{\omega=\pi} = 0, \tag{54}$$

where $G(e^{i\omega})$ is the transfert function of the scaling filter of impulse response $\{g_n\}$. This justifies that scaling filters are preferably *low-pass*.

Note that the first condition in (53) is simply a renormalization. In fact, in a DWT, one generally imposes [11], [7] $\sum_n g_n = \sum_n \tilde{g}'_n = \sqrt{2}$ so that biorthonormality (23) holds. Hence, the order of magnitude of the $\{g^j_n\}$ decreases as $2^{-j/2}$ as j increases. This does not mean that the scaling sequences $\{g^j_n\}$ involved in a DWT are not regular; simply, one has to renormalize $\{g_n\}$ according to (53) so that the order of magnitude of $\{g^j_n\}$ is preserved for different j's.

Fig 6 shows several examples of iterated wavelet sequences $\{h^j_n\}$ corresponding to different choices of $\{g_n\}$. In Fig 6 (a), the curve $\{h^j_n\}$ rapidly diverge as j increases. This example does not even satisfy (53), therefore it is not continuous. The example shown in Fig. 6 (b) is the first Daubechies wavelet of length 4 [11]. It satisfies (53) but is clearly not *very* regular; this shows that (53) is not sufficient to obtain a high regularity order. Fig. 6 (c) is a very regular example.(see below).

The example of Fig. 6 (d) is particularly interesting: it corresponds to one of Smith and Barnwell filters derived in [30]. Mathematically speaking, this example is *not* regular because (53) is not met.

However, in this case, the scaling filter has 40 dB attenuation in the stop band, hence the value of $G(e^{i\omega})$ at $\omega=\pi$ is very small (about 10^{-2}). As a result, (53) is "almost" satisfied and the iterated sequences $\{h^j_n\}$ "pretend" to be regular for small j's. For large j's the curves eventually diverge, with strong oscillations near the modes of the wavelet (see Fig. 6 (d)). Although rejected by the mathematical definition above, such wavelets may be "regular enough" in applications where the number of octaves in the multiresolution decomposition is not too large.

Necessary Conditions for Regularity: "Flat Filters" or "Vanishing Moments."

A regularity order $r>N$ requires more than (53), (54). In fact it can be shown [28] that the scaling transfert function $G(e^{i\omega})$ is necessarily of the form

$$G(e^{i\omega}) = 2^{-N}\,(1+e^{-i\omega})^{N+1}\,F(e^{i\omega}), \qquad (55)$$

where $F(1)=1$. Conditions (53), (54) follow for $N=0$. Here (55) implies that up to N derivatives of the transfert function $G(e^{i\omega})$ vanish at $\omega=\pi$, hence a regular scaling filter is "flat" at half the sampling frequency. In a DWT, using (35) one finds

$$\sum_n n^i\,\tilde{h}'_n = 0, \quad i=0,\ \ldots,\ N. \qquad (56)$$

and similarly for $\{h_n\}$ if the analysis scaling filter $\{\tilde{g}'_n\}$ is "flat." Equation (56) means that the wavelet sequence $\{\tilde{h}'_n\}$ has N+1 *vanishing moments* [12], and this can be written similarly for the continuous-time wavelet $\tilde{\psi}(t)$ [12], [22], [23]. For example, the wavelet shown in Fig. 6 (c) has six vanishing moments. This property may be interesting by itself for some applications [12], [22].

The condition (55) or (56) can be easily integrated in a filter design procedure [7], [35]. However, it is only a *necessary* condition for regularity: there exist non regular examples for which (55) holds [28]! It is therefore important to *a posteriori* determine the regularity order of a computed scaling filter $\{g_n\}$. An efficient method is presented next.

Estimation of Regularity.

Several regularity estimates have been derived. Daubechies estimate [11] is based on the determination of maxima of spectra. It may require many computations to determine a good estimate [11], [12] and is not optimal in general [28] (as far as the above definition is concerned). Daubechies and Lagarias have derived a method based on matrix algebra which yields optimal regularity estimates in some instances [14].

Another, easily implementable, method has been derived in [28]. We summarize it here because it is of general applicability. It uses the sequence $\{f_n\}$ associated to $F(e^{i\omega})$ in (55) to compute iterated sequences $f^j = (\mathbf{F}\uparrow)^j\,(\delta)$. Then, for any j, the regularity order is at least

$$r_j = N - \tfrac{1}{j} \log_2 \max_{0 \le m \le 2^j - 1} \sum_n \left| f^j_{2^j n + m} \right| \tag{57}$$

(note that r_j may be less than N). In addition, except for exceptional cases, r_j (rapidly) tends to the optimal regularity order as j increases [28].

Conclusion

The discrete-time multiresolution theory developped in this paper is entirely based on precise notions of *scale* and *resolution* (section I-III). We have described both the Pyramid transform (section IV) and the Discrete Wavelet Transform (section V-VI) using these notions. We have shown that, in the context of discrete-time signals, the DWT is naturally implemented as an octave-band filter bank.

Biorthonormality was derived as an essential condition for scaled versions of an original signal to be characterized by scale and resolution parameters. As explained in section VI, it also naturally arises in a DWT, because the DWT uses perfect reconstruction two-band filter banks. Orthonormality has been derived as a special case of biorthonormality, in which flow graphs are self transposed.

Because this paper develops different multiresolution techniques unified in a common framework, it enlightens either known or little-known links between the DWT and other multiresolution techniques:

Links with Pyramid Transforms. The Pyramid Transform, as described by Burt and Adelson [8], is based on an intuitive notion of multiresolution decomposition of discrete-time signals, and the regularity property has been observed as well. The presentation of this paper therefore applies naturally to Pyramid Transforms (section IV). Note that since a Pyramid Transforms uses "over-scaled" multiresolution components, scale is here clearly distinguished from resolution, both notions being sometimes confused in a DWT.

Links with Continuous-Time Wavelet Series. We have seen that the discrete-time multiresolution theory derived here shares the same rich properties as the continuous-time "multiresolution analysis" theory of Meyer [23]. These include basis expansions, scale and resolution notions, orthogonality [11], [22] or biorthogonality [7], regularity [11], and the like. Therefore, it is only a matter of taste to decide whether analog wavelets underly discrete-time ones or *vice versa*. The DWT hence provides a coherent alternative of seeing things, and both approaches have advantages and drawbacks. For example, a change of scale is evidently not so easily expressed for discrete sequences than it is for continuous-time signals. The discrete approach, however, avoids technical proofs or makes them easier and readily provides numerical algorithms.

Links with Filter Banks. Discrete-time Multiresolution Theory also gives a new way of looking at Filter Banks: it describes them as a temporal multiresolution decomposition rather than as an octave-band frequency decomposition. We have seen that new criteria in filter design are also brought by wavelets. In particular, there is a —presumably important— notion of *regularity*. In section VIII we have briefly discussed the regularity property for both discrete-time and continuous-time wavelets.

Regularity has been taken into account in several FIR filter design procedures, in connection with other, classical criteria such as linear phase [4], [7], [34], [35] or orthonomality [11] (i.e., self transposition — see section VI). Although regular filters have been used in practical systems [4], [4], [21], it is still not clear whether regularity is to play an important role in applications such as image coding. Besides, a good regularity estimate was not easily obtained [11]-[14]. In with respect, the recent estimate (57) might help [35]. Another, major difficulty in filter design is that many design constraints may be desirable (short lengths, linear phase, orthonormality, selectivity in frequency, regularity), and they often conflict with each other. For example, we have mentionned that orthonormality precludes linear phase, except for a trivial choice [11], [35]. It may therefore be necessary to relax most of the constraints in order to design scaling and wavelet filters that are e.g. close to be orthonormal, close to be linear phase, reasonably selective in frequency and with at least a given regularity order. Some results go in this direction: for example, [12], [30] contain orthonormal FIR filters which are close to being linear phase. On the other hand, [4], [7] contain biorthonormal, linear phase FIR filters which are close to being orthonormal.

APPENDIX A

GENERAL EXPRESSIONS FOR SCALING OPERATORS

Proof of (8). Let $\{g_n\}$ be the impulse response of the up-scaling operator, that is the up-scaled version of $\delta_n=1$ if $n=0$, 0 elsewhere. By assumption (b), g_{n-2k} is the response to δ_{n-k}. Since the input signal can be written

$$x_n = \sum_k x_k\, \delta_{n-k},$$

its up-scaled version is, using linearity (a),

$$\sum_k x_k\, g_{n-2k},$$

which is (8).

Proof of (9). Let $\{g^0_n\}$ be the impulse response of the down-scaling operator to $\{\delta_n\}$, and $\{g^1_{n-1}\}$ be the impulse response to $\{\delta_{n-1}\}$. The input signal can be written

$$x_n = \sum_k x_{2k}\,\delta_{n-2k} + x_{2k+1}\,\delta_{n-1-2k}\,.$$

Its down-scaled version is therefore

$$\sum_k x_{2k}\,g^0_{n-k} + x_{2k+1}\,g^1_{n-1-k}\,.$$

Now let $\{g'_n\}$ be defined by $g'_{2n} = g^0_n$ and $g'_{2n+1} = g^1_n$. The down-scaled version becomes

$$\sum_k x_{2k}\,g'_{2n-2k} + x_{2k+1}\,g'_{2n-2k-1}\,,$$

which reduces to (9). ■

APPENDIX B

DERIVATION OF BIORTHONORMALITY

(i)⇒(ii). Since A approximates at half the resolution, it should leave Ax itself unchanged. That is, both Ax and A^2x are at scale 1 and resolution 1/2, so they are equal and (21) holds. ■

(ii)⇒(iii). Equation (21) can be written, using (19),

$$\mathbf{G}\uparrow y = 0,$$

where $y = \downarrow\mathbf{G}'(\mathbf{G}\uparrow\downarrow\mathbf{G}' - \mathbf{I})\,x$. Now since up-scaling is one-to-one (17), this yields $y=0$, that is

$$(\downarrow\mathbf{G}'\ \mathbf{G}\uparrow - \mathbf{I})\downarrow\mathbf{G}'x = 0, \text{ for all signals } x. \tag{B1}$$

We now prove that *any* signal y with finite energy (13) can be written $\downarrow\mathbf{G}'x$ for some x. We can always write

$$y = y' + y'',$$

where y' belongs to the range of the down-scaling operator $\downarrow\mathbf{G}'$ (hence it can be put in the form $\downarrow\mathbf{G}'x$ for some x), and where y'' is orthogonal to any signal of this range:

$$\langle\, y''\,;\downarrow\mathbf{G}'z\,\rangle = 0, \text{ for all signals } z.$$

Using the transposition property (15b) one obtains

$$\langle\, \tilde{\mathbf{G}}'\uparrow y''\,; z\,\rangle = 0 \ \text{ for all } z,$$

hence $\tilde{\mathbf{G}}'\uparrow y'' = 0$, which implies $y''= 0$ similarly as for $\mathbf{G}\uparrow$ (17). Therefore any signal y can be written $y = y' = \downarrow\mathbf{G}'x$ for some x. Now from (B1), equation (22) immediately follows. ■

(iii)⇔(iv) Starting from (22) and the relation

$$\langle \delta_{n-k} ; \delta_{n-l} \rangle = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

and using (15b), we have

$$\langle \delta_{n-k} ; \delta_{n-l} \rangle = \langle \delta_{n-k} ; (\downarrow\mathbf{G}'\ \mathbf{G}\uparrow\delta_{n-l}) \rangle$$

$$= \langle (\tilde{\mathbf{G}}'\uparrow\delta_{n-k}) ; (\mathbf{G}\uparrow\delta_{n-l}) \rangle$$

$$= \langle (\tilde{\mathbf{G}}'\uparrow\delta)_{n-2k} ; (\mathbf{G}\uparrow\delta)_{n-2l} \rangle = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

which is the biorthonormality property (23). ■

(iii)⇒(i) From any possible expression of up-scaled versions of $\{x_n\}$, simplify using (22) each time this is possible to obtain (20). ■

APPENDIX C

DISCRETE WAVELET BASES

Proof of (33): The residue signal at resolution and scale 2^{-j} (31) can be written

$$w^j_k = (\downarrow\mathbf{H}'\ (\downarrow\mathbf{G}')^{j-1}\, x\,)_k \;= \langle \downarrow\mathbf{H}'\ (\downarrow\mathbf{G}')^{j-1}\, x_n \;;\; \delta_{n-k} \rangle$$

Using the transposition property (15b) we have

$$w^j_k = \langle x_n \;;\; (\tilde{\mathbf{G}}'\uparrow)^{j-1}\ \tilde{\mathbf{H}}'\uparrow\ \delta_{n-k} \rangle$$
$$= \langle x_n \;;\; (\,(\tilde{\mathbf{G}}'\uparrow)^{j-1}\ \tilde{h}')_{n-2^j k} \;\rangle$$

which reduces to (33a) by definition of $\tilde{h}'^j$ (32). One proves similarly that $v^J = (\downarrow\mathbf{G}')^J x$ is equivalent to (33b). ■

Proof of (34): To reconstruct the signal, the wavelet coefficients w^j, and v^J are brought back to scale 1 and (27) is applied. This gives

$$x = \sum_{j=1}^{J} (\mathbf{G}\uparrow)^{j-1}\ \mathbf{H}\uparrow w^j \;+\; (\mathbf{G}\uparrow)^J\, v^J$$

Using the formula $w^j_n = \sum_l w^j_k\, \delta_{n-k}$ (and similarly for v^J), linearity and (32), this equation is easily seen to reduce to (34). ■

APPENDIX D

PERFECT RECONSTRUCTION AND BIORTHONORMALITY

Proof of (35) *and biorthonormality* (23), *assuming perfect reconstruction* (30). It is well known [24], [30], [31] that perfect reconstruction of the filter bank of Fig. 5 (a) can be expressed by two conditions, expressing that both aliasing and linear distortion are cancelled at the reconstruction. This can be written in matrix notation, using z-transforms: $X(z) = \sum_n x_n z^{-n}$, as

$$\begin{pmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{pmatrix} \begin{pmatrix} G'(z) \\ H'(z) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

It follows that $G'(z) = 2H(\text{-z})/\Delta(z)$ and $H'(z) = \text{-}2G(\text{-z})/\Delta(z)$, where $\Delta(z)$ is the determinant of the matrix. Assuming FIR analysis filters, $\Delta(z)$ is an odd polynomial in z and z^{-1}: $\Delta(z) = G(z)H(\text{-}z) - H(z)G(\text{-}z)$. In order that synthesis filters be also FIR it is therefore necessary that $\Delta(z)$ be an odd delay. Within normalisation and time-shifts for impulse responses we can write $\Delta(z) = 2z^{-1}$. It follows that $G'(z) = z\,H(\text{-}z)$ and $H'(z) = \text{-}zG(\text{-}z)$, which reduces to (35). Note that whenever a causal implementation is required, one just have to shift finite impulse responses accordingly.

Using the above relations the perfect reconstruction condition $\Delta(z) = 2z^{-1}$ becomes

$$G(z)G'(z) + G(\text{-}z)G'(\text{-}z) = 2$$

This can be interpreted using flow graphs as (22). It was shown in section III that this is equivalent to the biorthonormality condition (23). ■

Proof of (36): We have seen in section III that (23) is equivalent to $\downarrow\mathbf{G'G}\uparrow = \mathbf{I}$. From the relations (35) on similarly proves that

$$\downarrow\mathbf{H'H}\uparrow = \mathbf{I},\ \downarrow\mathbf{H'G}\uparrow = \mathbf{0} \text{ and } \downarrow\mathbf{G'H}\uparrow = \mathbf{0}. \tag{D1}$$

Now, using the definition (32), the left-hand side of (36) can be written

$$\langle\, (\mathbf{G}\uparrow)^{j-1}\,\mathbf{H}\uparrow \delta_{n-k}\,;\ (\tilde{\mathbf{G}}'\uparrow)^{j-1}\,\tilde{\mathbf{H}}'\uparrow \delta_{n-l}\,\rangle$$

Using the transposition property (15b), we obtain

$$\langle\, \downarrow\mathbf{H}'\,(\downarrow\mathbf{G}')^{j-1}\,(\mathbf{G}\uparrow)^{j-1}\,\mathbf{H}\uparrow \delta_{n-k}\,;\ \delta_{n-l}\,\rangle$$

which, from (D1), reduces to the right-hand side of (36). ■

ACKNOWLEDGMENTS

I would like to express my appreciation to I. Daubechies (Bell Laboratories, Murray Hill, NJ), A. Grossmann (CPT, Marseilles-Luminy University) and Y. Meyer (Ceremade, Paris-Dauphine

University) for introducing me to mathematical aspects of wav-elets, when it was still very new. I am also grateful to M. Vetterli and C. Herley (Columbia University, NY) for valuable discussions on Filter Banks, Pyramids and Wavelets, to P. Duhamel (CNET, Paris) for carefully reading the manuscript, and to anonymous reviewers for valuable reports.

REFERENCES

[1] E.H. Adelson, E. Simoncelli and R. Hingorani, "Orthogonal Pyramid Transforms for Image Coding," in *Proc SPIE Visual Communications and Image Proc.*, Cambridge, MA, Vol. 845, pp.50-58, Oct. 27-29, 1987.

[2] A.N. Akansu, R.A. Haddad, and H. Caglar, "Binomial QMF-Wavelet Transform," in *Proc. 1990 Digital Signal Processing Workshop*, New Paltz, NY, pp. 6.10.1-2, September 16-19, 1990.

[3] J.B. Allen and L.R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proc. IEEE*, Vol. 65, No. 11, pp. 1558-1564, 1977.

[4] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Vector Quantization in the Wavelet Transform Domain," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, Apr.3-6, 1990, pp. 2297-2300.

[5] P.J. Burt and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Communications*, Vol.COM-31, No.4, April 1983.

[6] P.J. Burt, "Multiresolution Techniques for Image Representation, Analysis, and 'Smart' Transmission," *SPIE Visual Comm. Image Proc. IV*, Vol.1199, pp.2-15, 1989.

[7] A. Cohen, I. Daubechies and J.C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets," Technical Memo. #11217-900529-07, AT&T Bell Laboratories, Murray Hill.

[8] J.M. Combes, et. al. eds., *Wavelets, Time-Frequency Methods and Phase Space*, Proc. Int. Conf. Marseille, France, Dec. 14-18, 1987, IPTI, Springer Verlag Berlin , 315 pp., 1989.

[9] R.E. Crochiere and A.V. Oppenheim, "Analysis of Linear Digital Networks," *Proc. IEEE*, Vol. 63, pp. 581-595, April 1975.

[10] R.E. Crochiere, S.A. Weber and J.L. Flanagan, "Digital Coding of Speech in Subbands," *Bell Syst. Tech. J.*, Vol. 55, pp. 1069-1085, Oct. 1976.

[11] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. in Pure and Applied Math.*, Vol.41, No.7, pp.909-996, 1988.

[12] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme," Technical Memo. #11217-891116-17, AT&T Bell Laboratories, Murray Hill.

[13] I. Daubechies and J.C. Lagarias, "Two-Scale Difference Equations I.Existence and Global Regularity of Solutions,"*SIAM J.Math. Analysis*, to appear.

[14] I. Daubechies and J.C. Lagarias, "Two-Scale Difference Equations II.Local Regularity, Infinite Products of Matrices and Fractals," *SIAM J.Math. Analysis*, to appear.

[15] P. Goupillaud, A. Grossmann and J. Morlet, "Cycle-Octave and Related Transforms in Seismic Signal Analysis," *Geoexploration*, Vol.23, pp.85-102, Elsevier Science Publishers, B.V. Amsterdam, Netherlands, 1984/85.

[16] A. Grossmann, R. Kronland-Martinet, and J. Morlet, "Reading and Understanding Continuous Wavelet Transforms," in [8], pp.2-20, 1989.

[17] O. Herrmann, "On the Approximation Problem in Nonrecursive Digital Filter Design," *IEEE Trans. Circuit Theory*, Vol. CT-18, No. 3, pp. 411-413, May 1971.

[18] J. Kovacevic and M. Vetterli, "Perfect Reconstruction Filter Banks with Rational Sampling Rates in One and Two Dimensions," *SPIE Image Proc. Visual Comm.*, Nov. 1989.

[19] W.M. Lawton, "Necessary and Sufficient Conditions for Constructing Orthonormal Wavelet Bases," Aware Report. No. AD900402, University Place, 124 Mt. Auburn Street, Cambridge, MA 01238, USA, 1990.

[20] P.G. Lemarié, "Ondelettes à Localisation Exponentielle," [in French] *J.Math. Pures et Appl.*, Vol.67, pp.227-236, 1988.

[21] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.11, No. 7, pp.674-693, July 1989.

[22] S. Mallat, "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$,",*Trans.Amer.Math.Soc.*, Vol.315, No.1, pp.69-87, September 1989.

[23] Y. Meyer, *Ondelettes et Opérateurs, Tome I*, [in French] Herrmann ed., Paris, 1990.

[24] F. Mintzer, "Filters for Distortion-Free Two-Band Multirate Filter Banks," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-33, No.3, pp. 626-630, June 1985.

[25] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, 2nd edition, A.V. Oppenheim, ed., Prentice Hall Signal Processing Series, Englewood Cliffs, N.J., 879 pp., 1989.

[26] O. Rioul, "Fast Algorithms for the Continuous Wavelet Transform," in *Proc.1991 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Toronto, Ontario, Canada, May 14-17, 1991.

[27] O. Rioul and P. Duhamel, "Fast Algorithms for Discrete and Continuous Wavelet Transforms," submitted to *IEEE Trans. Info. Theory*, Special issue on Wavelets.

[28] O. Rioul, "Dyadic Up-Scaling Schemes: Simple Criteria for Regularity," submitted to *SIAM J. Math. Anal.*

[29] O. Rioul and M. Vetterli, "Wavelet Transforms in Signal Processing," *IEEE ASSP Magazine*, to appear.

[30] M.J.T. Smith and T.P. Barnwell III, "Exact Reconstruction Techniques for Tree Structured Subband Coders," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol.ASSP-34, No.3, pp.434-441, June 1986.

[31] P.P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, Vol. 78, No.1, pp.56-93, July 1990.

[32] P.P. Vaidyanathan and P-Q. Hoang, "Lattice Structures for Optimal Design and Robust Implementation of Two-Channel Perfect Reconstruction QMF Banks," *IEEE Trans. ASSP.*, Vol.36, No.1, pp. 81-94, January 1988.

[33] M. Vetterli, "Multi-Dimensional Sub-Band Coding: Some Theory and Algorithms," *Signal Processing*, Vol. 6, No. 2, pp. 97-112, Feb. 1984.

[34] M. Vetterli, C. Herley "Wavelets and Filter Banks: Relationships and New Results," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, pp. 1723-1726, Apr. 3-6, 1990.

[35] M. Vetterli and C. Herley "Wavelets and Filter Banks: Theory and Design," submitted to *IEEE Trans. Signal Proc.*

[36] J.E. Younberg and S.F. Boll, "Constant-Q Signal Analysis and Synthesis," *IEEE Int. Conf. on Acoust., Speech, and Signal Proc., ICASSP-78*, Tulsa, OK, pp. 375-378, 1978.

[37] W.R. Zettler, J. Huffman and D.C.P. Linden, "Application of Compactly Supported Wavelets to Image Compression," in *Proc. 1990 SPIE*, Vol.1244 Image Proc. Algorithms and Techniques, Santa Clara, CA, pp.150-160, Feb. 13, 1990.

FIGURE CAPTIONS

Fig. 1. Two successive up-scalings (by a factor 2) of an original discrete-time signal (top of figure). Up-scaling is performed by first-order interpolation (7). Up-scaled signals are stretched in time, but no resolution is added (see section III).

Fig. 2. Flow graphs of up and down scaling operators. (a) Up-scaling (8) is up-sampling plus post-filtering. (b) Down-scaling (9) is pre-filtering plus down-sampling. The filters of impulse response $\{g_n\}$ and $\{g'_n\}$ are called scaling filters.

Fig. 3. A flow-graph illustration of the fact that the approximation Ax of an original signal x at half the resolution is a projection.

(a) Down-scaling, followed by up-scaling approximate x at half its resolution.

(b) Re-approximating Ax by A leaves Ax unchanged. This is equivalent to (22), i.e., to the condition that up-scaling, followed by down-scaling, is the identity operator.

Fig. 4. Flow graph implementation of the Pyramid Transform.

(a) One step of decomposition; the original signal x is decomposed into a version of x at half its scale, and a residue signal at the same scale and half the resolution, the latter being obtained by difference.

(b) The reconstruction part uses (27). Here, one simply adds back what have been substracted.

(c) The elementary cell of Figs. (a), (b) is iterated J times to provide a full Pyramid Transform on J "octaves" (here J=3).

Fig. 5. Flow graph implementation of the Discrete Wavelet Transform (DWT)..

(a) One step of decomposition/reconstruction. The original signal x is decomposed into a version of x at half its scale and resolution, and a residue signal also at half its scale and resolution. This is simply a two-band perfect reconstruction filter bank; the decomposition (analysis) part uses the low-pass "scaling" filter $\{g'_n\}$ and the high-pass "wavelet" filter $\{h'_n\}$. The reconstruction (synthesis) part uses the low-pass "scaling" filter $\{g_n\}$ and the high-pass "wavelet" filter $\{h_n\}$

(b) The elementary cell of Fig. (a) is iterated J times to provide a full DWT on J "octaves."

Fig. 6. Plots of iterated wavelet sequences $\{h^{11}{}_n\}$, plotted against $n2^{-j}$, for four differents choices of the scaling filter $\{g_n\}$.

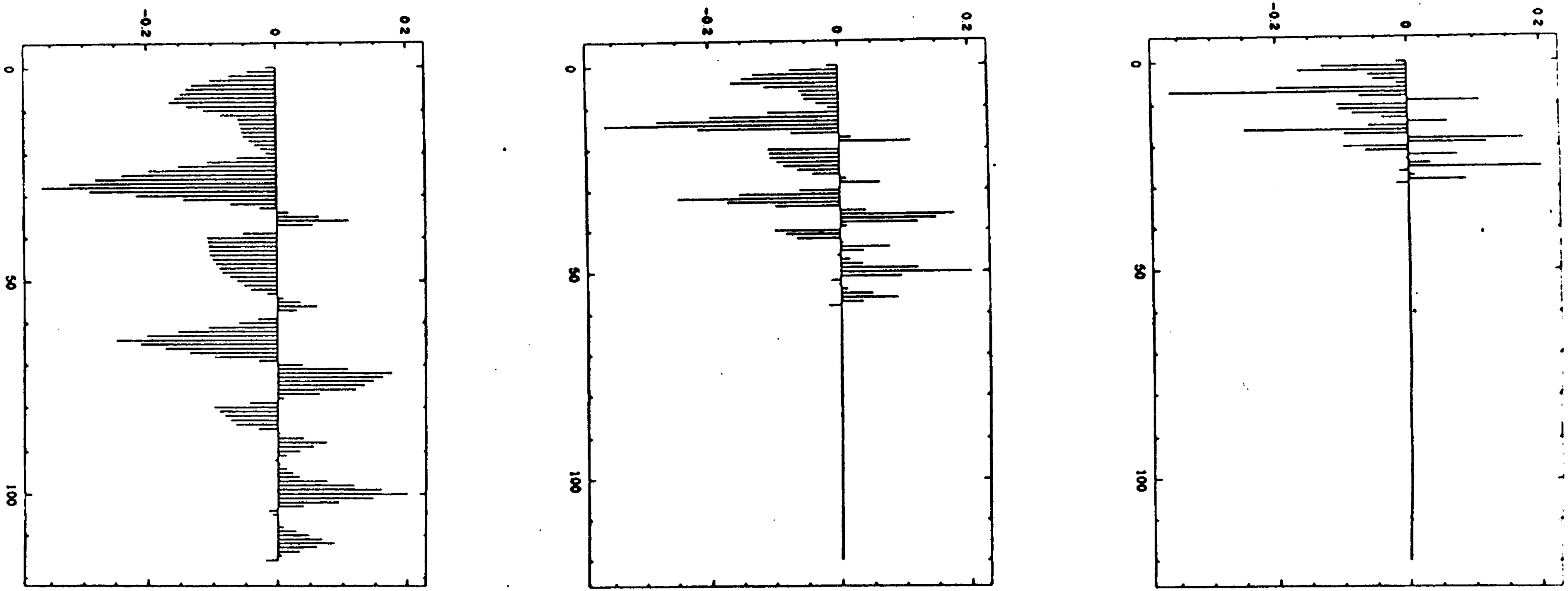
(a) $g_0 = g_1 = .7$, $g_2 = -g_3 = .1$, $g_n = 0$ otherwise. This example satisfies orthonormality (25), and can therefore be used in a perfect reconstruction paraunitary filter bank (see section VI). However, the obtained curve is a highly irregular, fractal function. In fact this curve rapidely diverges as j increases indefinitely.

(b) Daubechies wavelet of length 4: $g_0 = (1+\sqrt{3})/4\sqrt{2}$, $g_1 = (3+\sqrt{3})/4\sqrt{2}$, $g_2 = (3-\sqrt{3})/4\sqrt{2}$, $g_3 = (1-\sqrt{3})/4\sqrt{2}$. (after [11]).The regularity order is $r = 0.5500...$ The obtained curve rapidely converges, as j increases indefinitely, to a continuous, but not differentiable, function.

(c) Daubechies "closest to linear phase" wavelet of length 12 (after [12]). The regularity order, as determined in [28] is at least 2.102.

(d) This example corresponds to the 8-tap scaling filter designed by Smith and Barnwell [30]. Mathematically speaking, the obtained curve is irregular: due to small, but rapid, oscillations present in the wavelet, the curves slowly diverges as j increases indefinitely. However the obtained curves look "reasonably regular" for small j's, due to the fact that the scaling filter's transfert function is strongly attenuated in the stop band (see section VIII). This behavior cannot be predicted by the mathematical model of regularity.

Fig. 1. Two successive up-scalings (by a factor 2) of an original discrete-time signal (top of figure). Up-scaling is performed by first-order interpolation (7). Up-scaled signals are stretched in time, but no resolution is added (see section III).



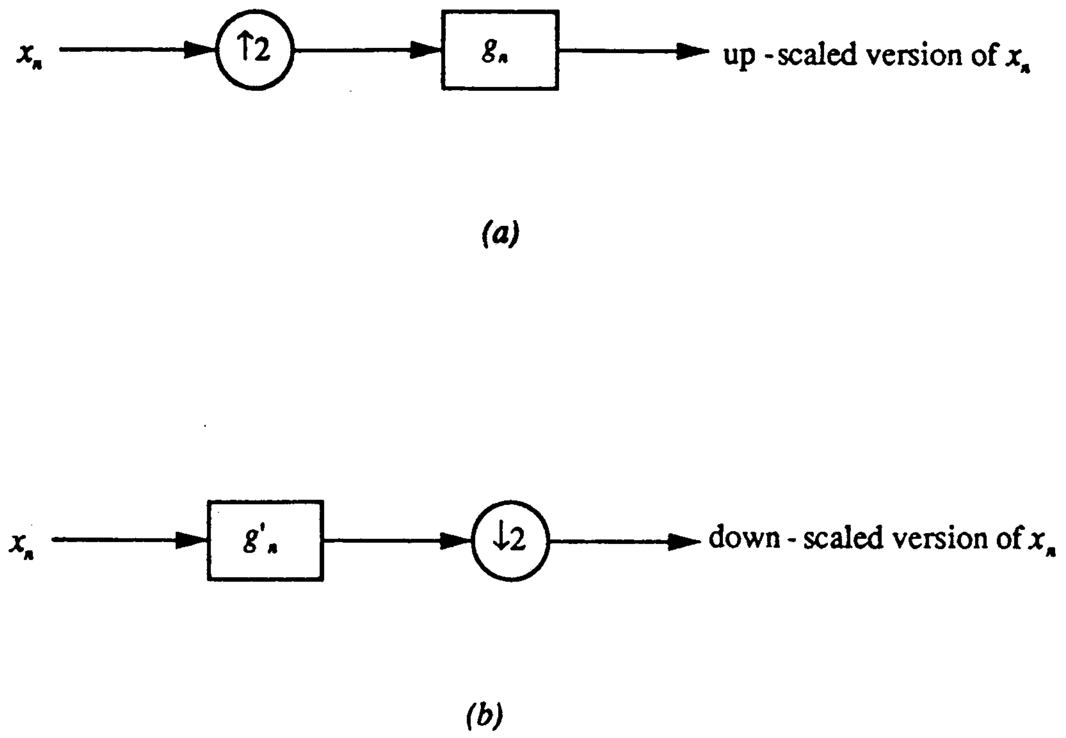


Fig. 2. Flow graphs of up and down scaling operators. (a) Up-scaling (8) is up-sampling plus post-filtering. (b) Down-scaling (9) is pre-filtering plus down-sampling. The filters of impulse response $\{g_n\}$ and $\{g'_n\}$ are called scaling filters.

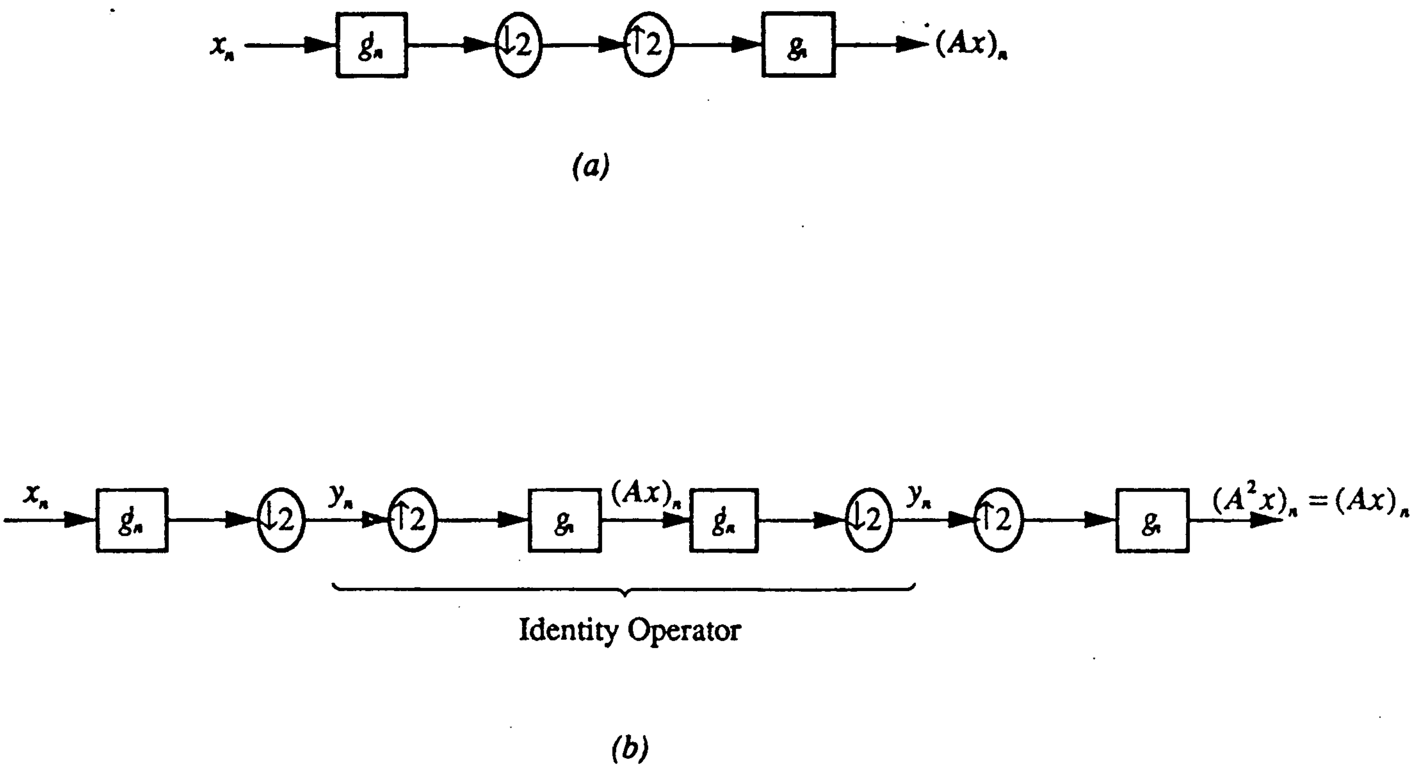


Fig. 3. A flow-graph illustration of the fact that the approximation Ax of an original signal x at half the resolution is a projection.

(a) Down-scaling, followed by up-scaling approximate x at half its resolution.

(b) Re-approximating Ax by A leaves Ax unchanged. This is equivalent to (22), i.e., to the condition that up-scaling, followed by down-scaling, is the identity operator.

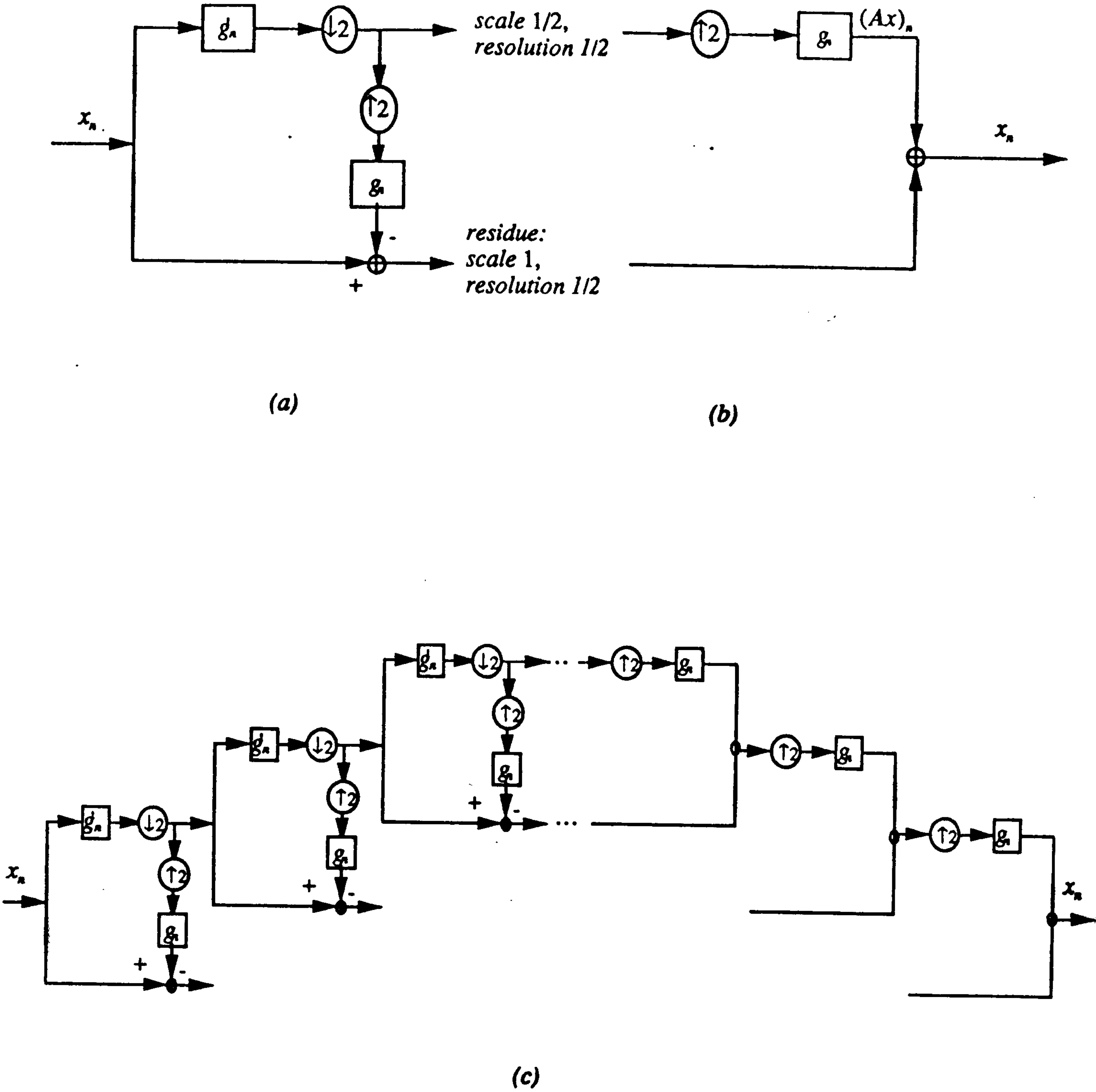


Fig. 4. Flow graph implementation of the Pyramid Transform.

(a) One step of decomposition; the original signal x is decomposed into a version of x at half its scale, and a residue signal at the same scale and half the resolution, the latter being obtained by difference.

(b) The reconstruction part uses (27). Here, one simply adds back what have been substracted.

(c) The elementary cell of Figs. (a), (b) is iterated J times to provide a full Pyramid Transform on J "octaves" (here J=3).

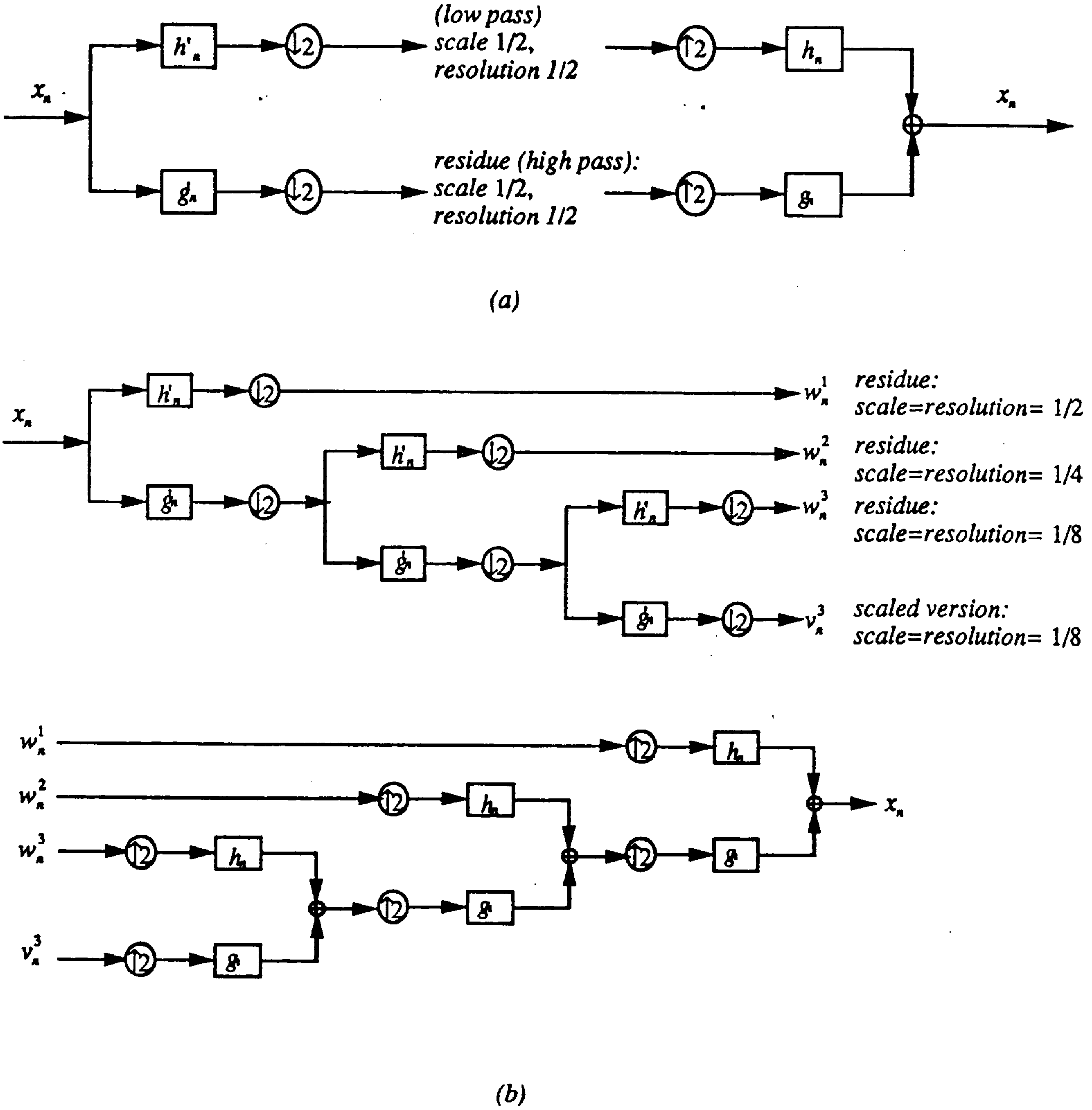


Fig. 5. Flow graph implementation of the Discrete Wavelet Transform (DWT)..

(a) One step of decomposition/reconstruction. The original signal x is decomposed into a version of x at half its scale and resolution, and a residue signal also at half its scale and resolution. This is simply a two-band perfect reconstruction filter bank; the decomposition (analysis) part uses the low-pass "scaling" filter $\{g'_n\}$ and the high-pass "wavelet" filter $\{h'_n\}$. The reconstruction (synthesis) part uses the low-pass "scaling" filter $\{g_n\}$ and the high-pass "wavelet" filter $\{h_n\}$

(b) The elementary cell of Fig. (a) is iterated J times to provide a full DWT on J "octaves."

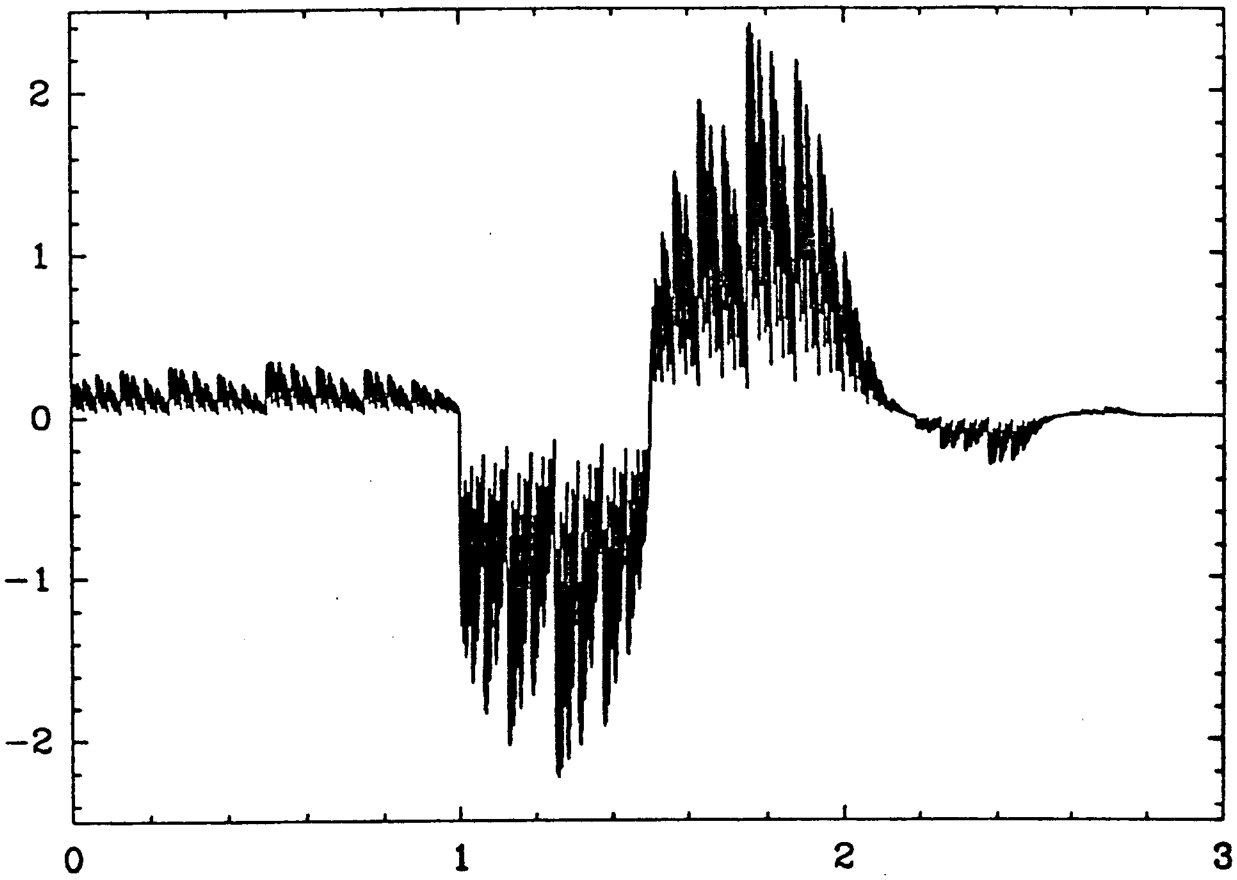
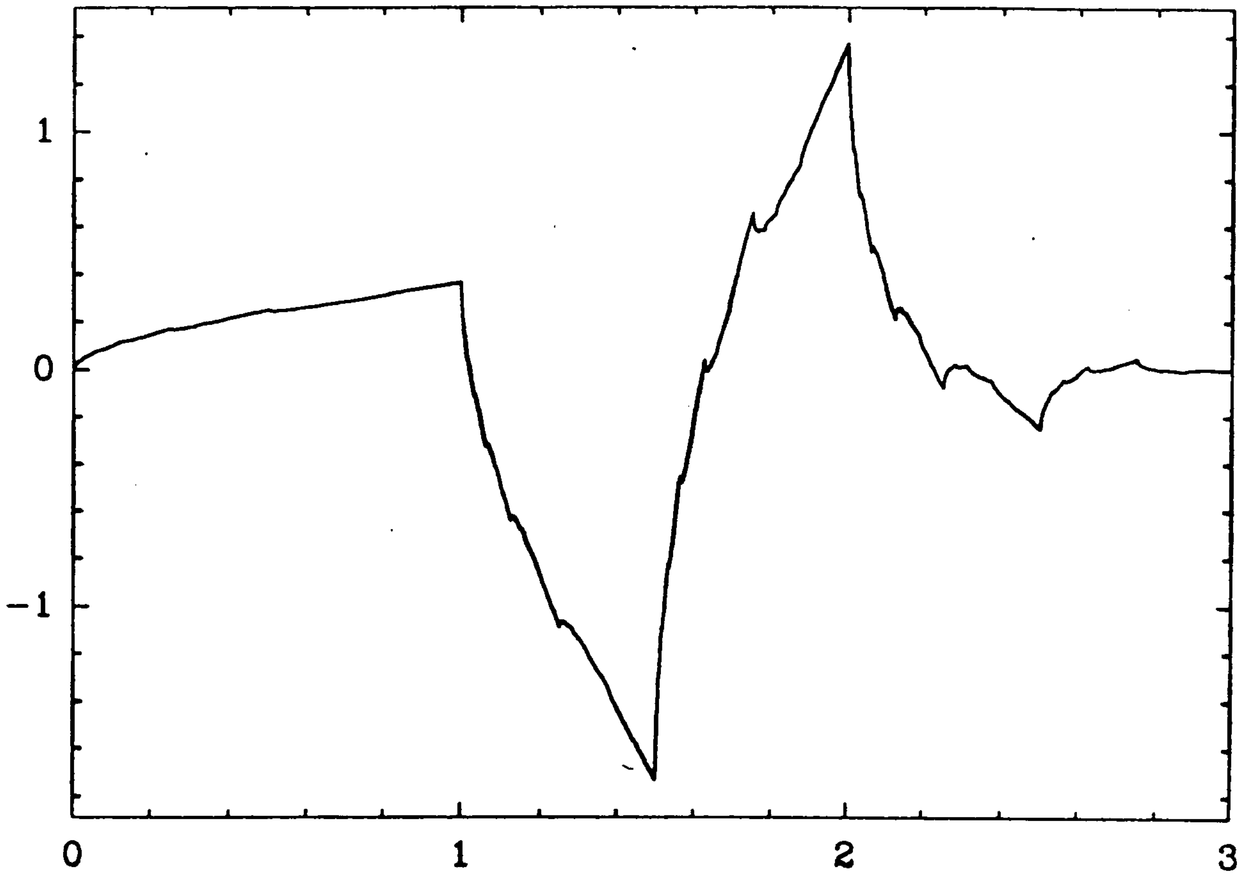
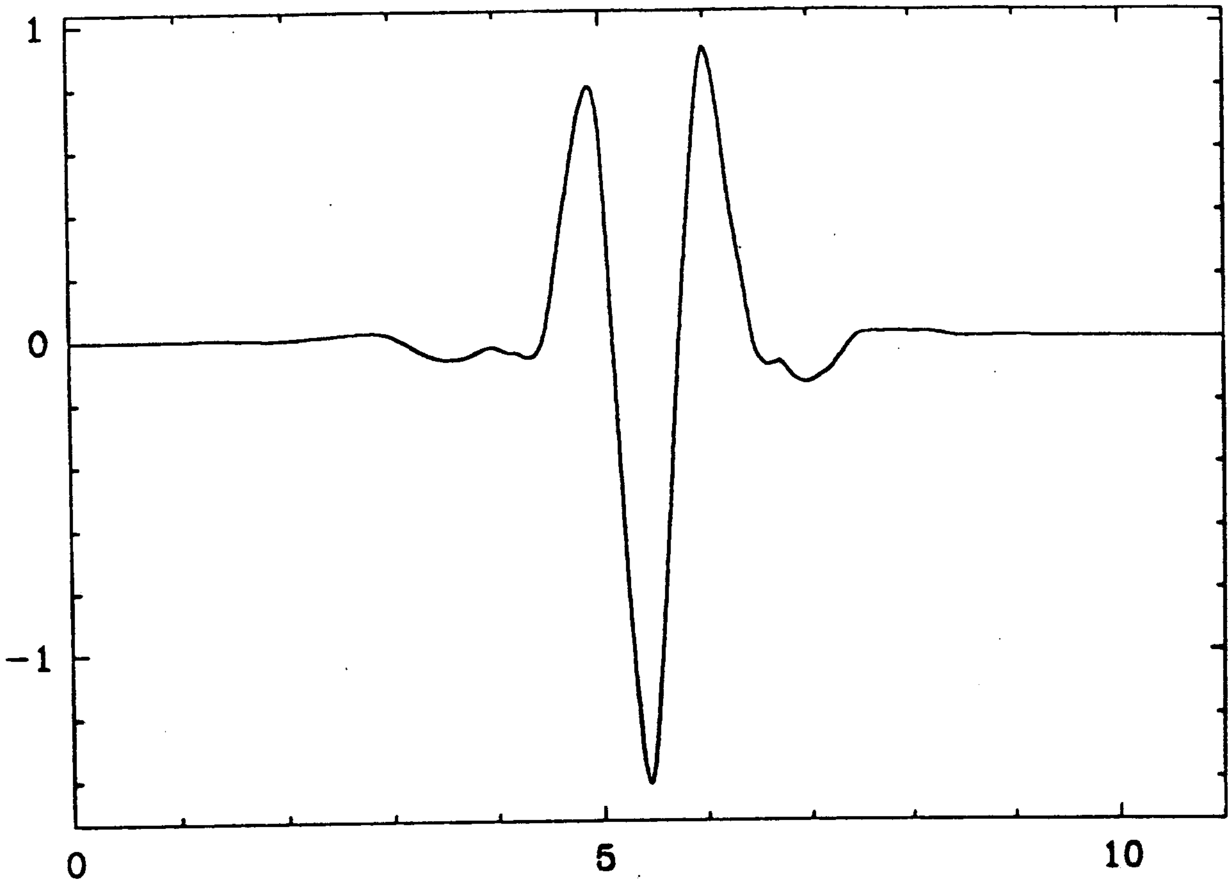


Fig. 6. Plots of iterated wavelet sequences $\{h^{11}{}_n\}$, plotted against $n2^{-j}$, for four differents choices of the scaling filter $\{g_n\}$.

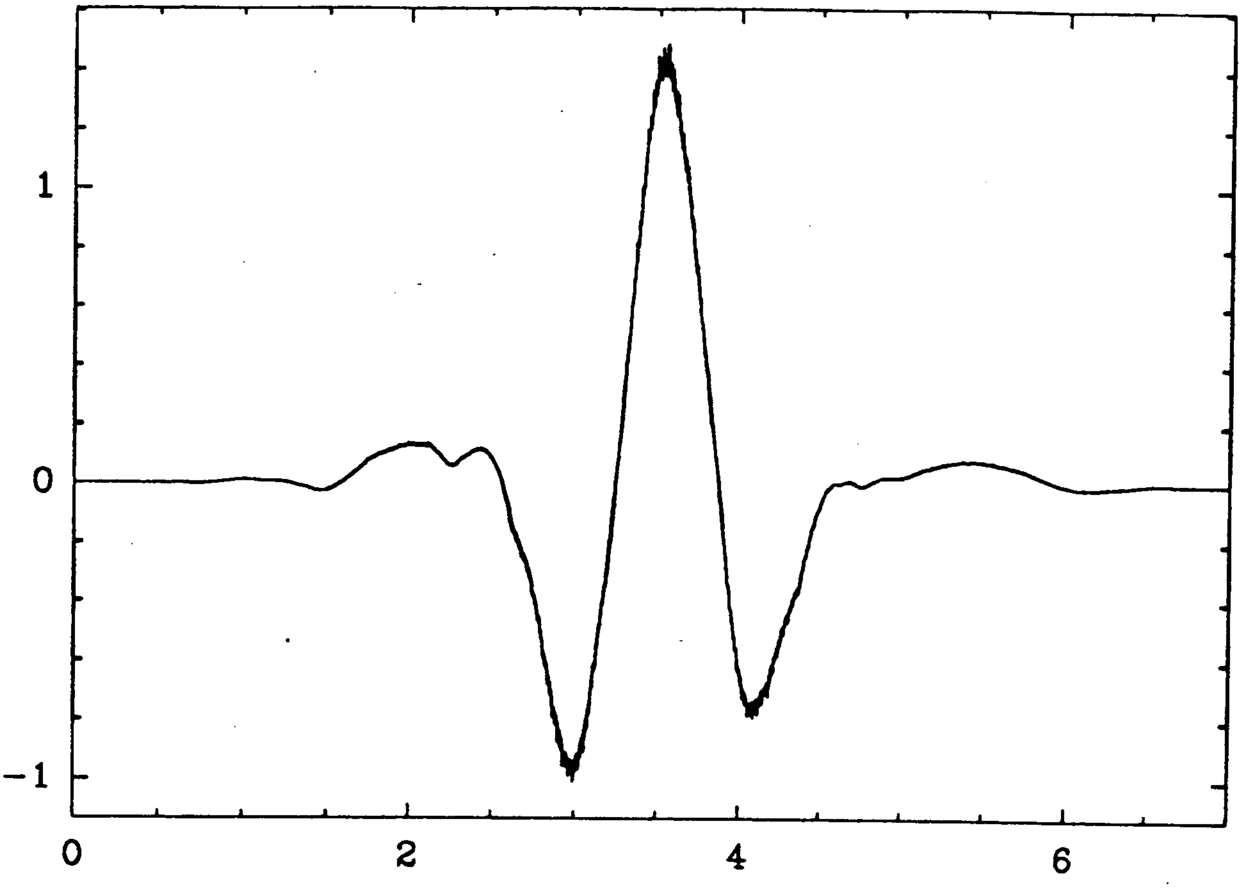
(a) $g_0 = g_1 = .7$, $g_2 = -g_3 = .1$, $g_n = 0$ otherwise. This example satisfies orthonormality (25), and can therefore be used in a perfect reconstruction paraunitary filter bank (see section VI). However, the obtained curve is a highly irregular, fractal function. In fact this curve rapidely diverges as j increases indefinitely.



(b) Daubechies wavelet of length 4: $g_0 = (1+\sqrt{3})/4\sqrt{2}$, $g_1 = (3+\sqrt{3})/4\sqrt{2}$, $g_2 = (3-\sqrt{3})/4\sqrt{2}$, $g_3 = (1-\sqrt{3})/4\sqrt{2}$. (after [11]).The regularity order is $r = 0.5500...$ The obtained curve rapidely converges, as j increases indefinitely, to a continuous, but not differentiable, function.



(c) Daubechies "closest to linear phase" wavelet of length 12 (after [12]). The regularity order, as determined in [28] is at least 2.102.



(d) This example corresponds to the 8-tap scaling filter designed by Smith and Barnwell [30]. Mathematically speaking, the obtained curve is irregular: due to small, but rapid, oscillations present in the wavelet, the curves slowly diverges as j increases indefinitely. However the obtained curves look "reasonably regular" for small j's, due to the fact that the scaling filter's transfert function is strongly attenuated in the stop band (see section VIII). This behavior cannot be predicted by the mathematical model of regularity.

ARTICLE 3

"Dyadic Up-Scaling Schemes: Simple Criteria for Regularity", par O. RIOUL

SIAM J. Math. Anal., soumis en Fev. 1991

Submitted to SIAM Journal of Mathematical Analysis, February 20, 1991.

DYADIC UP-SCALING SCHEMES: SIMPLE CRITERIA FOR REGULARITY

OLIVIER RIOUL*

Abstract. Convergence of dyadic up-scaling schemes to limit functions arises in several fields of applied mathematics and signal processing, to construct special curves and surfaces, fractals, and compactly supported wavelets. In this paper, we use a polynomial description to study existence and Hölder regularity of such limit functions. Sharp regularity estimates are derived, and optimality is proven in most cases. They can be easily implemented on a computer, and simulations show that the exact regularity order is accurately determined after a few iterations. Connection is made to regularity estimates of solutions to lattice two-scale difference equations derived by Daubechies and Lagarias, and to other known Fourier-based estimates. The former are generally optimal, while the latter are optimal only for a subclass of symmetric limit functions.

* - Centre National d'Etudes des Télécommunications, Centre Paris B, CNET/PAB/RPE/ETP, 38-40 rue du Général Leclerc, 92131 Issy-Les-Moulineaux, France.

Introduction and definitions.

This paper focuses on real-valued discrete sequences $u[n]$ ($n \in \mathbb{Z}$) of finite length, and on their behavior under repetitive action of a *dyadic up-scaling operator* G, defined as

$$\text{(1)} \qquad \mathrm{G}:\ u[n] \longrightarrow v[n] = \sum_{k \in \mathbb{Z}} u[k]\ g[n-2k].$$

The terminology "dyadic up-scaling" will be justified later. The fixed sequence $g[n]$ that parameterizes G is called the *scaling sequence*. It plays a central role in the following. Starting from the initial "impulse" sequence $\delta[n] = 1$ if $n=0$, 0 otherwise, a *dyadic up-scaling scheme* is an infinite collection of sequences $g_j[n]$ ($j \in \mathbb{N}$), defined as shown.

$$\begin{aligned} g_1[n] &= \mathrm{G}\{\ \delta[n]\ \} = g[n], \\ g_2[n] &= \mathrm{G}\{\ g_1[n]\ \}, \\ &\ \vdots \\ \text{(2)} \qquad g_{j+1}[n] &= \mathrm{G}\{\ g_j[n]\ \}, \\ &\ \vdots \end{aligned}$$

The $g_j[n]$'s are fully determined when the scaling sequence $g[n]$ is given. Of course other inital sequences can be considered.

The aim of this paper is to find the necessary and sufficient conditions on the scaling sequence $g[n]$ for the existence and *regularity* of a function of a continuous variable $g_\infty(x)$, obtained as a limit of (2) as j indefinitely increases. More precisely, we use the following definition.

$$\text{(3)} \qquad g_\infty(x) = \lim_{j \to +\infty} g_j[x_j 2^j],$$

where x_j is any sequence of dyadic rationals (of the form $n2^{-j}$, $n \in \mathbb{Z}$) converging to x as j tends to infinity. For example, x_j could be $\lfloor 2^j x \rfloor\, 2^{-j}$, where $\lfloor y \rfloor$ denotes the largest integer not exceeding y. Fig. 1 shows that the *limit function* $g_\infty(x)$ can be thought as a limit of the discrete curves $g_j[n]$, when plotted against $n2^{-j}$. By abuse of notations we say that the sequences $g_j[n]$ converge to $g_\infty(x)$ as $j \to +\infty$. The regularity is estimated using Hölder spaces $\dot{C}^r$ ($r \in \mathbb{R}$).

In this paper, we first describe (2) using the very convenient polynomial notation (section 1) and derive a basic necessary condition for (3) to hold (section 2). We then show how limit functions can be computed exactly on a computer (section 3). This leads us to define "ordinary" limit functions, to which the results of this paper fully apply (section 4). Fortunately, almost all limit functions are "ordinary."

To tackle the regularity problem, we characterize regularity properties of limit functions in terms of discrete sequences: continuity is connected to uniform convergence of the $g_j[n]$'s (section 5), Hölder regularity $\dot{C}^\alpha$ ($0<\alpha\leq 1$) is expressed by a very similar property of the $g_j[n]$'s (section 6), finite differences of the $g_j[n]$'s play the role of derivatives (section 7), and N-times continuously differentiable limit functions are therefore characterized by uniform convergence of finite differences (section 8). From these equivalences a full characterization of Hölder regularity $\dot{C}^r$ (for all $r>0$) naturally emerges in terms of discrete sequences (sections 9 and 10). The main result of this paper is an easily implementable, optimal regularity estimate derived in section 11. This estimate is then compared to related previous work [3]-[12].

The purpose of this paper is close to the one of Daubechies and Lagarias' [5]-[6] that studies the existence and regularity of solutions of *two-scale difference equations*. In fact we shall see in section 3 that the limit function $g_\infty(x)$ satisfies the following "lattice two-scale difference equation."

$$g_\infty(x) = \sum_k g[k]\, g_\infty(2x-k). \tag{4}$$

Although it can be shown [6] that solutions of (4) are not necessarily limit functions (3), both approaches are closely related. In fact, it can be shown [2] that the study of regularity of solutions of (4) can be reduced, after suitable transformation, to that of limit functions

(3). However, the contents, formulations and proofs of this paper differ notably from [5]-[6]; Daubechies and Lagarias derive conditions for the existence of L^1-solutions of (4) and estimate global and local regularity of solutions that are in fact limit functions. This paper concentrates on the determination of *optimal* estimates for global regularity of limit functions, with interpretation in terms of discrete sequences and comparison with Fourier-based techniques.

Dyadic up-scaling schemes and two-scale difference equations arise in several fields of applied mathematics. They have been used for curve fitting and to generate fractal curves or surfaces numerically [8]-[12]. They also play an important role in wavelet theory [1], [3]-[4], [13]-[15], a newly born theory in functional analysis closely realted to *filter banks* in signal processing [13], [15]. Whether limit functions $g_\infty(x)$ are regular or not may be relevant for image coding applications using wavelets [1], and this has motivated the work presented in this paper.

1. Polynomial notation.

Dyadic up-scaling schemes have been mostly described using Fourier transforms or matrices [3], [5]-[6], [8]-[10]. Throughout this paper we often use the polynomial description

$$U(X) = \sum_{n=0}^{L-1} u[n]\, X^n$$

of any causal sequence $u[n]$ of length L ($u[n] = 0$ for $n<0$ and $n>L$). Sequences of finite length can always be made causal by shifting; we therefore assume all sequences causal in the following.

In polynomial notation, the up-scaling operator (1) reads

$$U(X) \to V(X) = G(X)\; U(X^2) \tag{5}$$

This shows that it can be seen as resulting from two operations.

1) change X to X^2 in $U(X)$, i.e., insert zeros between every two samples of $u[n]$.

2) multiply by $G(X)$, i.e., convolve the result with the scaling sequence $g[n]$.

In other words the operator (1), (5) smooths $u[n]$ at twice its scale; this justifies the terminology "up-scaling." Equation (5) can therefore be seen as a discrete version of a dilation by two: $f(x) \rightarrow f(x/2)$.

Iterating (5) gives the polynomial $G_j(X)$ associated to the sequence $g_j[n]$ (2).

(6a) $$G_j(X) = G(X)\, G(X^2)\, G(X^4) \cdots G(X^{2^{j-1}}).$$

Of course, when the initial sequence in (2) is different from $\delta[n]$, say $h[n]$, (6) is simply multiplied by $H(X^{2^j})$.

(6b) $$H_j(X) = G_j(X)\, H(X^{2^j}).$$

Therefore the iterated sequence $h_j[n]$ is not $g_j[n]$ in this case, but $h_j[n] = \sum_k h[k]\, g_j[n\text{-}2^j k]$.

From definition (3), it follows that the limit function is $h_\infty(x) = \sum_k h[k]\, g_\infty(x\text{-}k)$ instead of $g_\infty(x)$ itself. Since the convergence and regularity properties of $h_\infty(x)$ and $g_\infty(x)$ are the same, we can restrict to the study of the $g_j[n]$'s and $g_\infty(x)$.

One easily finds that the length of $g_j[n]$ is $(2^j\text{-}1)(L\text{-}1)+1$ (where L is the length of $g[n]$) by estimating the polynomial degree of (6a). Therefore $g_\infty(x)$, if it exists, has compact support $[0, L\text{-}1]$. As a result, all functions considered in this paper are compactly supported.

Equation (6a) fully describes dyadic up-scaling schemes in terms of polynomials. It can be rewritten in recursive form in two ways.

(7a) $$G_{j+1}(X) = G(X)\, G_j(X^2), \text{ i.e., } g_{j+1}[n] = \sum_k g_j[k]\, g[n\text{-}2k].$$

(7b) $$G_{j+1}(X) = G_j(X)\, G(X^{2^j}), \text{ i.e., } g_{j+1}[n] = \sum_k g[k]\, g_j[n\text{-}2^j k].$$

Both are useful in the sequel. We also consider (6) for other polynomials than the fixed scaling polynomial $G(X)$: given any polynomial $U(X)$, $U_j(X)$ (with a subscript index j) is

(8) $$U_j(X) = U(X)\,U(X^2)\,U(X^4)\cdots U(X^{2^{j-1}}).$$

We shall use l^1 and l^∞ norms of discrete sequences in terms of polynomials,

$$\|U(X)\|_\infty = \max_k |u[k]|$$

$$\|U(X)\|_1 = \sum_k |u[k]|,$$

and the following well-known inequality.

(9) $$\|U(X)\ V(X)\|_\infty \leq \|V(X)\|_1 \|U(X)\|_\infty.$$

For polynomials with real coefficients, the following useful inequality holds whenever $V(X)$ has no roots *on* the unit circle.

(10) $$\|U(X)\|_\infty \leq c_V \|U(X)\ V(X)\|_\infty,$$

where c_V is a constant depending only on $V(X)$. This is trivially true for infinite sequences when the roots of $V(X)$ lie outside the unit circle; the constant c_V.is then the converging l^1-norm of the Laurent series' coefficients of $1/V(X)$, which is analytic in the complex-domain region $|X|\leq 1$. Here since $v[n]$ is a sequence of finite length L, the index reversal $n \leftrightarrow L\text{-}1\text{-}n$ in $v[n]$ transforms roots of $V(X)$ inside the unit circle into roots outside the unit circle. Hence (10) holds when $V(X)$ has no roots *on* the unit circle.

2. A necessary condition for existence of limit functions.

In order that $g_\infty(x)$ is well defined or does not vanish for all x's, the dyadic up-scaling sequences $g_j[n]$ should neither diverge nor tend to zero as $j \rightarrow +\infty$. This requires specific conditions to be fulfilled by the scaling sequence $g[n]$.

PROPOSITION 2.1. *If* $g_\infty(x)\neq 0$ *exists for some* $x \in \mathbb{R}$, *then*

(11) $$\sum_k g[2k] = \sum_k g[2k+1] \ = \ 1, \quad \text{i.e., } G(1) = 2 \text{ and } G(-1) = 0.$$

Proof. The key point is to consider the even and odd-indexed sequences $g_j[2n]$ and $g_j[2n+1]$ separately. On one hand, from definition (3), their common limit for $j \to +\infty$ is $g_\infty(2x)$. On the other hand, from (7a) we also have

$$g_j[2n] = \textstyle\sum_k g[2k]\, g_{j-1}[n-k]$$

$$g_j[2n+1] = \textstyle\sum_k g[2k+1]\, g_{j-1}[n-k].$$

Applying (3) to the right-hand sides of these equations, we easily obtain that their respective limits as $j \to +\infty$ are $(\sum_k g[2k])\, g_\infty(2x)$ and $(\sum_k g[2k+1])\, g_\infty(2x)$. By identification we have $g_\infty(2x) = (\sum_k g[2k])\, g_\infty(2x) = (\sum_k g[2k+1])\, g_\infty(2x)$, hence condition (11). ■

Condition (11) may be interpreted as follows. On one hand, $G(1)=2$ is just a normalization condition that ensures that the order of magnitude of the $g_j[n]$'s is preserved when $j \to +\infty$. On the other hand, the fact that $G(X)$ must have at least one zero at $X=-1$ is a "local" requirement. It ensures that the $g_j[n]$'s, for large j, do not rapidly oscillate in n between two different limits $(\sum_k g[2k])\, g_\infty(2x)$ and $(\sum_k g[2k+1])\, g_\infty(2x)$ (see Fig.2).

Note that (11) is not sufficient to ensure continuity of the limit. For example, the choice $G(X) = 1+X$, for which $g_j[n]=1$ for $0 \leq n < 2^j$ and 0 elsewhere, gives the non-continuous limit $g_\infty(x)=1$ for $0<x<1$, $g_\infty(x)=0$ for $x<0$ and $x>1$.

In fact, (11) does not even ensure convergence. Consider for example $G(X) = 1+X^3$: $G_j(X)$ is a polynomial in X^3, therefore $g_j[n]$ vanishes for $n=(3k+1)2^{-j}$ and $(3k+2)2^{-j}$ whereas $g_j[3k2^{-j}]=1$. It therefore cannot converge to a limit function.

3. Exact computation of limit functions.

Using (7b) and definition (3) one obtains the following two-scale difference equation [5]-[6] satisfied by $g_\infty(x)$, which is the same as (4).

$$g_\infty(x) = \sum_k g[k]\, g_\infty(2x\text{-}k).$$

This equation can be used to derive a method for computating exactly the limit function at dyadic rationals $g_\infty(n2^{-j})$, $n \in \mathbf{Z}$, with a finite number of operations.

Let $G_\infty{}^j(X)$ be the polynomial associated to the sequence $g_\infty(n2^{-j})$. Letting $x=2^{-j-1}$ in the two-scale difference equation yields $g_\infty(n2^{-j-1}) = \sum_k g[k]\, g_\infty(2^{-j}(n\text{-}2^j k))$, i.e., $G_\infty{}^{j+1}(X) = G_\infty{}^j(X)\, G(X^{2^j})$. By iteration we therefore have

(12) $$G_\infty{}^j(X) = G_\infty{}^0(X)\, G_j(X),$$

where

(13) $$G_\infty{}^0(X) = \sum_n g_\infty(n)\, X^n.$$

This means that the exact values of $g_\infty(n2^{-j})$ can be computed directly from the $g_j[n]$'s by convolving them with the sequence $g_\infty(n)$, provided that this latter can be precomputed.

There are several methods for precomputing $g_\infty(n)$, which is, by definition, the limit of $g_j[n2^j]$ as $j \to +\infty$. First note that one has, from (7b),

$$g_{j+1}[n2^{j+1}] = \sum_k g[k]\, g_j[(2n\text{-}k)2^j] = \mathbf{G}^*\{g_j[n2^j]\},$$

where $\mathbf{G}^*$ is the following "down-scaling operator," a transposed form of $\mathbf{G}$ (1) [12]:

$$\mathbf{G}^*: u[n] \longrightarrow v[n] = \sum_{k \in \mathbf{Z}} g[k]\, u[2n-k].$$

Therefore $g_\infty(n)$ can be determined as the limit of $(\mathbf{G}^*)^j\{\delta[n]\}$ when $j \to +\infty$. Another method stems from the resulting equality $g_\infty(n) = \mathbf{G}^*\{g_\infty(n)\}$. The sequence $\{g_\infty(n)\}$, $n=0, \ldots L\text{-}1$, is here determined, up to normalization, as the eigenvector of the operator $\mathbf{G}^*$ associated to the eigenvalue 1. Since we have, by proposition 2.1, $G(1)=2$ and $G(-1)=0$, it follows that $\mathbf{G}^*$ preserves the sum of the coefficients of sequences. Therefore the $g_\infty(n)$'s are normalized according to

(14) $$G_\infty{}^0(1) = \sum_n g_\infty(n) = 1.$$

4. Ordinary limit functions.

There are exceptional classes of limit functions $g_\infty(x)$ for which the regularity estimates derived in this paper will not be proven to be optimal. Optimality, as well as some other results of this paper will hold only if $g_\infty(x)$ is "ordinary," in the sense of the following definition.

DEFINITION. The limit function $g_\infty(x)$ is *ordinary* if no roots of $G_\infty^0(X)$ (13) lie on the unit circle, i.e.,

$$\sum_n g_\infty(n)\, e^{in\omega} \neq 0, \text{ for all } \omega \in \mathbb{R}.$$

This condition slightly restricts the choice of the scaling sequence $g[n]$. For example, if the scaling sequence's length is $L=4$, non-ordinary $g_\infty(x)$'s are such that $g[0]=g[3]$ and $g[1]=g[2]$. All (real-valued) limit functions are ordinary for lengths up to 3.

5. Continuous limit functions.

In this section we show that any ordinary continuous limit function $g_\infty(x)$ of a dyadic up-scaling scheme $\{g_j[n]\}$ is a *uniform* limit, i.e., that the "discrete curves" $g_j[n]$ of Fig. 1 converge "as a whole" to the limit curve $g_\infty(x)$.

THEOREM 5.1. *If $g_j[n]$ uniformly converges to the limit function $g_\infty(x)$, i.e.,*

$$\lim_{j\to+\infty} \sup_{x\in\mathbb{R}} \left| g_\infty(x) - g_j[x_j 2^j] \right| = 0 \tag{15}$$

then $g_\infty(x)$ is continous for all $x \in \mathbb{R}$. The converse is true whenever $g_\infty(x)$ is ordinary.

Proof. (⇒) Let $\varepsilon>0$ be an arbitrary small number. To prove that $g_\infty(x)$ is continuous at $x=x_0$, consider the inequality

(16) $$|g_\infty(x)-g_\infty(x_0)| \leq |g_\infty(x)-g_j[x_j2^j]|+ |g_j[x_j2^j]-g_j[x^0_j2^j]| + |g_\infty(x_0)-g_j[x^0_j2^j]|,$$

where x_j and x^0_j are dyadic rationals of the form $n2^{-j}$ converging to x and x_0, respectively. Assume for example that $x_j = \lfloor 2^j x \rfloor\, 2^{-j}$ and $x^0_j = \lfloor 2^j x_0 \rfloor\, 2^{-j}$. For j large enough, the first and third terms in the right-hand side of (16) are bounded by $\varepsilon/2$ because of (15). Now fix such a j and assume that x is close enough to x_0 so that both x and x_0 belong to the same interval $[n2^{-j}; (n+1)2^{-j}]$. Then $x_j = x^0_j$ and the second term in the right-hand side of (16) vanishes. Therefore $|g_\infty(x)-g_\infty(x_0)| \leq \varepsilon/2 + \varepsilon/2 = \varepsilon$, which shows that $g_\infty(x)$ is continuous.

(⇐) We have $\sup_x|g_\infty(x)-g_j[x_j2^j]| \leq \sup_x|g_\infty(x)-g_\infty(x_j)| + \sup_x|g_\infty(x_j)-g_j[x_j2^j]|$, where x_j is a sequence of dyadic rational $x_j=n_j2^{-j}$ converging to x. Since $g_\infty(x)$ is compactly supported and continuous, it is uniformly continuous. Therefore $\sup_x|g_\infty(x)-g_\infty(x_j)|$ tends to zero. The other term can be written $\sup_x |g_\infty(x_j)-g_j[x_j2^j]| = \|G_\infty^{\,j}(X) - G_j(X)\|_\infty$. From (12) we have $G^0_\infty(X)\,(G_\infty^{\,j}(X) - G_j(X)) = (G^0_\infty(X)-1)\, G_\infty^{\,j}(X)$. Since (14) holds, X-1 divides $G_\infty^{\,0}(X)$-1 and we can write, using (9), $\|G^0_\infty(X)\,(G_\infty^{\,j}(X) - G_j(X))\|_\infty \leq c\, \|(X-1)\, G_\infty^{\,j}(X)\|_\infty = c \sup_x |g_\infty(x_j-2^{-j})-g_\infty(x_j)|$. This tends to zero as $j\to+\infty$ because $g_\infty(x)$ is uniformly continuous. Now we can use (10) with $V(X) = G^0_\infty(X)$ because $g_\infty(x)$ is ordinary. This yields $\|G_\infty^{\,j}(X) - G_j(X)\|_\infty \to 0$ as $j\to+\infty$, which ends the proof. ■

The framework of uniform convergence is shown to be very convenient in the following. Note that uniform convergence fails for the choice $G(X)=1+X$ seen in section 2 because $\sup_x |g_\infty(x) - g_j[x_j2^j]| = 1$. This was to be expected since we have seen that the corresponding limit function is not continuous.

6. Lipschitz limit functions.

In this section we want to characterize Lipschitz limit functions $g_\infty(x) \in \dot{C}^\alpha$ $(0<\alpha\le 1)$. Recall that $g_\infty(x)$ is said to be Lipschitz of order α $(0<\alpha\le 1)$ if one has, for all x and $h \in \mathbb{R}$,

(17) $$|g_\infty(x+h) - g_\infty(x)| \le c\,|h|^\alpha,$$

where c is a constant. Here, since $g_\infty(x)$ is compactly supported, (17) only needs to be satisfied for small h's. Since the spaces $\dot{C}^\alpha$, for $0<\alpha\le 1$, interpolate between C^0 and C^1, a $\dot{C}^\alpha$-function will be said to be regular of order α. There is a slight irritation in that C^1 and $\dot{C}^1$ do not coincide; for example a linear spline function is $\dot{C}^1$ but not differentiable at its knots.

THEOREM 6.1. *If* $G(1)=2$, $G(-1)=0$, *and*

(18) $$\max_n |g_j[n+1] - g_j[n]| \le c\,2^{-j\alpha}$$

for some $0<\alpha\le 1$, *then* $g_\infty(x)$ *is* $\dot{C}^\alpha$. *The converse is true if* $g_\infty(x)$ *is ordinary. In addition, the more regular the limit, the faster the convergence to this limit:*

(19) $$\sup_x |g_\infty(x) - g_j[x_j 2^j]| \le c\,2^{-j\alpha},$$

for any sequence x_j *of dyadic rationals of the form* $n2^{-j}$ *such that* $|x-x_j| \le c\,2^{-j}$.

Proof. ($\Rightarrow$) Let us first prove equation (19). Let x_j satisfying the conditions of the theorem. One has $\sup_x |g_{j+1}[x_{j+1}2^{j+1}] - g_j[x_j 2^j]| = \max_n |g_{j+1}[2n+m] - g_j[n]|$, where $m = (x_{j+1}-x_j)2^{j+1}$ is a bounded integer. From (7a) we can write $g_{j+1}[2n+m] = \sum_k g[2k+m]\,g_j[n-k]$. Therefore the sequence $g_{j+1}[2n+m] - g_j[n]$ is a convolved version of $g_j[n]$, hence its associated polynomial can be written in the form $U(X)\,G_j(X)$. But from (11), we have $\sum_k g[2k+m] = 1$ (for all m) and therefore $U(1)=1$. Using (9) it follows that $\max_n |g_{j+1}[2n+m] - g_j[n]| = \|U(X)\,G_j(X)\|_\infty \le c\,\|(1-X)\,G_j(X)\|_\infty$, where c is a constant independant of j since m is bounded. Now from (18) the latter norm is bounded by $c' 2^{-j\alpha}$. We therefore end up with $\sup_x |g_{j+1}[x_{j+1}2^{j+1}] - g_j[x_j 2^j]| \leq c\, 2^{-j\alpha}$. Iterating this inequality we obtain, for $l>0$,

$$\sup_x |g_{j+l}[x_{j+l}2^{j+l}] - g_j[x_j 2^j]| \leq c\, 2^{-(j+l-1)\alpha} + \ldots + c\, 2^{-(j+1)\alpha} + c\, 2^{-j\alpha} \leq c''\, 2^{-j\alpha}.$$

Letting $l \to +\infty$ gives (19).

We now prove that $g_\infty(x)$ is $\dot{C}^\alpha$. Let x_j as before and $h_j 2^j = \lfloor h2^j \rfloor$. Consider the inequality $\sup_x |g_\infty(x+h) - g_\infty(x)| \leq \sup_x |g_\infty(x+h) - g_j[x_j 2^j + h_j 2^j]| + \sup_x |g_j[x_j 2^j + h_j 2^j] - g_j[x_j 2^j]| + \sup_x |g_j[x_j 2^j] - g_\infty(x)|$. By (19), the first and third terms in the right-hand side are bounded by $c\, 2^{-j\alpha}$. Now assume for example that $0 \leq h < 1$ and let j be such that $2^{-j} \leq h \leq 2^{-j+1}$. Then $\lfloor h2^j \rfloor = 1$ and $\sup_x |g_j[x_j 2^j + \lfloor h2^j \rfloor] - g_j[x_j 2^j]| = \max_n |g_j[n+1] - g_j[n]| \leq c\, 2^{-j\alpha}$. Putting all together this yields $\sup_x |g_\infty(x+h) - g_\infty(x)| \leq c'\, 2^{-j\alpha} \leq c\, |h|^\alpha$, i.e., $g_\infty(x)$ is $\dot{C}^\alpha$.

($\Leftarrow$) $G(1)=2$, $G(-1)=0$ results from proposition 2.1. Since $g_\infty(x)$ is $\dot{C}^\alpha$, we have $|g_\infty((n+1)2^{-j}) - g_\infty(n2^{-j})| \leq c2^{-j\alpha}$, i.e., $\|(1-X)G_\infty^{\,j}(X)\|_\infty = \|G_\infty^{\,0}(X)\ (1-X)G_j(X)\|_\infty \leq c2^{-j\alpha}$ (see (12)). Because $g_\infty(x)$ is ordinary, (10) applies to give $\|(1-X)\ G_j(X)\ \|_\infty \leq c'2^{-j\alpha}$, which is (18). ■

This theorem provides an intuitive interpretation of regularity of order $0 \leq \alpha < 1$ for dyadic up-scaling iteration schemes: regularity $\dot{C}^\alpha$ holds if and only if the absolute values of the "*slopes*" $(g_j[n+1] - g_j[n])/2^{-j}$ of the discrete curves $g_j[n]$'s (see next section) grow less than $2^{j(1-\alpha)}$ when j indefinitely increases. For example, if the slopes of $g_j[n]$ are always bounded for all j's, then $g_\infty(x)$ is $\dot{C}^1$. On the contrary, less regularity allows slopes to increase indefinitely and the resulting limit function, although continuous, may present a "fractal" structure as shown in Fig. 3. Note that in this case, (19) means that uniform convergence of the curves $g_j[n]$ is slower as slopes increase faster.

Since we have a characterization of regularity for ordinary $g_\infty(x)$'s, it is easy to find a condition on the scaling sequence $g[n]$ that states an *exact* regularity order $0<\alpha<1$.

COROLLARY 6.2. *Assume* $G(1)=2$, $G(-1)=0$, *and that* $g_\infty(x)$ *is ordinary. If, for* $0<\alpha<1$,

(20) $$\max_n |g_j[n+1] - g_j[n]| \textit{ decreases as } 2^{-j\alpha} \textit{ when } j \to +\infty,$$

then $g_\infty(x)$ *is* $\dot{C}^\alpha$ *but is not* $\dot{C}^{\alpha+\varepsilon}$, *for any* $\varepsilon>0$.

Proof. This is an immediate consequence of Theorem 6.1: If $g_\infty(x)$ were $\dot{C}^{\alpha+\varepsilon}$ (with $\varepsilon>0$ small enough so that $\alpha+\varepsilon<1$), we would have $\max_n |g_j[n+1] - g_j[n]| \leq c\, 2^{-j(\alpha+\varepsilon)}$, which is in contradiction with (20). ■

Corollary 6.2 does not hold if $\alpha=1$, since $\max_n |g_j[n+1] - g_j[n]|$ cannot decrease faster than 2^{-j} as $j \to +\infty$ when $g_\infty(x)$ is more regular than $\dot{C}^1$ (see section 9). Otherwise, intuitively the derivative of $g_\infty(x)$ would vanish identically, which would imply $g_\infty(x) = 0$ since $g_\infty(x)$ is compactly supported.

7. Finite differences.

We now turn to the definition of *finite differences* of the $g_j[n]$'s as a prerequisite to study derivatives of the limit function $g_\infty(x)$. The first finite difference is

(21) $$\Delta g_j[n] = (g_j[n] - g_j[n-1]) / 2^{-j}, \text{ i.e., } \Delta G_j(X) = 2^j(1-X)\, G_j(X)$$

In other words, the $\Delta g_j[n]$'s are the *slopes* of the "discrete curve" $g_j[n]$ plotted against $n2^{-j}$ (see Fig. 2). Finite differences $\Delta^k g_j[n]$ of order k are simply obtained by applying k times the difference operator Δ.

(22) $$\Delta^k G_j(X) = 2^{jk}(1-X)^k\, G_j(X)$$

In order to study finite differences $\Delta^k g_j[n]$ similarly as for the $g_j[n]$, it is convenient to express them as dyadic up-scaling sequences as well, associated to other scaling sequences than $g[n]$. The following lemma shows that this is possible when $G(X)$ has enough zeros at X=-1.

LEMMA 7.1. *Assume $G(X)$ has at least k zeros at X=-1. Define $G^k(X)$ by*

$$G(X) = \left(\tfrac{1+X}{2}\right)^k G^k(X). \tag{23}$$

Then the finite differences $\Delta^k g_j[n]$'s follow a dyadic up-scaling scheme with initial sequence's polynomial $(1\text{-}X)^k$ and scaling sequence's polynomial $G^k(X)$.

Proof. From (22), (23), we have $\Delta^k G_j(X) = 2^{jk}(1\text{-}X)^k \prod_{i=0}^{j-1}\left(\frac{1+X^{2^i}}{2}\right)^k G^k_j(X)$, where $G^k_j(X) = (G^k)_j(X)$ is defined by (8). Using the identity $(1\text{-}Y)(1+Y)=1\text{-}Y^2$ for $Y = X, X^2, \ldots$ we therefore obtain $\Delta^k G_j(X) = G^k_j(X)\,(1\text{-}X^{2^j})^k$, which from (6b) proves the lemma. ■

8. N-times continously differentiable limit functions.

Using the preceding sections we are ready to extend the results of section 5 to higher order regularity C^N (N-times continously differentiable functions).

THEOREM 8.1. *If finite differences $\Delta^k g_j[n]$ uniformly converge for k=0, ..., N, then $g_\infty(x)$ is C^N. The converse is true whenever $g_\infty(x)\neq 0$ is ordinary. In addition:*

$$G(X) \text{ has at least } N+1 \text{ zeros at } X=-1, \tag{24}$$

(25) *$\Delta^k g_j[n]$ uniformly converges to $g_\infty^{(k)}(x)$, the kth derivative of $g_\infty(x)$, for k=0, ..., N.*

Proof. ($\Rightarrow$) Let us first prove (25) by backward induction on k. We have to show that if $\Delta^{k+1}g_j[n]$ converges uniformly to some (continuous) function $f(x)$, then $\Delta^k g_j[n]$ converges uniformly to the compactly supported primitive of $f(x)$, called $F(x)$. For simplicity we assume k=0, the proof being identical for k>0.

Since $F(x)$ is compactly supported and C^1, it is uniformly continuously differentiable and therefore $\sup_x \mid \Delta g_j[x_j 2^j] - (F(x_j)-F(x_j-2^{-j}))/2^{-j} \mid$ tends to zero as $j \to +\infty$, where x_j are dyadic rationals $n2^{-j}$ converging to x. This can be written $\|2^j(1-X)(G_j(X)-F^j_\infty(X))\|_\infty \to 0$, where $F^j_\infty(X)$ is the polynomial associated to the sequence $f(n2^{-j})$. Now for any polynomial $U(X)$ we have $\|U(X)\|_\infty = \max_k |u[k]| \leq \sum_k |u[k]-u[k-1]| \leq d\ \|(1-X)U(X)\|_\infty$, where d is the degree of $U(X)$. Applying this to $U(X) = G_j(X) - F^j_\infty(X)$ of degree $(L-1)(2^j-1)$, where L is the length of $g[n]$, we obtain $\|(G_j(X)-F^j_\infty(X))\|_\infty \leq (L-1)\ \|2^j(1-X)(G_j(X)-F^j_\infty(X))\|_\infty$, which tends to zero. Therefore $\sup_x \mid g_j[x_j 2^j] - F(x_j)\mid = \|(G_j(X)-F^j_\infty(X))\|_\infty$ tends to zero. This proves that $g_j[n]$ converges uniformly to $F(x) = g_\infty(x)$, hence $\Delta g_j[n]$ converges uniformly to $F'(x) = g_\infty'(x)$, and by induction (25) holds.

It follows from (25) that the continuous uniform limit of $\Delta^N g_j[n]$ is $g_\infty^{(N)}(x)$, therefore $g_\infty(x)$ is C^N. By forward induction on the derivative order k, (24) follows as a consequence of proposition 2.1 and lemma 7.1.

($\Leftarrow$) We prove (25) from the assumption that $g_\infty(x)$ is ordinary and C^N by forward induction on k. For k=0, (25) is true by theorem 5.1. It remains to prove that this implies $\lim_{j\to+\infty} \sup_x |g_\infty^{(k)}(x) - \Delta^k g_j[x_j 2^j]| =0$ for k=1, ..., N, where x_j where x_j are dyadic rationals $n2^{-j}$ converging to x. For simplicity assume k=1.The proof is identical for larger k's when one replaces Δ by Δ^k. Define $\Delta G^j_\infty(X) = 2^j(1-X)G^j_\infty(X)$ or $\Delta g_\infty(x_j) = 2^j\ (g_\infty(x_j) - g_\infty(x_j-2^{-j}))$. We have

$$\sup_x |g_\infty'(x)-\Delta g_j[x_j 2^j]| \leq \sup_x |g_\infty'(x)-g_\infty'(x_j)| + \sup_x |g_\infty'(x_j)-\Delta g_\infty(x_j)|$$

$$+ \sup_x |\Delta g_\infty(x_j) - \Delta g_j[x_j 2^j]|, \tag{26}$$

The first term in the right-hand side of tends to zero x as $j\to+\infty$ because $g_\infty'(x)$ is continuous and compactly supported, therefore uniformly continuous. The second term also tends to zero because $g_\infty(x)$ is uniformly continuously differentiable on a compact support. Note that this implies

(27) $$\sup_x |\Delta g_\infty(x_j)-\Delta g_\infty(x_j-2^{-j})| = \|(1-X)\Delta G^j_\infty(X)\|_\infty \to 0.$$

The third term in the right-hand side of (26) can be written $\|\Delta G^j_\infty(X)-\Delta G_j(X)\|_\infty$. From (12) we have $G^\infty_0(X)\ (\Delta G^j_\infty(X)-\Delta G_j(X)) = (G^\infty_0(X)-1)\Delta G^j_\infty(X)$. Since (14) holds, X-1 divides $G^\infty_0(X)$-1 and we can write, using (9), $\|G^\infty_0(X)\ (\Delta G^j_\infty(X)-\Delta G_j(X))\|_\infty \leq c\ \|(X-1)\ \Delta G^j_\infty(X)\|_\infty$ which tends to zero by (27). Now we can use (10) with $V(X) = G^\infty_0(X)$ because $g_\infty(x)$ is ordinary. This yields $\|\Delta G^j_\infty(X)-\Delta G_j(X)\|_\infty \to 0$ as $j \to+\infty$, which ends the proof. ■

This theorem is useful because it allows to estimate regularity of the derivatives of an ordinary limit function $g_\infty(x)$ the same way as for $g_\infty(x)$ itself: since $G(X)$ has at least $(N+1)$ zeros at X=-1, the finite differences of the $g_j[n]$'s, which converge to the derivatives of $g_\infty(x)$, all follow dyadic up-scaling schemes.

Theorem also provides an upper bound for regularity. Since it is necessary that $G(X)$ has N+1 zeros at X=-1 to obtain C^N ordinary limit functions $g_\infty(x)$, the regularity order of $g_\infty(x)$ is always bounded by the number of zeros at X=-1 in $G(X)$. We shall see that this upper bound may be attained. However, it is important to note that imposing zeros at X=-1 in $G(X)$ does *not* ensure any regularity in general. It does not even ensure convergence, as in the example $G(X) = (1+X^3)^{N+1}$. Here there is N+1 zeros at X=-1, but $g_j[n]$ does not converge for the same reason as for the choice $G(X) = 1+X^3$ treated in section 2.

9. Checking a given Hölder regularity order.

Recall the definition of Hölder regularity. The limit function $g_\infty(x)$ is regular of order $r=N+\alpha$ $(0<\alpha\le 1)$, $g_\infty(x) \in \dot{C}^r$, if it is C^N and its Nth derivative $g_\infty^{(N)}(x)$ is Lipschitz of order α, $g_\infty^{(N)}(x) \in \dot{C}^\alpha$, as defined earlier by (17). Hölder spaces $\dot{C}^r$ well interpolate the spaces C^N of N-times continuously differentiable functions. As already mentionned for $N=1$, $\dot{C}^N$ contains functions that are not C^N, such as spline functions of degree N. In fact "$g_\infty(x)$ is $\dot{C}^N$" can be thought of "$g_\infty(x)$ is almost C^N," since if $g_\infty(x)$ is $\dot{C}^{N+\varepsilon}$, for some $\varepsilon>0$, then $g_\infty(x)$ is truly C^N. Other spaces, based on the Fourier transform of $g_\infty(x)$, are sometimes used to define a regularity order $r\in\mathbb{R}$ as well. They will be considered later in section 17.

Using the results of the preceding sections, we can extend the characterization of Lipschitz limit functions $\dot{C}^\alpha$ $(0<\alpha\le 1)$, derived in section 6, to any Hölder regularity order r.

THEOREM 9.1. *If* $G(1)=2$, $G(X)$ *has at least* $N+1$ *zeros at* $X=-1$ *and*

$$\max_n |\Delta^N g_j[n+1] - \Delta^N g_j[n]| \le c\, 2^{-j\alpha}, \tag{28}$$

for some $\alpha>0$, *then* $g_\infty(x)$ *is* $\dot{C}^{N+\alpha}$. *The converse is true whenever* $g_\infty(x)\neq 0$ *is ordinary. Moreover,* (28) *implies* $\alpha\le 1$ (*if* $g_\infty(x)\neq 0$). *In addition,*

$$\max_n |\Delta^N g_j[n+1] - \Delta^N g_j[n]| = \|(1-X)\Delta^N G_j(X)\|_\infty \tag{29}$$

can be replaced in (28) *by any of the following.*

$$\max_n |g^N_j[n+1] - g^N_j[n]| = \|(1-X)G^N_j(X)\|_\infty \tag{30}$$

$$\max_n |f^N_j[n]| = \|F^N_j(X)\|_\infty \tag{31}$$

$$\max_{0\le m\le 2^j-1} \sum_n |f^N_j[2^j n+m]|, \tag{32}$$

where we have noted $G(X) = (\frac{1+X}{2})^N G^N(X) = (\frac{1+X}{2})^N (1+X) F^N(X)$, *and where the iterated polynomials and sequences* $G^N_j(X)$, $g^N_j[n]$, $F^N_j(X)$, $f^N_j[n]$ *are defined by* (8).

Proof. ($\Rightarrow$) Assume for the moment that $\alpha\leq 1$. Since (28) implies, by theorem 6.1, that $\Delta^N g_j[n]$ converges uniformly to a $\dot{C}^\alpha$ function, it follows from the proof of theorem 8.1 that all finite differences $\Delta^k g_j[n]$ converge uniformly to $g_\infty^{(k)}(x)$, for k=0, .., N. Hence $g_\infty(x)$ is $\dot{C}^{N+\alpha}$.

($\Leftarrow$) If $g_\infty(x)$ is ordinary and C^N, then by theorem 8.1, $\Delta^N g_j[n]$ converges uniformly to $g_\infty^{(N)}\in\dot{C}^\alpha$. Therefore by theorem 6.1, (28) holds.

It remains to be proved that (29)-(32) are "equivalent" in the following sense. Two sequences u_j and v_j are equivalent if there are two constants c_1 and c_2, independent of j, such that $c_1 v_j \leq u_j \leq c_2 v_j$. From lemma 7.1, we have $\Delta^N G_j(X) = (1-X^{2^j})^N G^N{}_j(X)$. Using (9) we therefore have $\|(1\text{-}X)\Delta^N G_j(X)\|_\infty \leq 2^N \,\|(1\text{-}X)G^N{}_j(X)\|_\infty$. Now, since the degree of $(1\text{-}X)G^N{}_j(X)$ is less than 2^jL, where L is the length of the sequence $g^N[n]$, we also have

$$\|(1\text{-}X)G^N{}_j(X)\|_\infty = \|(1-X^{2^jL})^N(1\text{-}X)G^N{}_j(X)\|_\infty = \left\|\left(\frac{1-X^{2^jL}}{1-X^{2^j}}\right)^N(1\text{-}X)\Delta^N G_j(X)\right\|_\infty$$
$$\leq c_N\|(1\text{-}X)\Delta^N G_j(X)\|_\infty.$$

This proves that (29) and (30) are equivalent. The proof of (30) $\Leftrightarrow$ (31) is very similar, based on the relation $(1\text{-}X)\; G^N{}_j(X) = (1-X^{2^j})F^N{}_j(X)$, which comes from lemma 7.1. The equivalence (31) $\Leftrightarrow$ (32) is obvious.

We now prove that (28) implies $\alpha\leq 1$. Since $G(1)=2$, we have $F^N(1)=F^N{}_j(1)=1$. Therefore $\|F^N{}_j(X)\|_\infty \geq 2^{-j}\,\|F^N{}_j(X)\|_1 \geq 2^{-j}\,|F^N{}_j(1)| = 2^{-j}$, which shows, from (28) written with (31), that $\alpha\leq 1$. ■

All the above equivalent sequences (29)-(32) will be useful in the following. As in section 6, the following corollary immediately results from theorem 9.1.

COROLLARY 9.2. *Assume* $G(1)$=2 *and* $G(X)$ *has at least* N+1 *zeros at* X=-1. *If, for* $0<\alpha\leq 1$,

(33) $$\max_n |\Delta^N g_j[n+1] - \Delta^N g_j[n]| \approx 2^{-j\alpha} \text{ as } j\to+\infty,$$

then $g_\infty(x)$ is $\dot{C}^{N+\alpha}$, but is not $\dot{C}^{N+\alpha+\varepsilon}$, for any $\varepsilon>0$.

This does not hold for $\alpha=1$, since by theorem 9.1, (28) implies $\alpha\leq 1$. Of course in (33) one can choose either (29), (30), (31) or (32).

Note that the characterization (28), or the criterion (33) depends on the choice of N. Theorem 9.1 (or corollary 9.2) therefore allows to check whether the exact regularity order r (that is, the number such that $g_\infty(x)$ is $\dot{C}^r$ but not $\dot{C}^{r+\varepsilon}$, for any $\varepsilon>0$) falls in the range $N \leq r < N+1$.

Assume for example that (33) is tested for some $N=N_0$ larger than the *unknown* exact regularity order r. This test necessarily fails, and that only ensures that $g_\infty(x)$ is not $\dot{C}^{N_0}$. On the other hand, if the value of N is too small, i.e. $N=N_1 < r-1$, then necessarily (33) is satisfied with $\alpha=1$. This shows that $g_\infty(x)$ is $\dot{C}^{N_1+1}$, but does not tell whether $g_\infty(x)$ is actually more regular or not. In both cases (under or over-estimated N's), the criterion (33) has to be checked all over again for other values of N to determine r. It is only when it turns out that $N<r\leq N+1$ that the criterion is really optimal and provides $N+\alpha=r$.

Therefore, the exact regularity order cannot be determined in general unless all possible values of N are tried. This problem is addressed and solved in the next section.

10. Determining the exact Hölder regularity order.

In the preceding section we have seen that (28) or (33), for a given N truly determines the exact regularity order of the limit function $g_\infty(x)$ only if this regularity order does not exceed N by one. This value of N is *a priori* unknown. However, if (28) can be extended to *negative* values of α, then the exact regularity order r is determined even if N is "too large," i.e., $N+1 \geq r$. That is, even if the criterion (28) for regularity $r>N$ fails, it could be used to characterize lower regularity $0< r \leq N$. In particular, if one uses all zeros at X=-1 in $G(X)$, (i.e., if $G(X)$ has no more than N+1 such zeros), then the characterization (28), extended to any $\alpha\leq 1$, necessarily provides the exact regularity order r. This extension is provided by the following theorem.

THEOREM 10.1. *Theorem* 9.1 *and corollary* 9.2 *hold for* $-N < \alpha \leq 1$, *with the following slight restriction.*

If (28) *holds for* $\alpha=-n$, n=0, 1, ..., N-1, *then* $g_\infty(x)$ *is only "almost"* $\dot{C}^{N-n}$, *i.e., its* (N-n-1)*th derivative satisfies*

(34) $$|g_\infty^{(N-n-1)}(x+h) - g_\infty^{(N-n-1)}(x)| \leq C.|h|\ |\log|h||\ \textit{for all } x, h\in \mathbb{R}.$$

This theorem will be proven if we can simultaneously increase α and decrease N by 1 in (28). We therefore need the following lemma.

LEMMA 10.2. *Assume* $G(1)$ =2, *and* $G(X)$ *has at least* N+1 *zeros at* X=-1. *The condition*

(35) $$\max_n |\Delta^{N-1} g_j[n+1] - \Delta^{N-1} g_j[n]| \leq c\ 2^{-j(\alpha+1)},$$

implies (28). *The converse implication holds for* $\alpha<0$ *only. When* $\alpha=0$, (28) *implies*

(36) $$\max_n |\Delta^{N-1} g_j[n+1] - \Delta^{N-1} g_j[n]| \leq c\, j\ 2^{-j}.$$

Proof of lemma. From theorem 9.1 we know that both (28) and (35) can be written under several forms, using one of the sequences (29)-(32). Using (31), condition (35) is $\|F^{N-1}{}_j(X)\|_\infty \leq c2^{-j(\alpha+1)}$. But since $F^{N-1}(X)=G^N(X)/2$, we have $F^{N-1}{}_j(X)=2^{-j}G^N{}_j(X)$, hence (35) can be written $\|G^N{}_j(X)\|_\infty \leq c2^{-j\alpha}$. Now (28), using the equivalent form (30), is $\|(1-X)G^N{}_j(X)\|_\infty \leq c2^{-j\alpha}$. Since using (9), we have $\|(1-X)G^N{}_j(X)\|_\infty \leq 2\|G^N{}_j(X)\|_\infty$, (35) implies (28).

Assume $\alpha \leq 0$. To prove the converse implication, rewrite (28) and (35) using (31), knowing that $F^{N-1}{}_j(X) = 2^{-j}\left(\frac{1-X^{2^j}}{1-X}\right)F^N{}_j(X)$ by lemma 7.1. We therefore have to prove that (28): $\|F^N{}_j(X)\|_\infty \leq c2^{-j\alpha}$ implies (35): $\|\left(\frac{1-X^{2^j}}{1-X}\right)F^N{}_j(X)\|_\infty \leq c2^{-j\alpha}$. There is a problem at $X = 1$; we therefore subtract $F^N{}_j(1) = F^N(1) = 1$ to $F^N{}_j(X)$ as shown.

$$\|\left(\frac{1-X^{2^j}}{1-X}\right)F^N{}_j(X)\|_\infty \leq \|\left(\frac{1-X^{2^j}}{1-X}\right)\,(F^N{}_j(X)-1)\,\|_\infty + \|\left(\frac{1-X^{2^j}}{1-X}\right)\|_\infty$$

The second term in the right-hand side equals 1. Denote the first one by $\|H_j(X)\|_\infty$. From (7a), we have $F^N{}_j(X)-1 = (F^N{}_{j-1}(X^2)-1) + (F^N(X)-1)F^N{}_{j-1}(X^2)$, where $(X-1)$ divides $(F^N(X)-1)$ since $F^N(1)=1$. Therefore $H_j(X) = H_{j-1}(X^2)\,(1+X) + \left(\frac{F^N(X)-1}{X-1}\right)(X^{2^j}-1)F^N{}_{j-1}(X^2)$ and $\|H_j(X)\|_\infty \leq \|H_{j-1}(X)\|_\infty + c2^{-(j-1)\alpha}$. By induction on j, for $\alpha<0$, $\|H_j(X)\|_\infty \leq c'2^{-j\alpha}$ follows, which implies (35). When $\alpha=0$, we have $\|H_j(X)\|_\infty \leq c'j$, which implies (36). ■

Proof of theorem 10.1. If α is not a negative integer, the generalization of theorem 9.1 to $-N<\alpha\leq 0$ follows from several applications of lemma 10.2. When $\alpha=-n$, $n=0, \ldots, N-1$, by n successive applications of lemma 10.2, (28) implies $\max_n |\Delta^{N-n}g_j[n+1] - \Delta^{N-n}g_j[n]| \leq c$. Applying lemma 10.2 again we only obtain $\max_n |\Delta^{N-n-1}g_j[n+1] - \Delta^{N-n-1}g_j[n]| \leq c\, j2^{-j}$. By theorem 9.1 this only implies that $g_\infty(x)$ is $\dot{C}^{N-n-\varepsilon}$ (for any $\varepsilon>0$), but we have a little more: mimicing the proof of the direct part of theorem 6.1, we have $\sup_x |g_\infty{}^{(N-n-1)}(x+h) - g_\infty{}^{(N-n-1)}(x)| \leq c\, j2^{-j}$ for $2^{-j} \leq h \leq 2^{-j+1}$, which is (34). ■

11. A practical, optimal Hölder regularity estimate.

Theorem 10.1 already provides an optimal regularity criterion (28) (with $-N<\alpha\leq 1$). However, it is not implementable on a computer as written since it needs to be verified for all j's and since the order of magnitude of the constant c in (28) is unknown. The aim of this section is to transform this criterion into an easily implementable estimate for Hölder regularity, computable with a finite number of operations.

The following theorem assumes some properties and notations we have already met:

- $G(1)=2$,
- $G(X)$ has at least $N+1$ zeros at $X=-1$,
- $F^N(X)$ (corresponding to the sequence $f^N[n]$) is, as defined in theorem 9.1., $G(X)$ without its $N+1$zeros at $X=-1$, i.e.,

$$G(X) = (\tfrac{1+X}{2})^N(1+X)F^N(X)$$

It generates iterated polynomials $F^N_j(X)$ and sequences $f^N_j[n]$ by (8).

THEOREM 11.1. *With the above notations and assumptions, define the Hölder regularity estimate $N+\alpha_j^N$ by*

$$2^{-j\alpha_j^N} = \max_{0\le m\le 2^j-1} \sum_n |f^N_j[2^jn+m]| \tag{37}$$

and let $\alpha^N = \sup_j \alpha_j^N \le 1$.

If there exists j such that $N+\alpha_j^N > 0$, then $g_-(x)$ is $\dot{C}^{N+\alpha_j^N}$ (almost $\dot{C}^{N+\alpha_j^N}$ if $\alpha_j^N \in -\mathbb{N}$, see (34)). Therefore $g_-(\mathrm{x})$ is $\dot{C}^{N+\alpha^N-\varepsilon}$ for any $\varepsilon>0$.

In addition, if $g_-(x)$ is ordinary then α_j^N tends to α^N as $j\to+\infty$, with

$$0\le \alpha^N-\alpha_j^N \le c\,/\,j, \tag{38}$$

and the regularity estimate is optimal: If $\alpha^N \neq 1$, or if $\alpha^N = 1$ and $G(X)$ has no more than $N+1$ zeros at $X=-1$, then $g_-(\mathrm{x})$ is $\dot{C}^{N+\alpha^N-\varepsilon}$ but is not $\dot{C}^{N+\alpha^N+\varepsilon}$, for any $\varepsilon>0$.

Proof. From (37), the condition (28), rewritten with (32) is satisfied only when $\alpha\le\liminf_{j\to+\infty}\alpha_j^N\le\alpha^N$. From theorem 9.1 we also have $\alpha_j^N\le1$ for all j, hence $\alpha^N\le1$. Now if $g_-(x)$ were ordinary and $\dot{C}^{N+\alpha^N+\varepsilon}$, where $\alpha^N<1$ and $\varepsilon>0$, (28) would hold with $\alpha=\alpha^N+\varepsilon$, which contradicts $\alpha\le\alpha^N$. Therefore if $g_-(x)$ is ordinary, $\alpha^N<1$ implies that $g_-(x)$ is not

$\dot{C}^{N+\alpha^N+\varepsilon}$ for any $\varepsilon>0$. In addition, $g_\infty(x)$ cannot be $\dot{C}^{N+1+\varepsilon}$ if $G(X)$ has no more than $N+1$ zeros at $X=-1$ because of theorem 8.1.

We now prove that theorem 10.1 holds when $\alpha=\alpha_j^N$, for any fixed $j\geq 0$ such that $\alpha_j^N > -N$. Let $G^N(X)=(1+X)F^N(X)$ and $G^N_j(X)$ as in theorem 9.1. By lemma 7.1, the sequence $\delta_l[n] = g_l^N[n+1]-g_l^N[n]$ follows a dyadic up-scaling scheme with scaling sequence $f^N[n]$. Therefore by iterating (7b) for $F^N(X)$ we have $\Delta_{l+j}(X) = F_j^N(X)\,\Delta_l(X^{2^j})$, i.e., $\delta_{l+j}[n]=\sum_k \delta_l[k]\,f^N_j[n-2^jk]$. Now write $n=2^jp+m$ with $0\leq m\leq 2^j-1$. We obtain 2^j convolutions indexed by m, $\delta_{l+j}[2^jp+m]=\sum_k \delta_l[k]\,f^N_j[2^j(p-k)+m]$. Therefore $|\delta_{l+j}[2^jp+m]| \leq (\sum_n |f^N_j[2^jn+m]|\,)\max_n|\delta_l[n]|$ for all m. This gives $\max_n|\delta_{l+j}[n]| \leq 2^{-j\alpha_j^N}\max_n|\delta_l[n]|$, which by iteration implies, for $l=l_1\,j+l_2$, $\max_n|\delta_l[n]| = \|(1-X)G^N_l(X)\|_\infty \leq c_j\,2^{-l\alpha_j^N}$, where c_j depends only on j. This equation is exactly (28), written with (30), where $\alpha=\alpha^N_j$. Therefore theorem 10.1 applies for any $\alpha=\alpha^N_j$ such that $\alpha_j^N > -N$. The limit function is thus $\dot{C}^{N+\alpha_j^N}$ (with the restriction (34)), and therefore g_∞(x) is $\dot{C}^{N+\alpha^N-\varepsilon}$ for any $\varepsilon>0$. Uniform convergence of finite differences to derivatives is a consequence of theorem 8.1.

We finally prove (38). When $g_\infty(x)$ is ordinary and $\dot{C}^{N+\alpha^N-\varepsilon}$, by theorem 10.1, (28), written with (32), holds for $\alpha=\alpha^N-\varepsilon$. By definition of α^N_j (37), we thus have $2^{-j\alpha_j^N} \leq c\,2^{-j(\alpha^N-\varepsilon)}$ for any $\varepsilon>0$, which implies (38). ■

Let us precise the practical outcomes of theorem 11.1. For a given number of iterations j, and a given N, the computation of $N+\alpha^N_j$ with a finite number of operations by (37) always gives a Hölder regularity estimate for $g_\infty(x)$. Of course the estimate is likely to be improved when the number of iterations increases. Note that from theorem 8.1, finite differences $\Delta^k g_j[n]$ converge uniformly to the derivatives of $g_\infty(x)$ whenever these derivatives exist.

Fig. 3 shows that N must be chosen large enough because the estimate $N+\alpha^N_j$ is bounded by $N+1$, whereas the exact regularity order of $g_-(x)$ might be greater than $N+1$. It is therefore recommanded that N whould be chosen maximal (i.e., such that $G(X)$ has exactly $N+1$ zeros at $X=-1$). In this case theorem 11.1 ensures that the regularity estimates $N+\alpha^N_j$ are close to the exact regularity order of $g_-(x)$. Besides, it was numerically found that the accuracy of the regularity estimates $N+\alpha^N_j$ generally increases with N.

In most cases, $g_-(x)$ is ordinary and the regularity estimates $N+\alpha^N_j$ (where N is large enough) tend to exact regularity order of $g_-(x)$ as $j\to+\infty$. Note that if N is not large enough, $N+\alpha^N_j$ necessarily converges to $N+1$ (see Fig. 4). The rate of convergence of the estimate to the exact regularity order is at least proportional to $1/j$ by (38). In practice, Fig. 4 shows that this convergence rate is fairly high. When the scaling sequence length L is not too large (e.g. $L\leq 10$), the exact regularity order is numerically determined with precision 0.1 after a few dozens of iterations j.

Theorem 11.1 is the main result of this paper. It permits to (sharply) estimate Hölder regularity in most cases of interest. The remainder of this paper is devoted various connections of this result to related work on Hölder regularity estimates, and illustrates results with examples.

12. Connection with Daubechies and Lagarias' estimates.

In [6], Daubechies and Lagarias have determined sharp conditions for Hölder regularity based on matrix products. Although the approach in [6] relies on two-scale difference equations (4) rather than on limit functions (3), the above results of this paper, which were derived independently, are closely related to what can be found in [6]. In fact, (29) reads, in matrix notation:

(39) $$2^{-j\alpha_j^N} = \max_{\varepsilon_i = 0 \text{ or } 1} \left\| \prod_{i=1}^{j} \mathbf{F}_{\varepsilon_i}{}^N \right\|_1,$$

where the matrices $\mathbf{F}_0{}^N$ and $\mathbf{F}_1{}^N$ of size $(L\text{-}1)\times(L\text{-}1)$ (where L is the length of the sequence $f^N[n]$) are defined as shown.

(40) $$\mathbf{F}_0{}^N = \begin{bmatrix} f^N[0] & 0 & 0 & 0 & \cdots \\ f^N[2] & f^N[1] & f^N[0] & 0 & \cdots \\ f^N[4] & f^N[3] & f^N[2] & f^N[1] & \cdots \\ f^N[6] & f^N[5] & f^N[4] & f^N[3] & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

(41) $$\mathbf{F}_1{}^N = \begin{bmatrix} f^N[1] & f^N[0] & 0 & 0 & \cdots \\ f^N[3] & f^N[2] & f^N[1] & f^N[0] & \cdots \\ f^N[5] & f^N[4] & f^N[3] & f^N[2] & \cdots \\ f^N[7] & f^N[6] & f^N[5] & f^N[4] & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

and where $\|.\|_1$ denotes the l^1-norm of a square matrix $\mathrm{A} = (a_{i,j})$, i.e., $\|\mathrm{A}\|_1 = \max_i \sum_j |a_{i,j}|$.

To prove (30), one may notice, using some polynomial algebra, that $f^N_j[2^j n+m]$, seen as a vector indexed by n, is equal to the first column of $\prod_{i=1}^{j} \mathbf{F}_{\varepsilon_i}{}^N$, where $m = \varepsilon_1\varepsilon_2 \ldots \varepsilon_j$ in base 2. It can be shown that the l^1-norm of the this column dominates the others and is therefore equal to the l^1-norm of the whole matrix.

Using (39) one may easily recover the results concerning global Hölder regularity in [6] from the results of this paper. The formulation (39) and that used in [6] differ only by some minor differences: Daubechies and Lagarias use l^2-norms rather than l^1-norms, and the matrices they consider are a bit larger than (40)-(41), because they correspond to $G(X) = \left(\frac{1+X}{2}\right)^N (1+X)\, F^N(X)$ rather than to $F^N(X)$. Although regularity estimates are not proven optimal in general in [6], Daubechies and Lagarias prove optimality for several examples, such as those of section 14.

Working with matrices is useful when one wants to find optimal estimates of regularity "by hand," [6] without implementing (37). Unfortunately, it seems difficult to derive a general method for determining the optimal regularity by matrix manipulation. As a result, unlike an implementation of (37) on a computer, each example has to be investigated separately and requires fastidious treatment. We here recall for completeness the basic method used in [6].

Daubechies and Lagarias' Method [6].

1. *Write out the matrices* $\mathbf{F}_0^N$ *and* $\mathbf{F}_1^N$ *and compute their eigenvalues.*

2. *Diagonalize the matrix that has the largest module of eigenvalue (spectral radius)* ρ. *Assume for example that this matrix is* $\mathbf{F}_0^N$. *Define the matrix* $\mathbf{B}$, *whose columns are proportional to the eigenvectors of* $\mathbf{F}_0^N$ *(that is,* $\mathbf{B}^{-1}\,\mathbf{F}_0^N\,\mathbf{B}$ *is diagonal and its* l^1*-norm is* ρ*).* $\mathbf{B}$ *is parameterized by* L-1 *numbers, one for each column.*

3. *Compute* $\mathbf{B}^{-1}\,\mathbf{F}_1^N\,\mathbf{B}$. *If one can find a parameterization of* $\mathbf{B}$ *such that*

$$\|\,\mathbf{B}^{-1}\,\mathbf{F}_1^N\,\mathbf{B}\,\| \leq \rho, \tag{42}$$

(where $\|.\|$ *is any matrix norm) then the optimal Hölder regularity (for ordinary limit functions* $g_-(x)$*) is* N - $\log_2 \rho$.

Proof. First we have $2^{-j\alpha_j^N} = \max\limits_{\varepsilon_i=0\text{ or }1} \left\| \prod_{i=1}^{j} \mathbf{F}_{\varepsilon_i}{}^N \right\|_1 \geq \|(\mathbf{F}_0{}^N)^j\|$ by specifying ε_i=0 for all i. Let λ be an eigenvalue of $\mathbf{F}_0^N$ and v an associated nonzero eigenvector. We have, on one hand, $\|(\mathbf{F}_0{}^N)^j.v\| \leq \|(\mathbf{F}_0{}^N)^j\|.\|v\|$, and on the other hand, $\|(\mathbf{F}_0{}^N)^j.v\| = |\lambda|^j\,\|v\|$. It follows that $\rho^j = \sup|\lambda|^i \leq \|(\mathbf{F}_0{}^N)^j\| \leq 2^{-j\alpha_j^N}$. Secondly, with the change of basis $\mathbf{B}$ we have $2^{-j\alpha_j^N}$ =

$\max_{\varepsilon_i=0\text{ or }1} \left\| \mathbf{B}\left(\prod_{i=1}^{j} \mathbf{B}^{-1}\mathbf{F}_{\varepsilon_i}{}^{N}\mathbf{B}\right)\mathbf{B}^{-1} \right\|_1 \leq c \max_{\varepsilon_i=0\text{ or }1} \prod_{i=1}^{j} \left\|\mathbf{B}^{-1}\mathbf{F}_{\varepsilon_i}{}^{N}\mathbf{B}\right\|_1$. But one has $\|\mathbf{B}^{-1}\mathbf{F}_0{}^{N}\mathbf{B}\| = \rho$ and (42), therefore $2^{-j\alpha_j^N} \leq c\, \rho^j$ follows. We therefore have proven $\rho^j \leq 2^{-j\alpha_j^N} \leq c\, \rho^j$, which implies that $\alpha^N = \lim_j \alpha_j^N = -\log_2\rho$. ■

Note that this method is only optimal if (42) is met for at least one matrix norm, and it is not clear whether this holds for a large class of scaling sequences $g[n]$ [6].

13. A sharp upper bound for regularity.

So far we have seen two types of Hölder regularity estimates: One is optimal in (almost) all cases (section 11), but many iterations (performed on a computer) are required to accurately determine the regularity order. The other (section 12) requires only the calculation of two spectral radii of matrices, but is likely to be sometimes suboptimal. Based on the latter, it is nevertheless possible to obtain a (possibly sharp) upper bound for regularity which only requires the computation of one spectral radius. This upper bound is derived as follows.

Using the notations of the preceding section we have $2^{-j\alpha_j^N} = \max_{\varepsilon_i=0\text{ or }1} \left\| \prod_{i=1}^{j} \mathbf{F}_{\varepsilon_i}{}^{N} \right\|_1 \geq$ $\max(\|(\mathbf{F}_0{}^N)^j\|, \|(\mathbf{F}_1{}^N)^j\|)$. As seen in section 12, this is greater than $\max(\rho(\mathbf{F}_0{}^N), \rho(\mathbf{F}_1{}^N))^j$, where $\rho(\mathbf{F}_0{}^N)$ ($\rho(\mathbf{F}_1{}^N)$) is the spectral radius of $\mathbf{F}_0{}^N$ ($\mathbf{F}_1{}^N$). Therefore an upper bound for the Hölder regularity of the limit funcion $g_{\infty}(x)$ is $N\text{-}\log_2 \max(\rho(\mathbf{F}_0{}^N), \rho(\mathbf{F}_1{}^N))$. In fact (42) ensures that this upper bound is attained for ordinary limit functions.

The computation of this upper bound can be simplified to the search of the spectral radius of only one matrix $\mathbf{F}^N$, defined as the common sub-matrix of $\mathbf{F}_0{}^N$ and $\mathbf{F}_1{}^N$.

$$\mathbf{F}^N{}_0 = \left[\begin{array}{c|ccc} f^N[0] & 0 & \cdots & 0 \\ \hline f^N[2] & & & \\ f^N[4] & & \mathbf{F}^N & \\ \vdots & & & \end{array}\right] \quad \text{and} \quad \mathbf{F}^N{}_1 = \left[\begin{array}{ccc|c} & & & \vdots \\ & \mathbf{F}^N & & f^N[L-3] \\ & & & f^N[L-2] \\ \hline 0 & \cdots & 0 & f^N[L-1] \end{array}\right].$$

We have max $(\rho(\mathbf{F}_0{}^N), \rho(\mathbf{F}_1{}^N)) = \max(|f^N[0]|, |f^N[L\text{-}1]|, \rho(\mathbf{F}^N))$. Therefore

(43) The regularity order of $g_\infty(x)$ is at most $N\text{-}\log_2 \max(|f^N[0]|, |f^N[L\text{-}1]|, \rho(\mathbf{F}^N))$.

This result, used in conjonction with theorem 11.1, can be used to compute sharp lower and upper bounds of Hölder regularity of $g_\infty(x)$. Table I provides these bounds for the examples presented in the next section.

14. Examples: Daubechies' orthonormal wavelets.

Daubechies has derived orthonormal bases of compactly supported wavelets using dyadic up-scaling schemes [3]. The "basic wavelet" is defined as the limit function $h_\infty(x)$ of the scheme (2) or (6), with scaling sequence $g[n]$ of length L, and with initial sequence $h[n]=(-1)^n g[L\text{-}1\text{-}n]$. She shows [3] that regular basis functions $2^{-j/2}h_\infty(2^{-j}x\text{-}k)$, defined for all integers j and k, form an *orthonormal* basis of $\mathrm{L}^2(\mathbb{R})$ if L is even and

(44) $$G(X)\,\tilde{G}(X) - G(-X)\,\tilde{G}(-X) = 4\,X^{L-1},$$

where $\tilde{G}(X)$ is the polynomial associated to the sequence $g[L\text{-}1\text{-}n]$. In [3], $G(X)$ is moreover required to have as many zeros at X=-1 as possible. This results in several possible solutions for $G(X)$ that have exactly $L/2$ zeros at X=-1 [3]-[4]. Examples of G(X) in [3] have all zeros outside the unit circle ("minimum phase" choice in the signal processing terminology, since X corresponds to a delay). In [6], the optimal regularities of "minimum phase" wavelets $h_\infty(x)$ for L=4, 6, 8 are obtained using the method described in the preceding section. It turns out that (42) holds for these lengths: the upper bound (43) is attained and actually equals $|f^N[0]|$, where N=$L/2$-1. The estimated regularity of Daubechies "minimum phase" wavelets derived in [4] is therefore -$\log_2 |g[0]|$ for lengths L=4, 6, 8. This estimate is optimal as can be checked directly from theorem 10.1 by noting that the first "slope" of $\Delta^N g_j[n]$ is $|2g[0]|^j=2^{j(1-\alpha)}$ where $\alpha=-\log_2 |g[0]|$. Table I lists these optimal regularities and upper bounds (43), as well as the corresponding program outputs implementing (37).

There are other solutions $g[n]$, derived for $L\geq 8$ in [4], which, unlike "minimum phase solutions" are close to being symmetric. Table I shows that the regularity estimates for these wavelets, based on theorem 11.1, are lower than those of "minimum phase" wavelets.

15. "Strictly linear phase" symmetric limit functions.

In this section we apply the above results to a subclass of scaling sequences which is often encountered. This section is also a prerequiste to study the differences between Hölder regularity estimates and those determined using Fourier techniques (section 17). The subclass considered here consists of scaling sequences for which either $G(X)$ or $G(X)/(1+X)$ is "strictly linear phase," in the following sense.

DEFINITION. A polynomial $U(X)$ (or its associated sequence $u[n]$ of finite length L) is *strictly linear phase* if it is symmetric: $u[n] = u[L\text{-}1\text{-}n]$ and if $U(e^{i\omega})\ e^{i\frac{L-1}{2}\omega}$ does not change sign for any $\omega \in \mathbb{R}$.

Symmetry of $u[n]$ implies $U(e^{i\omega})\ e^{i\frac{L-1}{2}\omega} \in \mathbb{R}$. This condition is called "linear phase" in signal processing [5]. The above definition requires more, namely that no discontinuities of the phase due to a change of sign in $U(e^{i\omega})\ e^{i\frac{L-1}{2}\omega}$ occur. Therefore, complex zeros of the symmetric polynomial $U(X)$ occur in pairs $(z, 1/z^*)$ not only for $|z|\neq 1$, but also *on* the unit circle. That is, roots on the unit circle have even order. It follows that $U(X)$ has an even number of roots, hence L is odd.

If either $G(X)$ or $G(X)/(1+X)$ is strictly linear phase, then for N odd (even, respectively), the sequence $f^N[n]$ in (37) is strictly linear phase. The following theorem shows that in this case the determination of the *exact* regularity order of the limit function $g_\infty(x)$ only requires the computation of the spectral radius of one matrix. This is to be compared with sections 12 and 13, where it is shown that two matrices are involved in the general case, and that the computation of one matrix's spectral radius only provides an upper bound for regularity.

The following regularity estimate has been derived independently, by other means, and on particular examples of strictly linear phase scaling sequences, in [3] and [8] (see sections 16 and 17).

THEOREM 15.1. *Assume* $G(1)=2$, $G(X)$ *has at least* $N+1$ *zeros at* $X=-1$: $G(X) = (\frac{1+X}{2})^N(1+X)F^N(X)$, *and* $F^N(X)$ *is strictly linear phase. Define the* $\frac{L-1}{2}\times\frac{L-1}{2}$ *matrix* $\hat{\mathbf{F}}^N$ *by folding the following* $\frac{L-1}{2}\times(L-2)$ *matrix:*

$$\begin{bmatrix} \cdots & \hat f^N[3] & \hat f^N[2] & \hat f^N[1] & \hat f^N[0] & \hat f^N[1] & \hat f^N[2] & \hat f^N[3] & \cdots \\ \cdots & \hat f^N[1] & \hat f^N[0] & \hat f^N[1] & \hat f^N[2] & \hat f^N[3] & \hat f^N[4] & \hat f^N[5] & \cdots \\ \cdots & \hat f^N[1] & \hat f^N[2] & \hat f^N[3] & \hat f^N[4] & \hat f^N[5] & \hat f^N[6] & \hat f^N[7] & \cdots \\ & & & & \vdots & & & & \end{bmatrix}$$

(where $\hat f^N[n] = f^N[\frac{L-1}{2}\pm n]$ *) around its middle column:*

$$\hat{\mathbf{F}}^N = \begin{bmatrix} \hat f^N[0] & 2\hat f^N[1] & 2\hat f^N[2] & \cdots \\ \hat f^N[2] & \hat f^N[1]+\hat f^N[3] & \hat f^N[0]+\hat f^N[4] & \cdots \\ \hat f^N[4] & \hat f^N[3]+\hat f^N[5] & \hat f^N[2]+\hat f^N[6] & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{45}$$

Let ρ *be its spectral radius. One has* $\rho\geq 1/2$. *If* $\rho < 2^N$, *then the limit function* $g_\infty(x)$ *is* $\dot{C}^{N-\log_2\rho}$ *(almost* $\dot{C}^{N-\log_2\rho}$ *in the sense* (34) *if* $\alpha\in -\mathbb{N}$). *In addition, if* $g_\infty(x)$ *is ordinary,* *and if either* $\rho>1/2$ *or* $\rho=1/2$ *and* $G(X)$ *has no more than* $N+1$ *zeros at* $X=-1$, *then* $g_\infty(x)$ *is not* $\dot{C}^{N-\log_2\rho+\varepsilon}$ *for any* $\varepsilon>0$.

Proof. Define $\hat{f}^N{}_j[n]=f^N{}_j[2^{j-1}(L-1)+n]$. This non-causal, symmetric sequence is strictly linear phase. We first prove that $\|F^N{}_j(X)\|_\infty=\max_n|\hat{f}^N{}_j[n]|=|\hat{f}^N{}_j[0]|$. Using Fourier coefficients, we have $\hat{f}^N{}_j[n]=\frac{1}{2\pi}\int_0^{2\pi}\hat{F}_j^N(e^{i\omega})e^{-in\omega}d\omega$ where $\hat{F}_j^N(e^{i\omega})=\sum_n\hat{f}_j^N[n]e^{in\omega}=\left|\hat{F}_j^N(e^{i\omega})\right|$. Therefore $\max_n|\hat{f}^N{}_j[n]|\leq\frac{1}{2\pi}\int_0^{2\pi}|\hat{F}_j^N(e^{i\omega})|d\omega=|\hat{f}^N{}_j[0]|$.

The theorem therefore results from theorem 10.1 if we prove that $|\hat{f}_j^N[0]|$ is equivalent to ρ^j, as $j\to+\infty$. From (7b) written for $F^N(X)$, we have, for $0\leq m\leq 2^j-1$, $f^N{}_{j+1}[2^{j+1}n+m+2^j]=\sum_k f^N[k+1]\ f^N{}_j[2^j(2n-k)+m]$. This means, in matrix notation, $(f^N{}_{j+1}[2^{j+1}n+m+2^j])_n=\mathbf{F}^N{}_1\ (f^N{}_j[2^jn+m])_n$, where $\mathbf{F}^N{}_1$ is defined by (41). Let $m=2^j-\frac{L-1}{2}$ (for sufficiently large j's to ensure $m\geq0$.) The above equation is then rewritten, in terms of the $\hat{f}_j^N[n]$, as $(\hat{f}^N{}_{j+1}[2^{j+1}(n-\frac{L-3}{2})])_n=\mathbf{F}^N{}_1\ (\hat{f}^N{}_j[2^j(n-\frac{L-3}{2})])_n$. By symmetry this equation can be restricted to $n=0, ..., \frac{L-3}{2}$, in which case the action of $\mathbf{F}^N{}_1$ is exactly that of $\hat{\mathbf{F}}^N$. It follows by induction on j that $|\hat{f}_j^N[0]|$ is equivalent to $\|(\hat{\mathbf{F}}^N)^j\|_\infty$, hence to ρ^j, when $j\to+\infty$. ■

16. Examples: Deslauriers and Dubuc's interpolation schemes.

Deslauriers and Dubuc have studied regularity of limit functions of particular dyadic up-scaling schemes [9]-[10]. In such schemes, the iterated sequence is left unchanged at each iteration, and one simply inserts between the coefficients $g_j[n]=g_{j+1}[2n]$ and $g_j[n+1]=g_{j+1}[2n+2]$ the value $g_{j+1}[2n+1]$ of the Lagrangian polynomial interpolation of the K consecutive values $g_j[n+1-K/2]$, ..., $g_j[n]$, $g_j[n+1]$, ..., $g_j[n+K/2]$, where K is even. This reduces to a dyadic up-scaling scheme (2), where the scaling sequence $g[n]$, of length $L=2K-1$, when made *causal* by shifting, is defined by

$$g[2n] = \delta[n] = 1 \text{ if } n{=}0, 0 \text{ elsewhere.}$$

(46) $$g[2n{+}1] = L_n(\tfrac{K-1}{2}),$$

where $L_n(X)$ is the Lagrangian polynomial $L_n(X) = \prod_{k \neq n} \left(\frac{X-k}{n-k} \right)$ associated to the interpolation points $k{=}0, ..., K{-}1$.

We have to show that theorem 15.1 applies in this case. It turns out [14] that $G(X)$ of length $L{=}2K{-}1$ is exactly $G_W(X)\ \tilde{G}_W(X)$, where $G_W(X)$ is the scaling polynomial of Daubechies' wavelets of length K (see section 14. - this fact will be useful in section 17). From section 14 it follows that $G(X)$ has exactly K zeros at $X{=}{-}1$. Moreover it is strictly linear phase because $G(e^{i\omega}) = G_W(e^{i\omega})\ \tilde{G}_W(e^{i\omega}) = |G_W(e^{i\omega})|^2 \geq 0$. Thus theorem 15.1 applies with $N{=}K{-}1$.

In addition, since $g_{j+1}[2n] = g_j[n]$, we have $g_\infty(n) = \lim_{j \to +\infty} g_j[2^j n] = g[2n] = \delta[n]$, therefore the limit function $g_\infty(x)$ is ordinary: theorem 15.1 gives the *exact* regularity order of $g_\infty(x)$.

The computation of the matrices $\hat{\mathbf{F}}^{K-1}$ (45) needed by theorem 15.1 can be easily done using the formula

$$f^{K-1}[n] = c \binom{K-2}{n}^{-1} \sum_{i=0}^{n} (-1)^i \binom{K-1}{i}^2, \ n{=}0, ..., K{-}2,$$

which results from (46) after some calculation. Determination of their spectral radii yields to the optimal regularities of Table II.

In [8], Deslauriers and Dubuc extended the study of the previous scaling sequence for $L{=}7$ (i.e., $K{=}4$) to the following scaling sequence (here defined for $n{=}{-}3, ..., 3$).

$$g[0]{=}1,\ g[\pm 1]{=}1/2{-}a,\ g[\pm 3]{=}a, \text{ and 0 elsewhere,}$$

where $a \in \mathbb{R}$. The case $a{=}{-}1/16$ correspond to the previous example, which from Table 2 yields regularity $\dot{C}^2$.

The simplicity and usefulness of theorem 15.1 is well illustrated through this example. Note that since $g[2n]=\delta[n]$, the limit function is ordinary as above for all $a\in\mathbf{R}$. This scaling sequence is easily seen to be strictly linear phase for $-1/16\leq a\leq 1/2$, therefore theorem 15.1 applies in this case (for other values of a one has to use more general theorems such as theorem 11.1). Here (for $a\neq -1/16$) $G(X)$ has exactly two zeros at $X=-1$, and we can therefore apply theorem 15.1 with $N=1$. We have $\hat{f}^1[0]=1+4a$, $\hat{f}^1[\pm 1]=4a$, and $\hat{f}^1[\pm 2]=2a$, hence

$$\hat{\mathbf{F}}^1=\begin{pmatrix}1+4a & -8a\\ 2a & -4a\end{pmatrix}$$

Its spectral radius is $\rho=\frac{1}{2}\left(1+\sqrt{1+16a}\right)$. From theorem 15.1, the exact regularity order of $g_\infty(x)$ is $r = 2 - \log_2\,(1+\sqrt{1+16a})$, which decreases from 2 to 0 when a increases from -1/16 to 1/2 (see Fig. 5).

17. Comparison with Fourier-based regularity estimates.

This paper has developped a direct approach based on the definition of Hölder regularity to estimate regularity of limit functions $g_\infty(x)$ of dyadic up-scaling schemes. But various other approaches for estimating regularity were also published. Some of them estimate the decay of the Fourier Transform $\hat{g}_\infty(\omega)$ of the (compactly supported) limit function $g_\infty(x)$ when $|\omega|\to+\infty$. One has easy access to $\hat{g}_\infty(\omega)$ from the scaling sequence $g[n]$ by [3]

$$(47)\qquad \hat{g}_\infty(\omega) = \lim_{j\to+\infty} G_j(e^{i\omega}).$$

Here other spaces than the $\dot{C}^r$ are used to interpolate the spaces C^N of N-times continuously differentiable functions. One generally considers one of the following spaces: $H_1^{\,r}$, $H_2^{\,r}$, $H_\infty^{\,r}$, defined by the conditions $|\omega|^r\,\hat{g}_\infty(\omega)\in$ $\mathbf{L}^1$, $\mathbf{L}^2$, $\mathbf{L}^\infty$, respectively. Estimations of parameters r for these spaces guarantees some Hölder regularity, since we have, for any $\varepsilon>0$,

$$(48) \qquad H_\infty^{r+1+2\varepsilon} \subset H_2^{r+1/2+\varepsilon} \subset H_1^{r} \subset \dot{C}^{r}.$$

Most of these inclusions are easily proven. The second one uses the Cauchy-Schwarz inequality and [9] contains a proof of the last one.

In [3], Daubechies has derived an estimate for $g_\infty(x) \in \dot{C}^{r-\varepsilon}$ based on $H_\infty^{r+1+\varepsilon}$. This estimate is easily deduced from the above results of this paper. We have, using the notations of theorem 9.1,

$$\|F^N_j(X)\|_\infty = \max_n |f^N_j[n]| \leq \frac{1}{2\pi}\int_0^{2\pi} |F^N(e^{i\omega})| d\omega \leq \max_{\omega\in\mathbf{R}} |F^N(e^{i\omega})|.$$

Define the number β_j such that $2^{-j\beta_j} = \max_{\omega\in\mathbf{R}} |F^N(e^{i\omega})|$. Then by theorem 10.1 and 11.1, $g_\infty(x)$ is $\dot{C}^{N+\beta-\varepsilon}$, where $\beta = \limsup_{j\to+\infty} \beta_j$. In fact, A.Cohen has shown [7] that this method gives the best number β such that $g_\infty(x) \in H_\infty^{N+\beta+1}$. This estimate is therefore optimal "in the Fourier sense."

But is this estimate optimal for Hölder regularity? The following theorem shows that the answer is *no*. The basic reason for this is that the exact Hölder regularity order of $g_\infty(x)$ depends on the *phase* of $\hat{g}_\infty(\omega)$, i.e., on the phase of $G(e^{i\omega})$ by (47), whereas Fourier-based regularity estimates only depend on the *module* of $\hat{g}_\infty(\omega)$. This theorem also shows that in the framework of section 15 (the "strictly linear phase" case), optimal Fourier-based estimates are also optimal for Hölder regularity. Note that the strictly linear phase case corresponds to limit functions that can be made *zero-phase* by shifting, i.e., $\hat{g}_\infty(\omega) \geq 0$.

THEOREM 17.1. *Fourier-based regularity estimates, i.e., based on either H_1^r, H_2^r, or H_∞^r, are not optimal for Hölder regularity in general. They are nonetheless off by less than* 1 *compared to optimal Hölder regularity estimates.*

For strictly linear phase sequences, based regularity estimates based on H_1^r are optimal for Hölder regularity.

Proof. We first prove optimality in the strictly linear phase case. From (47), the framework of section 15 can easily be reduced to the case $\hat{g}_\infty(\omega)\geq 0$. Optimality for spaces H_1^r and $\dot{C}^r$ coincide if we prove that in this case $g_\infty(x) \in \dot{C}^\alpha$ implies $g_\infty(x) \in H_1^{\alpha-\varepsilon}$, for any $\varepsilon>0$. We may restrict to $0<\alpha\leq 1$, otherwise just consider a derivative of $g_\infty(x)$. The integral $I(\omega) = \int_{\mathbb{R}} \sin(\omega h/2)\ |h|^{-1-\alpha+\varepsilon}\ dh$ absolutely converges for $0<\alpha\leq 1$; making a change of variable yields $I(\omega) = |\omega|^{\alpha-\varepsilon}\ I(1)$. Therefore $\int \hat{g}_\infty(\omega)\ |\omega|^{\alpha-\varepsilon}\ d\omega = c \iint \hat{g}_\infty(\omega)\ \sin(\omega h/2)\ |h|^{-1-\alpha+\varepsilon}\ dh\ d\omega$ $= c \int (g_\infty(h/2)-g_\infty(-h/2))\ |h|^{-1-\alpha+\varepsilon}\ dh$ absolutely converges because $g_\infty(x)$ is compactly supported and $\dot{C}^\alpha$. This proves $g_\infty(x) \in H_1^{\alpha-\varepsilon}$.

When $g[n]$ is not strictly linear phase, however, Fourier-based regularity estimates are not optimal. This is the case for Daubechies' wavelets seen in section 14. In fact, Table I shows that the regularity estimates of "more symmetric" Daubechies' wavelets are numerically found to be less than those of "minimum phase" Daubechies' wavelets. These two families differ only in the phase of $G(e^{j\omega})$, therefore by (47) Hölder regularity depends on the phase of $\hat{g}_\infty(\omega)$ .Let us explain this point precisely.

As previously mentionned in section 16, the scaling polynomial $G(X)$ of a Daubechies' wavelet is such that $G(X)\tilde{G}(X)$ is the scaling polynomial of a Deslauriers and Dubuc scheme. We have $G(e^{j\omega})\tilde{G}(e^{j\omega}) = |G(e^{j\omega})|^2$, thefore from (47) the Fourier Transform of the limit function of a Deslauriers and Dubuc scheme is $|\hat{g}_\infty(\omega)|^2$, where $\hat{g}_\infty(\omega)$ is the Fourier Transform of the limit function of the corresponding wavelet. Since the Deslauriers and Dubuc limit functions are strictly linear phase, their optimal Hölder regularity estimates r, listed in Table II, are optimal for H_1^r. This implies $g_\infty(x) \in H_2^{r/2}$, therefore $g_\infty(x) \in \dot{C}^{\frac{r-1}{2}-\varepsilon}$. Table I shows that these regularities $(r-1)/2$, first derived in [3], are not optimal for Hölder regularity although they are for $H_2^{r/2}$ spaces.

It remains to be proved that optimal Fourier-based estimates are greater than r-1, where r is the exact Hölder regularity estimate. This results from $\dot{C}^r \subset H^r_\infty$, which is certainly known in the literature, although I was unable to find a reference about it. A proof is as follows. If $g_-(x)$ is $\dot{C}^\alpha$ for $0<\alpha\leq 1$ (for $\alpha>1$ on just has to consider derivatives of $g_-(x)$), then $|\hat{g}_-(\omega)(e^{i\xi h}-1)| \leq \int |g_-(x+h)-g_-(x)|\,dx \leq 2Lc|h|^\alpha$ where L is the length of $g_-(x)$. Specifying $h=1/\omega$ for $\omega\neq 0$ yields $g_-(x) \in H_-^\alpha$. ■

Acknowledgements. The author would like to thank P. Duhamel and T. Blu (CNET, Paris) for fruitful discussions and for carefully reading the manuscript, and I. Daubechies (AT&T Bell Labs, Murray Hill) for letting me use her not yet published material [4]-[6] and for suggesting me the result of A. Cohen mentioned in section 17.

REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, *Image Coding Using Vector Quantization in the Wavelet Transform Domain*, in Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc., Albuquerque, NM, Apr. 3-6, 1990, pp.2297-2300.

[2] T. Blu, personal communication.

[3] I.Daubechies, *Orthonormal Bases of Compactly Supported Wavelets*, Comm. in Pure and Applied Math., Vol.41, No.7 (1988), pp.909-996.

[4] I. Daubechies, *Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme*, to appear.

[5] I.Daubechies and J.C.Lagarias, *Two-Scale Difference Equations I.Existence and Global Regularity of Solutions*, to appear in SIAM J.Math. Anal..

[6] I.Daubechies and J.C.Lagarias, *Two-Scale Difference Equations II.Local Regularity, Infinite Products of Matrices and Fractals*, to appear in SIAM J.Math. Anal.

[7] I. Daubechies, personal communication.

[8] G.Deslauriers and S.Dubuc, *Interpolation Dyadique et Fractals*, in *Dimensions non entières et applications*, G.Cherbit ed., Masson, Paris, (1987), pp.44-56 .

[9] G. Deslauriers and S. Dubuc, *Symmetric Iterative Interpolation Processes*, Constructive Approximation, Vol. 5 (1989), pp. 49-68.

[10] S.Dubuc, *Interpolation Through an Iterative Scheme*, J. Math. Anal. Appl., Vol.114, (1986), pp.185-204.

[11] N. Dyn, D.Levin, and J.A. Gregory, *A 4-point Interpolatory Subdivision Scheme for Curve Design*, Computer Aided Geometric Design, Vol.4, (1987), pp.257-268.

[12] C.A. Micchelli and H. Prautzsch, *Refinement and Subdivision for Spaces of Integer Translates of a Compactly Supported Function*, in *Numerical Analysis*, D.F.Griffith and G.A.Watson eds., Academic Press, New York, (1987), pp.192-222.

[13] O. Rioul, *A Unifying Discrete-Time Multiresolution Theory that unifies Octave-Band Filter Banks, Wavelet and Pyramid Transforms*, submitted to IEEE Trans. Signal Proc., (June 1990).

[14] M.J. Shensa, *Affine Wavelets: Wedding the Atrous and Mallat Algorithms*, IEEE Trans. Signal Proc., to appear.

[15] M.Vetterli and C.Herley, *Wavelets and Filter Banks: Relationships and New Results*, in Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc., Albuquerque, NM, Apr. 3-6, 1990, pp. 1723-1726.

FIGURE LEGENDS

Fig. 1. A dyadic up-scaling scheme converging to a limit function (after [3]). The discrete sequences $g_j[n]$ are plotted as pulses, against $n2^{-j}$ for j=1, 2, 3, and 6. At each iteration step the up-scaling operator (1) is applied, which approximatively doubles the number of coefficients while preserving a global shape. When $j \to +\infty$, these discrete curves converge to a "nice-looking," regular limit function, compactly supported on [0, 13].

Fig. 2. Two examples of diverging dyadic up-scaling schemes. Figures (a), (c) show six plots of the discrete sequences $g_j[n]$ (j=1, ..., 6), represented as coefficients joined by segments and plotted against $n2^{-j}$. Figures (b), (d) show the obtained curve after 9 iterations.

(a), (b) $g[0]=g[1]=g[2]=2/3$, $g[n]=0$ elsewhere. Here $G(-1)=2/3 \neq 0$. Note that up-scaling follows a fractal law.

(c), (d) $g[0]=g[4]=0.5$, $g[1]=g[3]=0.99$, $g[2]=1$, $g[n]=0$ elsewhere, renormalized such that $G(1)=2$. Here $G(-1) \cong 0.01$ is so small that divergence is not obvious at the level of the figure. Divergence is here due to very small oscillations that occur in the graph of $g_j[n]$. Although very small, these oscillations are so rapid that they preclude convergence.

Fig. 3. Two examples of converging dyadic up-scaling schemes (after [3]). The $g_j[n]$'s are plotted against $n2^{-j}$ for j=1, ..., 6, with coefficients joined by segments, so that the behavior of the "slopes" can be observed.

(a) The limit function is $\dot{C}^{0.5500...}$ and not C^1. Therefore slopes are allowed to increase indefinitely near the peaks of the limit function.

(b) The limit function is $\dot{C}^{1.0878...}$, therefore C^1. Slopes are constrained to be bounded, especially near the apparent "peaks" of the limit function.

Fig. 4. Program output of regularity estimates $N+\alpha^N_j$ (37) with N=0, 1, 2, for j=1 to 20 iterations. The corresponding limit function is the Daubechies "minimum phase" wavelet of length 5 (see section 14), whose exact regularity order is r=1.0878.... For N=0, the estimate is bounded by 1 and therefore does not converge to r. For N=2 the estimate converges fairly rapidly to r. After 20 iterations one finds $2+\alpha^2_{20} = 1.0831$....

Fig. 5. Plots of Deslauriers and Dubuc limit functions corresponding to $g[0]=1$, $g[\pm1]=0.5-a$, $g[\pm3]=a$, $g[n]=0$ elsewhere. The successive values of a are a=-1/16 (regularity order 2), a=0 (regularity order 1), a=1/4 (regularity order $\log_2(\sqrt{5}-1)$=0.305...) and a=0.4 (regularity order 0.104...).

TABLE I

REGULARITY ESTIMATES OF DAUBECHIES'

ORTHONORMAL COMPACTLY SUPPORTED WAVELETS [3]-[4]

(for both types of wavelets)		Minimum Phase Daubechies' Wavelets			More symmetric Wavelets
Scaling sequence length L	Best Fourier-based estimates [3] - section 7	Program output (20 iterations) eq. (37)	Optimum regularity [6]	Upper Bound $-\log_2\lvert g[0]\rvert$ section 14	Program output (20 iterations) eq. (37)
4	0.4999	0.5500	0.5500	0.5500	-
6	0.9150	1.0831	1.0878	1.0878	-
8	1.1275	1.6066	1.6179	1.6179	1.3961
10	1.596	1.942	-	2.143	1.762
12	1.888	2.163	-	2.665	2.102

TABLE II.

REGULARITY ESTIMATES

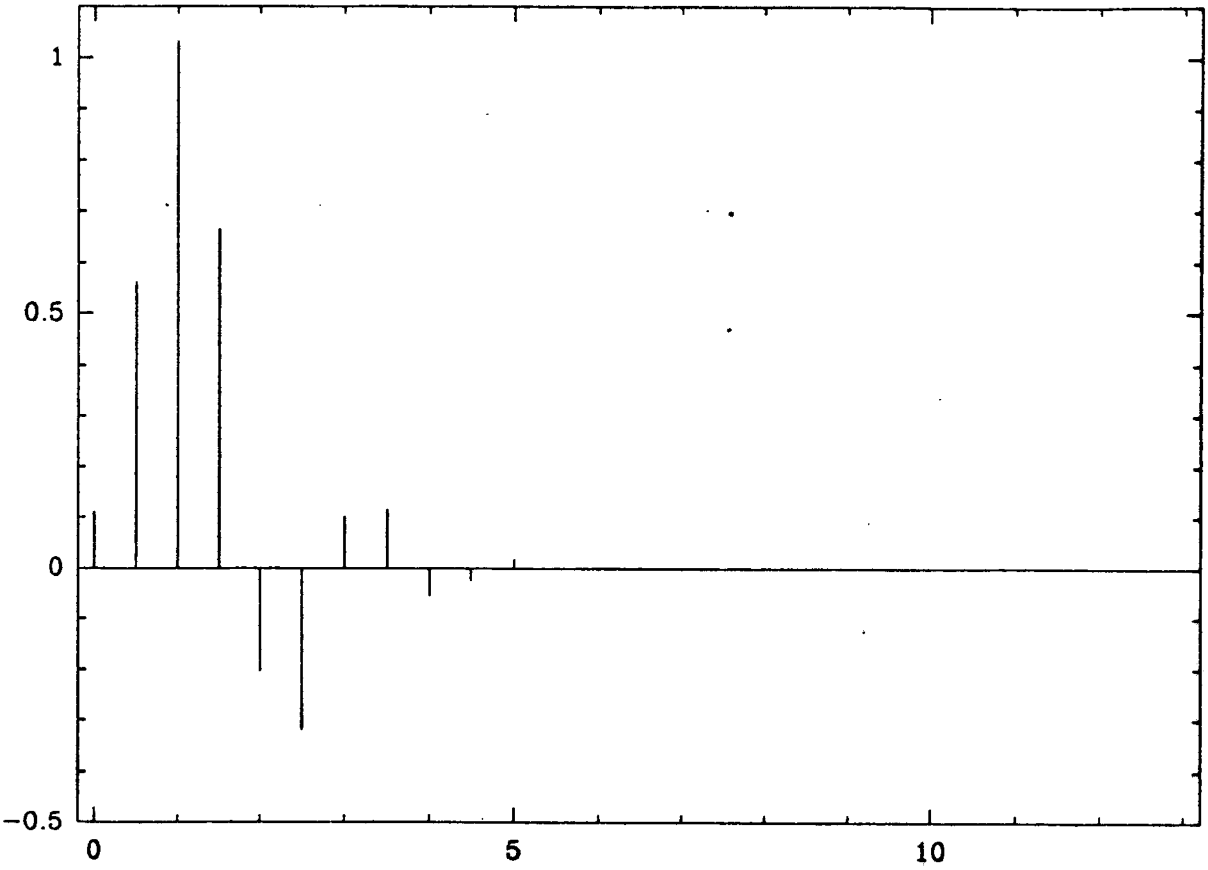
FOR DYADIC LAGRANGIAN INTERPOLATION

SCHEMES OF DESLAURIERS AND DUBUC [8]-[10]

Interpolation order K	Length L	Optimal Hölder Regularity
2	3	1.
4	7	2.
6	11	2.83
8	15	3.25
10	19	4.19
12	23	4.77

Fig. 1.

j=1



j=2

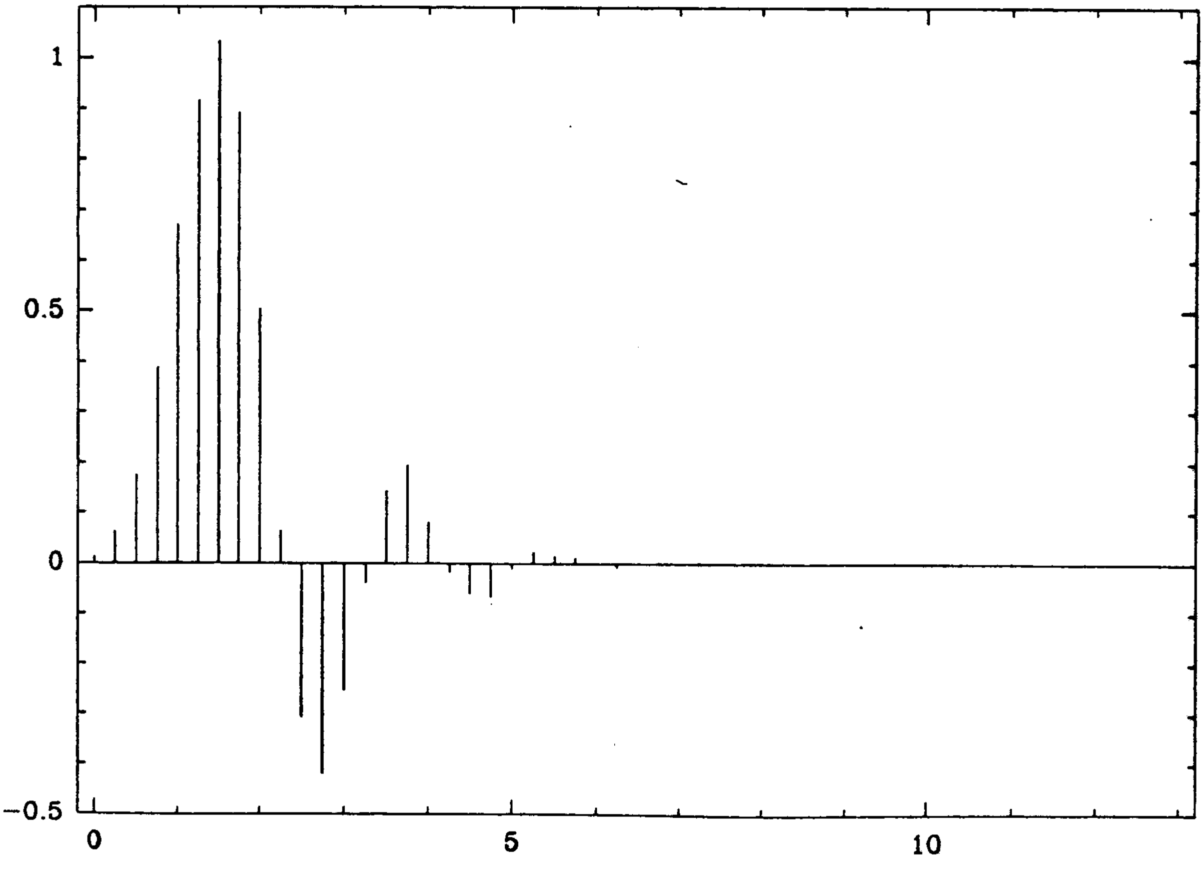
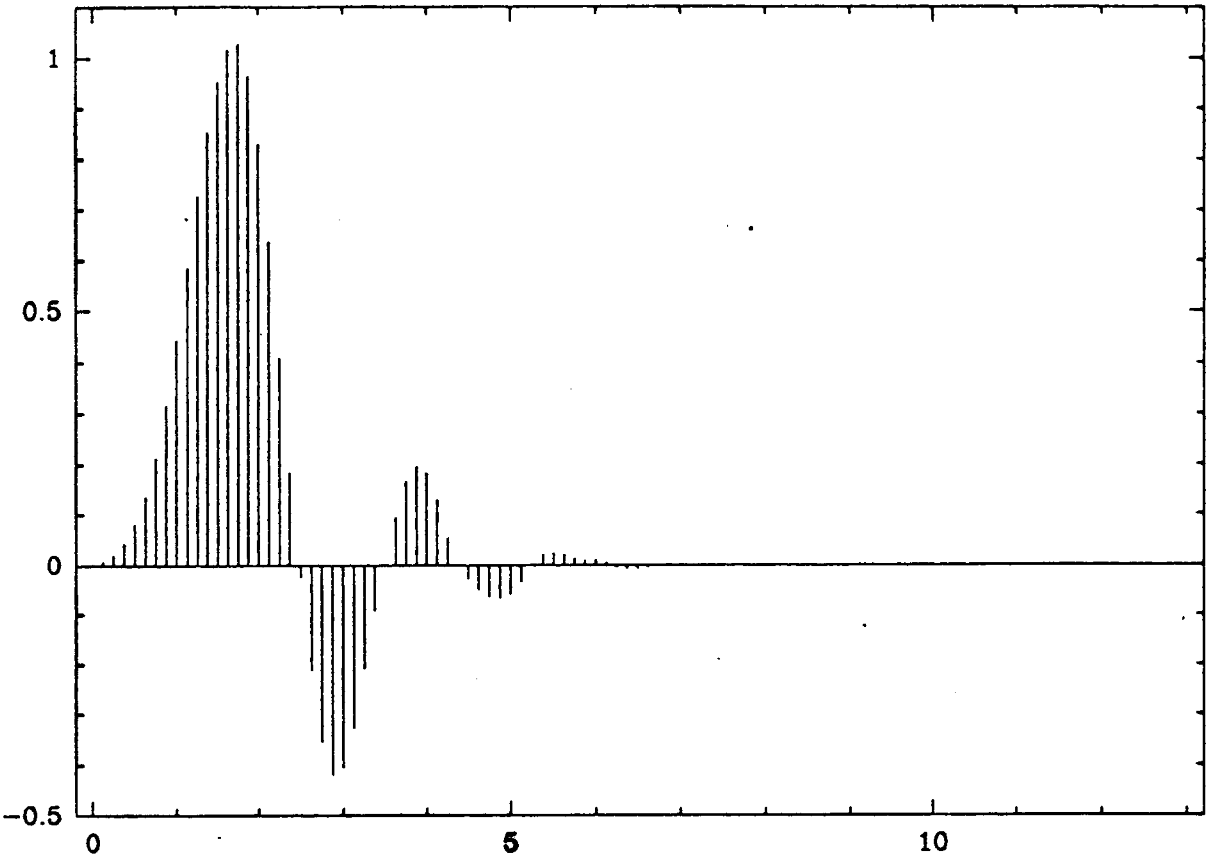


Fig. 1 (cont'd)

j = 3



j = 6

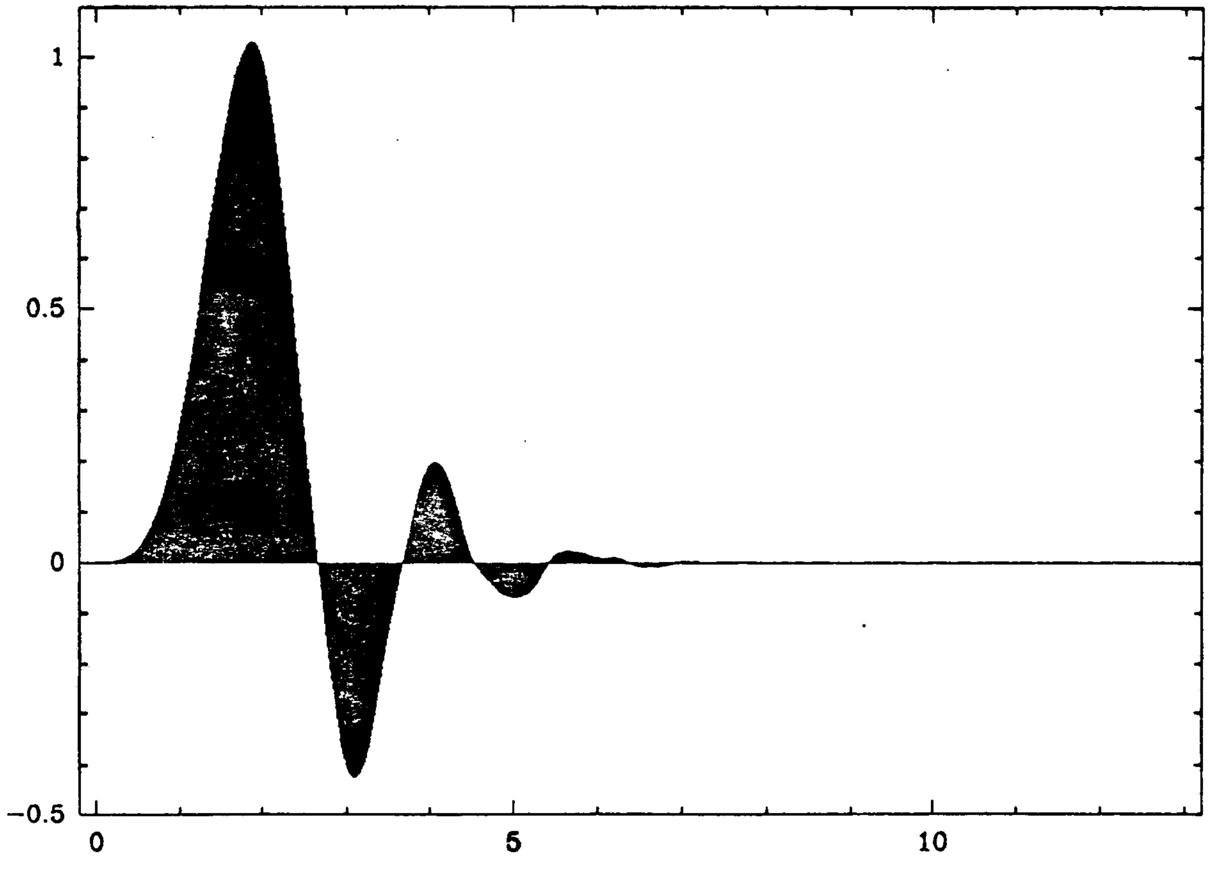
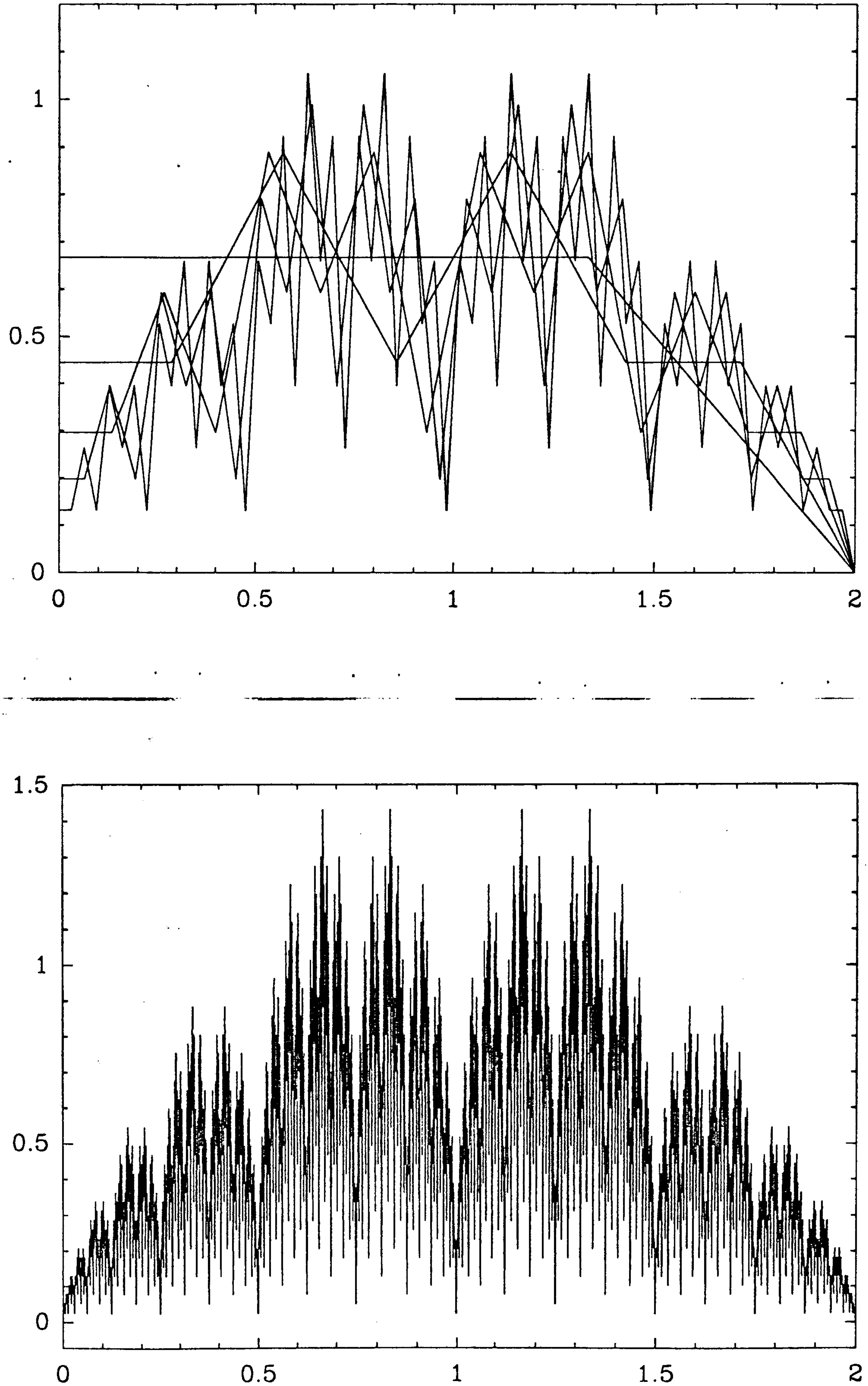


Fig. 2



=1,...,6
(a)

j=8
(b)

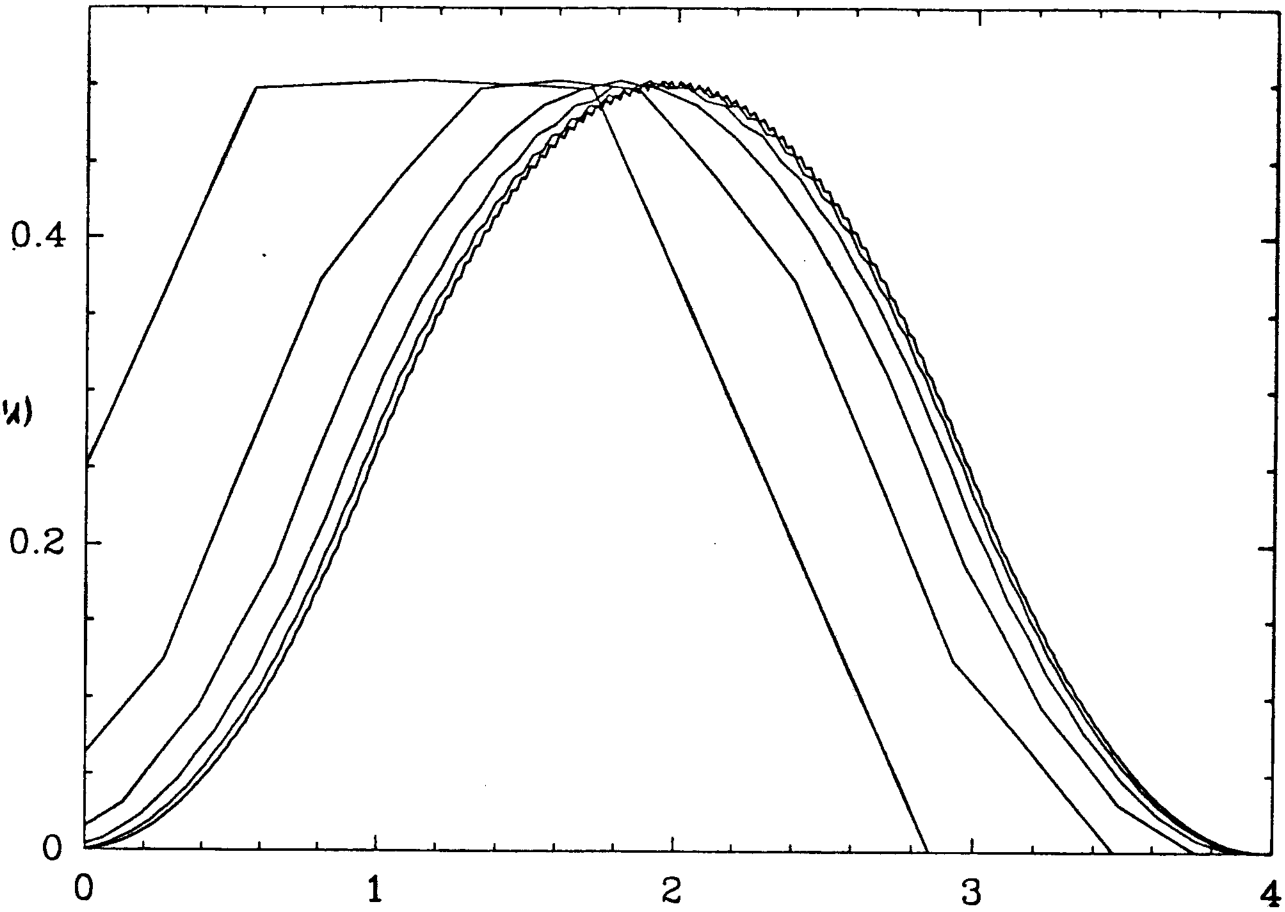
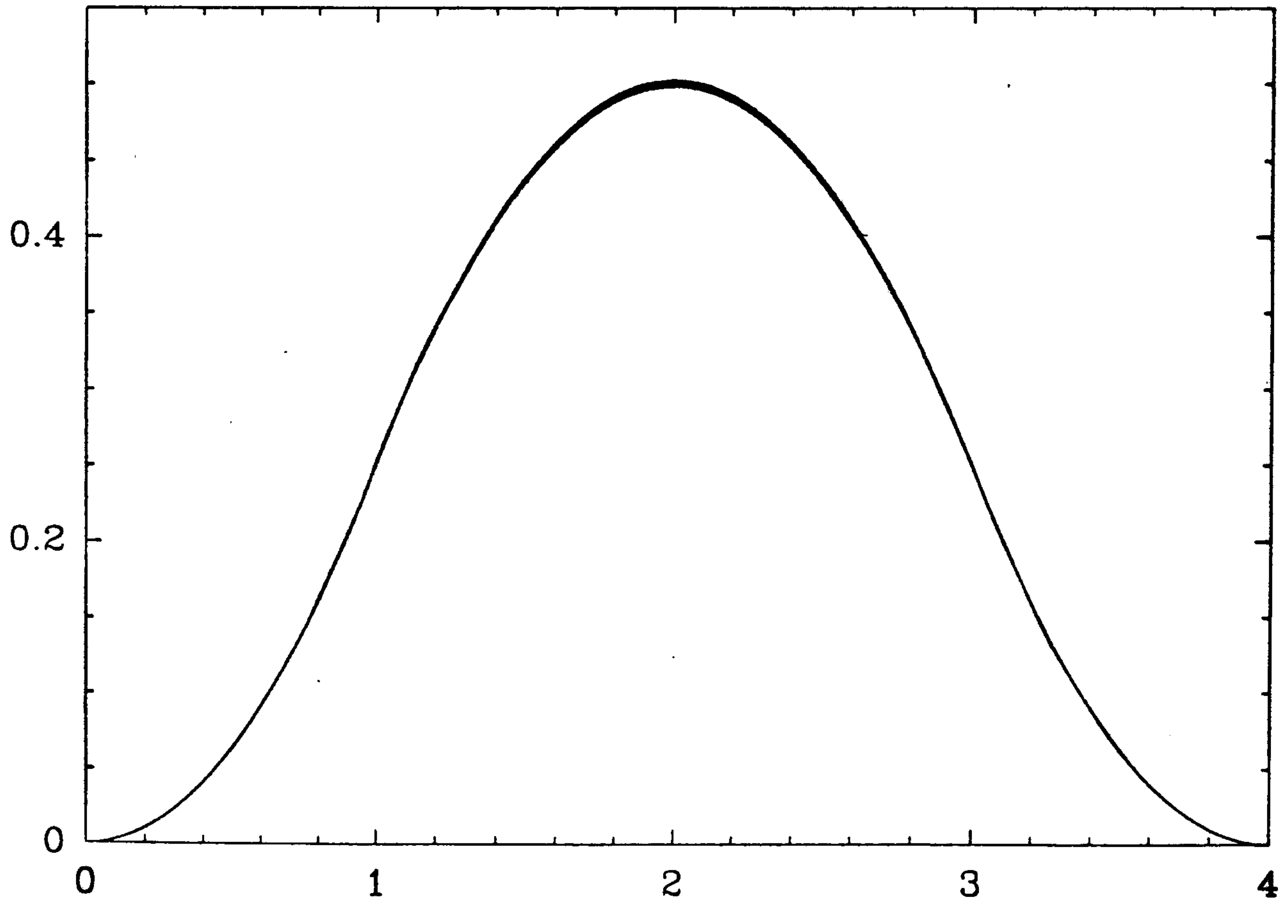


Fig. 2 (cont'd)

j=1,..., 6

(c)

8 iter



j=8

(d)

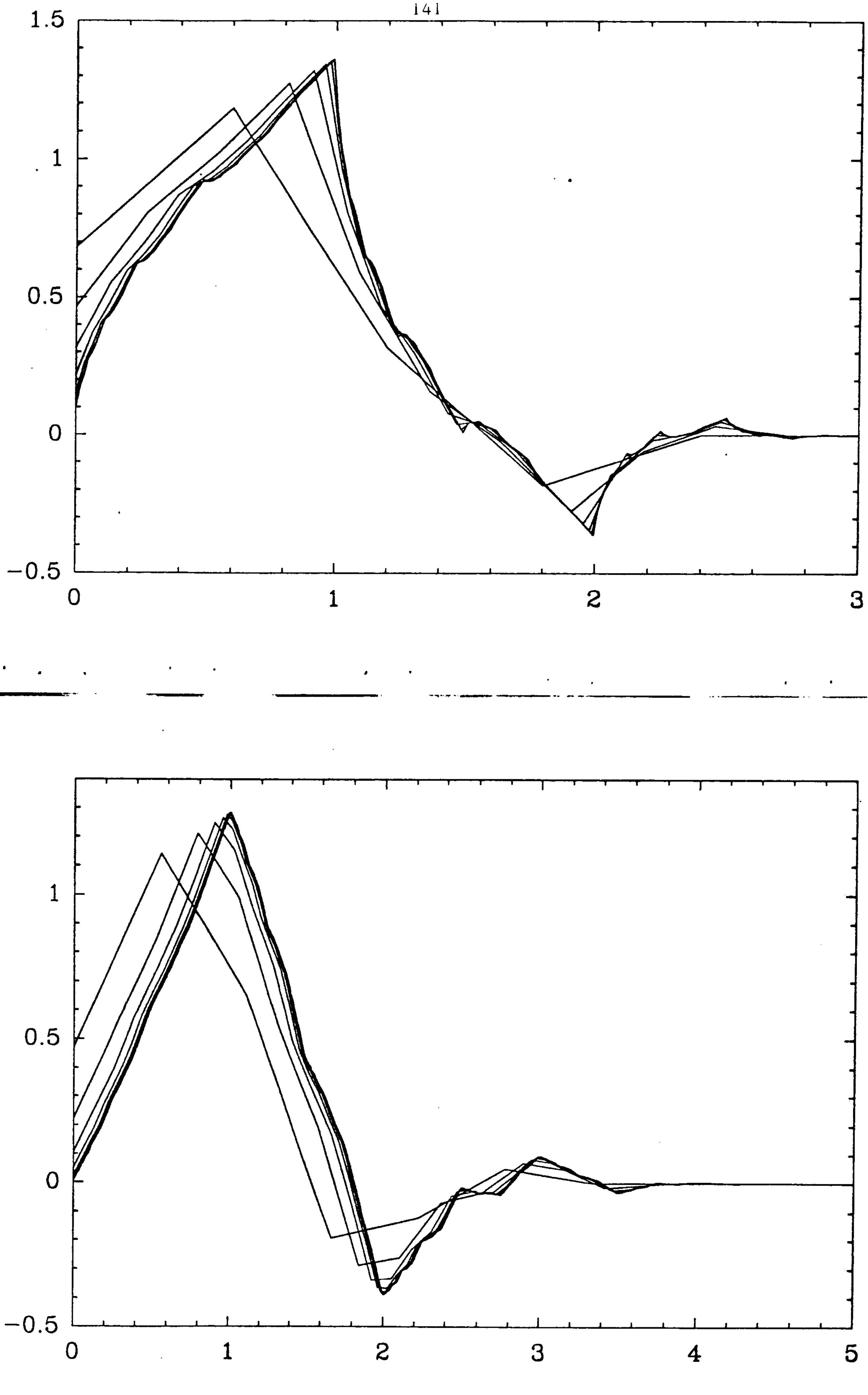
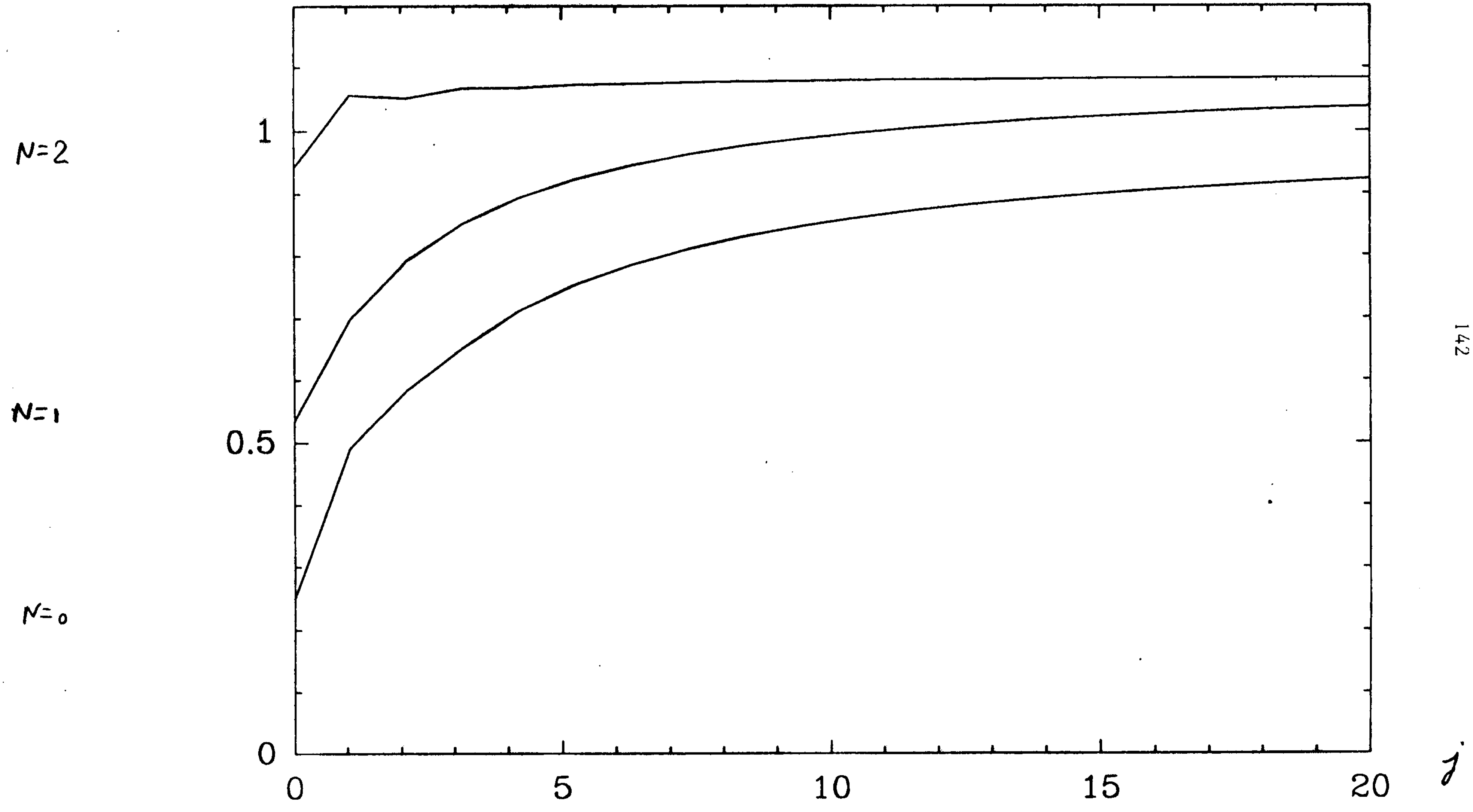


Fig. 3

(a)

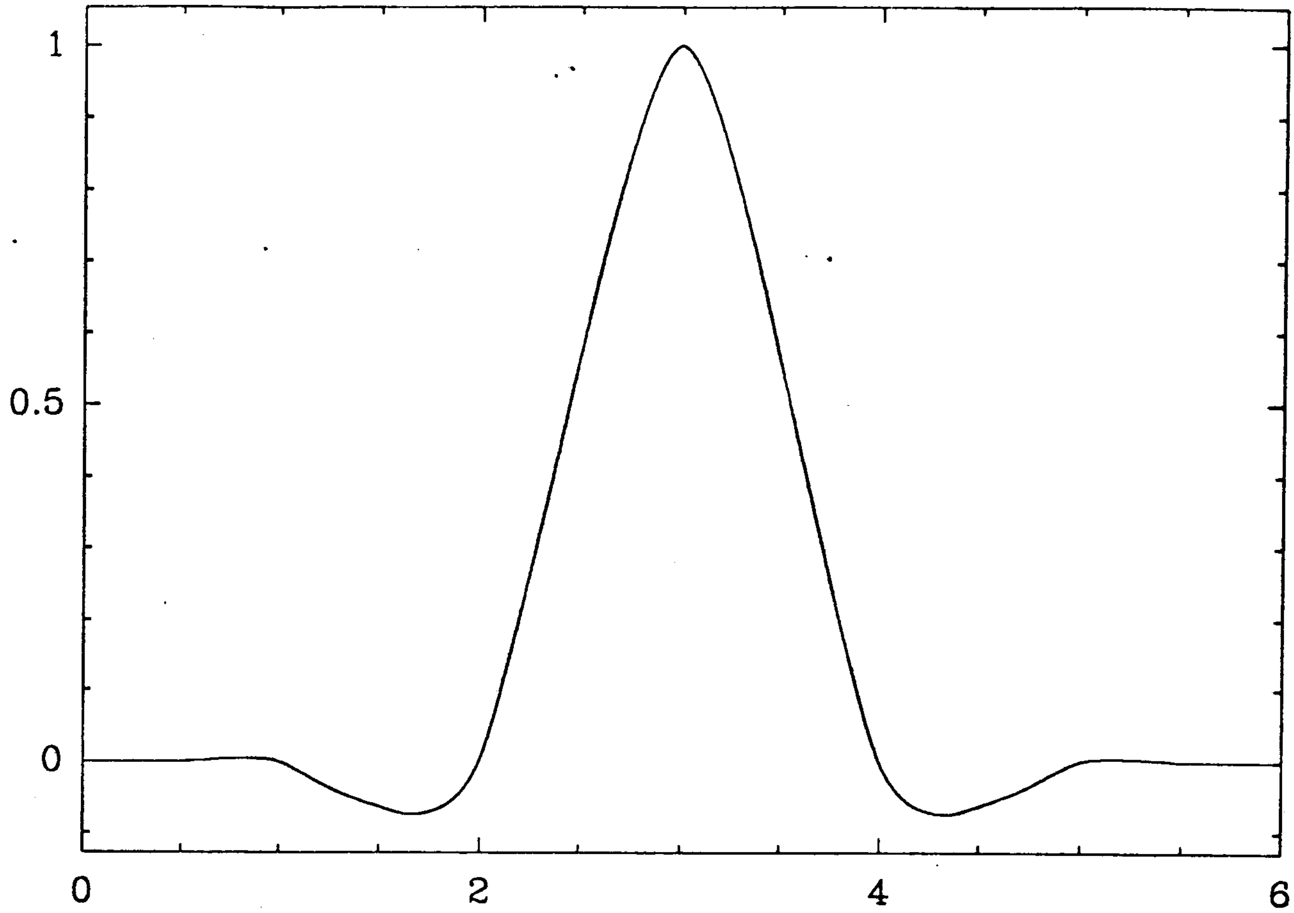
(b)

Fig. 4.



ig 5

$= -\frac{1}{16}$



$= 0$

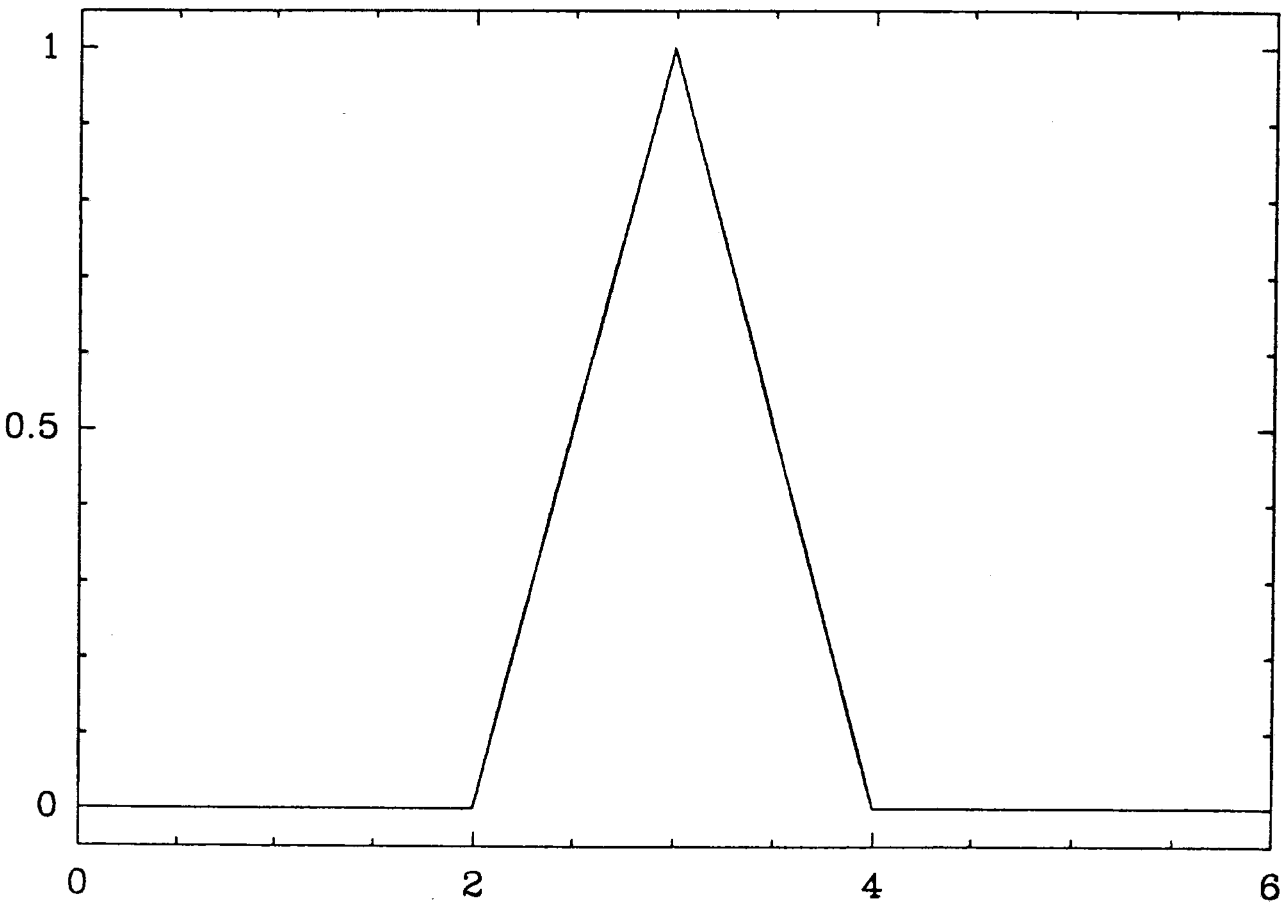
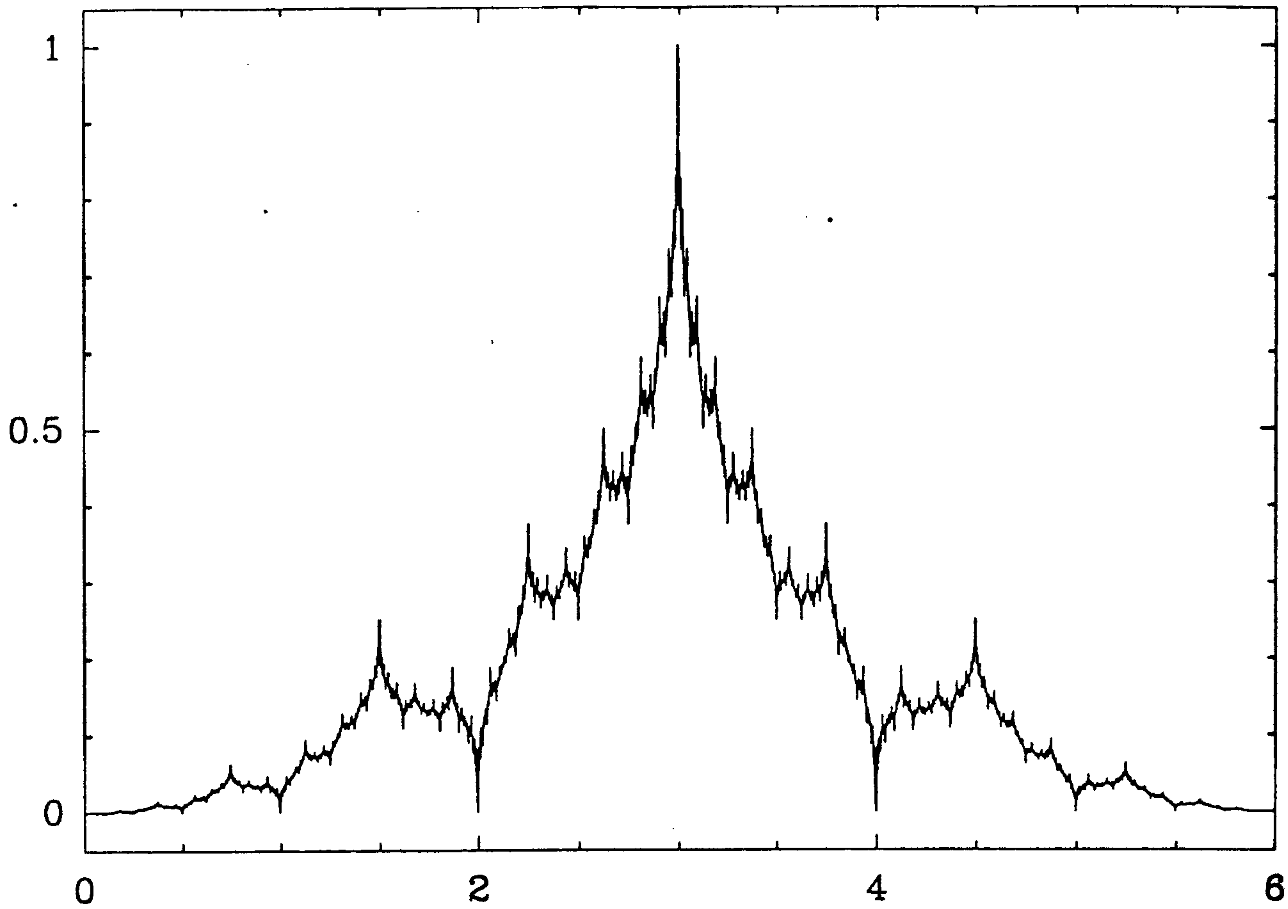


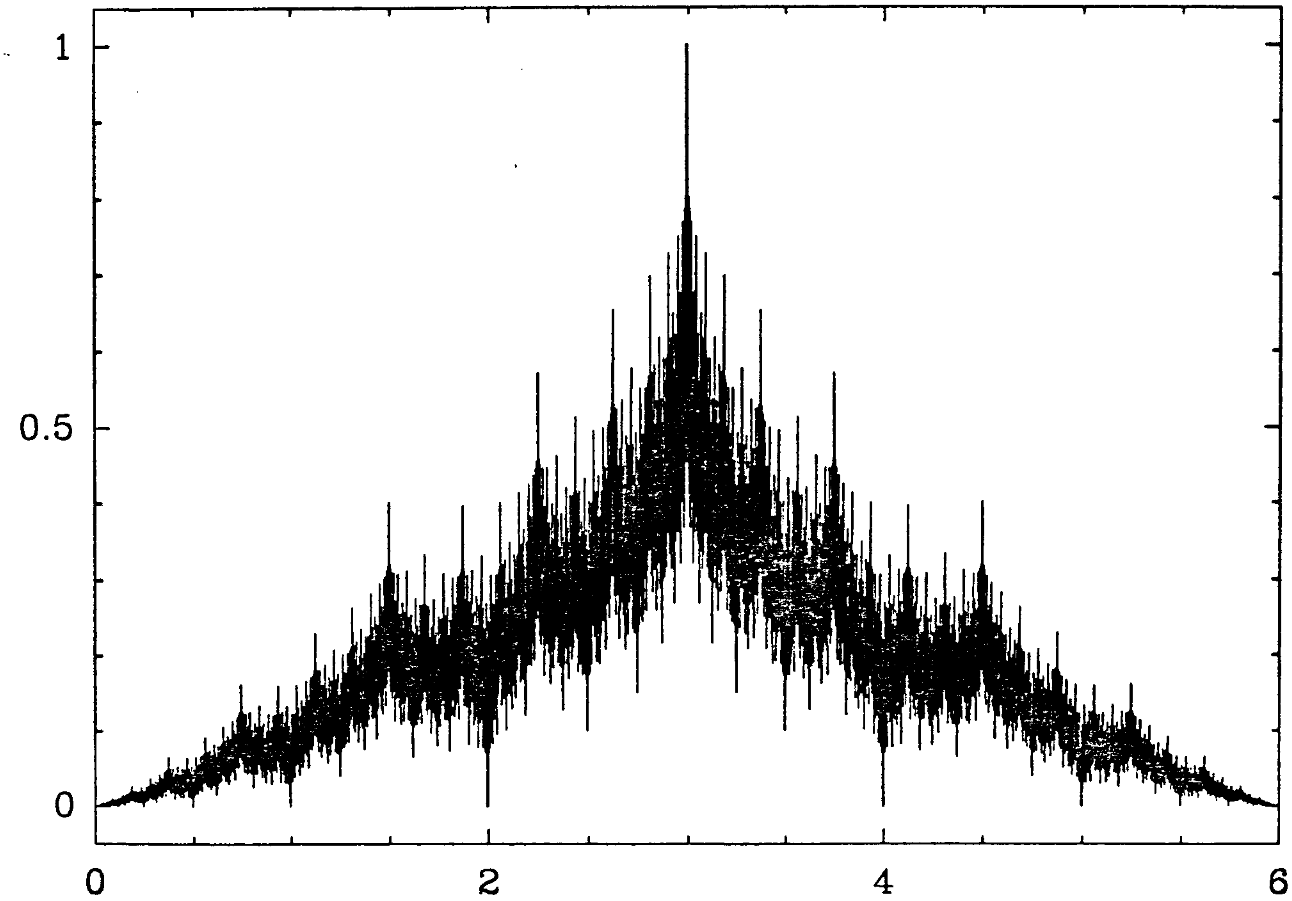
Fig.5
(cont'd)

= $\frac{1}{4}$



ame--

= 0.4



ARTICLE 4

"Fast Algorithms for Discrete and Continuous Wavelet Transforms", par O. RIOUL et P. DUHAMEL

IEEE Trans. Info. Theory, soumis en Fev.. 1990

Fast Algorithms for Discrete and Continuous Wavelet Transforms

Olivier Rioul and Pierre Duhamel
CNET, Centre Paris B
CNET/RPE/ETP
38-40, rue du Général Leclerc
92131 Issy-Les-Moulineaux
France

Feb. 1991

Submitted IEEE Trans. Info. Theory
Special issue on Wavelets

Abstract - This paper first derives the precise conditions under which the Wavelet Transforms of analog signals are computed exactly using discrete algorithms. Implementation issues for the Discrete Wavelet Transform, Wavelet Series, and the Continuous Wavelet Transform are addressed. Fast algorithms, either FFT-based or Fast-FIR-based are then derived. These algorithms reduce the arithmetic complexity of wavelet transforms in any practical case of interest.

INTRODUCTION

Wavelet Transforms are becoming of increasing interest for various tasks in Signal Processing. Their continous version, known as the Continuous Wavelet Transform [12], [13], [18], [36] provides an alternative [10], [27], [29] to classical time-frequency representations of signals such as the Short-Time Fourier Transform. Their semi-discrete version (Wavelet Series), as well as their fully discrete version (Discrete Wavelet Transform) have been recently used in various signal coding schemes, including image compression [1], [21] and in various tasks in computer vision [19], [20]. This paper intends to provide efficient algorithms for computing these various wavelet transforms, hence concentrates on their computational structure rather than on their properties.

The computational structure of the Discrete Wavelet Transform (DWT) was soon recognized to be an octave-band filter bank [4], [19], [28], [34]. The primary purpose of this paper is to show that filter banks also arise naturally when implementing Wavelet Series and Continuous Wavelet Transforms, thereby precising the strong links existing between the different types of wavelet transforms.

As far as implementation issues are concerned, filter banks have the interesting property that they are easily implemented by repetitive application of identical cells: they have a regular structure. In this paper, we derive fast algorithms by applying various computational techniques on these elementary cells.

Wavelet Transforms can be seen as an inner product of the signal with various analysing functions, named *wavelets*. The wavelets to be used at scale a and around time b are deduced from a basic prototype $\psi(t)$ (which can be thought of as any band-pass function) by

$$\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{1}$$

where a is a dilation/contraction factor, b a time-shift, and $1/\sqrt{a}$ a factor that ensures energy preservation. Roughly speaking, all wavelet tranforms compute the coefficients representing the signal $x(t)$ in a basis of wavelets (1). Various discretization schemes are possible, yielding various types of wavelet transforms. More precisely, we use the following terminology that parallels the classical one used for Fourier Transforms:

Continuous Wavelet Transform (CWT): continuous time (t) and time-scale parameters (b,a).

Wavelet Series' Transform (WST): continuous time, but discrete time-scale parameters.

Discrete Wavelet Transform (DWT): Both time and time-scale parameters are discrete.

(the WST is sometimes called DWT in the litterature [11], [20]).

This paper is organized as follows: Section II addresses the links existing between the WST and the DWT, and shows that the WST can be computed as a DWT on a prefiltered version of the signal, at least when scale is sampled as $a=2^j$. Denser sampling in scale is also discussed. Section III then provides fast versions of the DWT. The first one, based on FFT's, is most efficient for large wavelet prototypes (corresponding to filter lengths $L \geq 16$), while the second one is of greater interest for short-length ones. Note that the set of algorithms provided here allows for a reduction of the arithmetic complexity in any case of interest. Section IV shows that successive WST computations provides a full CWT, and derives a modification of the basic DWT computational cell, suitable for the CWT computation (the organization of the computation is slightly changed because a denser set of outputs is required). Tables of the required number of operations are provided for all algorithms presented in this paper.

The material of this paper develops and extends previous papers [25], [26]. A brief comparison with some previously published algorithms [2], [9], [11], [17], [19], [30], [32], [34] is made.

I. Implementing Wavelet Series

Before deriving precise algorithms for implementing Wavelet Series, we first outline some properties of the WST and the DWT.

A. The Wavelet Series' Transform (WST).

Wavelet series

$$x(t) = \sum_{j \in \mathbf{Z}} \sum_{k \in \mathbf{Z}} c_{j,k} \, \tilde{\psi}_{j,k}(t) \tag{2}$$

decompose an analog signal $x(t)$ into a basis of analog wavelets $\tilde{\psi}_{j,k}(t)$. These wavelets usually correpond to discretized parameters $a=2^j$ (an "octave-by-octave" computation), and $b=k2^j$ in (1):

$$\tilde{\psi}_{j,k}(t) = 2^{-j/2} \tilde{\psi}(2^{-j} t - k), \qquad j, k \in \mathbf{Z}. \tag{3}$$

Other choices are sometimes considered [5] (see section I.D).

The Wavelet Series' Transform (WST) computes the wavelet series' coefficients $c_{j,k}$ as

$$\mathrm{WST}\{x(t);\, a=2^j,\, b=k2^j\} = c_{j,k} = \int x(t) \psi^*_{j,k}(t)\, dt, \tag{4}$$

where the "analysis wavelets"

$$\psi_{j,k}(t) = 2^{-j/2} \, \psi(2^{-j}t - k), \qquad j, k \in \mathbf{Z} \tag{5}$$

are chosen to guarantee perfect reconstruction in the synthesis process (2), and are not necessarily the same ones as the "synthesis wavelets" $\tilde{\psi}_{j,k}(t)$. Equation (2) is therefore the expression of the Inverse Wavelet Series' Transform (IWST)

$$\mathrm{IWST}\{c_{j,k}\} = \sum_{j \in \mathbf{Z}} \sum_{k \in \mathbf{Z}} c_{j,k} \, \tilde{\psi}_{j,k}(t). \tag{6}$$

The perfect reconstruction requirement: IWST{WST{$x(t)$}} $= x(t)$, can be met by many choices of wavelet prototypes $\psi(t)$ and $\tilde{\psi}(t)$. These choices either fall in the theory of "wavelet frames" [5], [14], [15] or in the theory of "biorthonormal" or "orthonormal" wavelets [4], [19], [20], [28], [35], [36]. The resulting constraints on the shape of wavelets can sometimes be used to (slightly) reduce the computational load of the WST. Nevertheless, we focus here on implementation issues, not on the wavelet design. Therefore, to be as general as possible, the constraints on $\psi(t)$ and $\tilde{\psi}(t)$ will not be taken into account.

Various data compression schemes, although fully discrete in nature, have been described using the WST formalism, because wavelet coefficients are indexed by a (relatively sparse) discrete set of numbers (j,k). A two-dimensional version was successfully used for image compression [1], [20], [21]. Wavelet series are also closely related to octave-band filter banks used in speech coding [32] *via* the Discrete Wavelet Transform.

B. The Discrete Wavelet Transform (DWT)

The DWT formalism closely follows the Wavelet Series' formalism, the main difference being that one deals with discrete-time signals $x[n]$, $n \in \mathbf{Z}$. The DWT coefficients are (compare with (4), (5))

$$\mathrm{DWT}\{x[n];\, 2^j,\, k2^j\} = c_{j,k} = \sum_n x[n]\, h_j^*[n-2^j k], \qquad j \geq 1,\ j,\ k \in \mathbf{Z}, \tag{7}$$

and the inverse transform (IDWT) reconstructs the signal by (compare with (2), (3))

$$\mathrm{IDWT}\{c_{j,k}\} = \sum_{j=1}^{+\infty} \sum_{k \in \mathbf{Z}} c_{j,k}\, \tilde{h}_j[n-2^j k] \tag{8}$$

This latter expression equals $x[n]$ only if both $h_j[n]$ and $\tilde{h}_j[n]$ satisfy specific constraints [4]. Since we use here both DWT and IDWT as intermediate computation steps to compute various wavelet transforms, these constraints do not always hold in the context of this paper and, unless mentioned, are not considered.

Note that the octave parameter j is here restricted to $j \geq 1$ so that the sampling rate of $c_{j,k}$ in k, which is 2^{-j}, is less than or equal to half the sampling rate of $x[n]$. Therefore, from a set of N input points, the DWT computes about $N(1/2 + 1/4 + \ldots + 2^{-j} + \ldots) \cong N$ wavelet coefficients.

The discrete "analysis wavelets" $h_j[n-2^j k]$ and synthesis "wavelets" $\tilde{h}_j[n-2^j k]$ are discrete substitutes for $\psi_{j,0}(t-2^j k) = \psi_{j,k}(t)$ and $\tilde{\psi}_{j,0}(t-2^j k) = \tilde{\psi}_{j,k}(t)$, respectively. They are stretched (or dilated) versions of two prototypes $h[n]$ and $\tilde{h}[n]$, as in the continuous-time case. In constrast with continuous-time wavelets, however, a dilation by 2^j is not so easily expressed as

$$\psi(t) \quad \rightarrow \quad 2^{-j/2}\, \psi(2^{-j} t) = \psi_{j,0}(t)$$

since the $h_j[n]$'s and $\tilde{h}_j[n]$'s are discrete sequences. We therefore define the discrete equivalent of a dilation (by a factor two) as the following interpolation processes [28], [29], [35]:

$$h_j[n] \quad \rightarrow \quad h_{j+1}[n] = \sum_k h_j[k]\, g[n-2k]$$

$$\tilde{h}_j[n] \quad \rightarrow \quad \tilde{h}_{j+1}[n] = \sum_k \tilde{h}_j[k]\, \tilde{g}[n-2k]. \tag{9}$$

This defines the wavelets $h_j[n]$ and $\tilde{h}_j[n]$ by induction starting from $h_1[n] = h[n]$ and $\tilde{h}_1[n] = \tilde{h}[n]$. This "dilation" operators use a "scaling sequence" $g[n]$ (or $\tilde{g}[n]$), which is the impulse response of a low-pass interpolation filter.

The DWT and the IDWT can always be seen as an analysis and synthesis *octave-band filter bank* [23], [32], depicted in Fig.1. In this filter bank the basic filters are precisely $g[n]$, $h[n]$, $\tilde{g}[n]$ and $\tilde{h}[n]$. It can be shown that DWT and IDWT are true inverse transforms of eachother if and only if this filter bank allows perfect reconstruction [3], [23], [28], [31], [32], [35].

C. DWT Computation of the WST.

1. *Basic Assumptions.* The whole paper is centered around the computation of wavelet transforms of a continuous-time signal $x(t)$ by means of the DWT, which is the only wavelet transform we can compute exactly on a computer anyway. Since the DWT acts on discrete-time signals, we need a precise writing of continuous-time signals in terms of their discrete counterparts. This requires some knowledge about $x(t)$. Once these assumptions hold with sufficient accuracy, the rest of the computations will be exact.

Assume that $x(t)$ is related to its discrete version $x[n]$ by a (possibly non-perfect) D/A converter of the form

$$x(t) = \sum_n x[n]\, \chi(t-n). \tag{10}$$

Throughout the paper we assume that the sampling rate of $x(t)$ is 1. A typical choice is the zero-order holder, for which $\chi(t) = 1$ for $0 \le t \le 1$, 0 elsewhere. This choice corresponds to approximating a continuous curve by a piecewise constant one.

Also assume a similar correspondance for all the wavelets used in the WST:

$$2^{-j/2}\, \Psi(2^{-j}t) = \sum_n h_j[n]\, \phi(t-n), \quad j \ge 1, \tag{11}$$

where $\phi(t)$ is some interpolating function of time. The sequences $\{h_j[n]\}$ are seen as the discrete versions of the wavelets $2^{-j/2}\, \Psi(2^{-j}t)$, therefore they must follow, as in the DWT, the "discrete dilation" scheme (9).

Although (10), (11) are assumed to hold exactly in the following, when met with sufficiently high accuracy, the resulting DWT computation is a good approximation of the WST.

Since (11) must be met for all $j \geq 1$, it is not usable as such, and it is more practical to replace this set of equations by a single one on the basic wavelet (j=1), plus a writing of the equivalence of the continuous and discrete dilations. This is obtained by writing (11) in two equivalent forms,

$$2^{-(j+1)/2}\,\Psi(2^{-(j+1)}t) = \sqrt{2}\sum_n h_j[n]\,\phi(t/2-n)$$

$$= \sum_n \left(\sum_k h_j[k]\; g[n-2k]\right)\phi(t-n).$$

By identification, $\phi(t)$ satisfies a "two-scale difference equation" [6]

$$\phi(t) = \sqrt{2}\sum_n g[n]\,\phi(2t-n). \tag{12}$$

Now clearly (11) is equivalent to (12), plus (11) written for j=1:

$$1/\sqrt{2}\;\Psi(t/2) = \sum_n h[n]\,\phi(t-n). \tag{13}$$

We shall see that the DWT computation of the WST is exact once (12) and (13) are assumed to hold. It is therefore crucial to choose a good interpolating function $\phi(t)$ satisfying (12) such that the basic wavelet $\psi(t)$ can be accurately approximated by (13).

2. *Examples of interpolating functions*. These are two simple examples of $\phi(t)$:

Assume that $\psi(t/2)$ is band-limited to [-1/2; 1/2]. By Nyquist's sampling theorem, $\phi(t)= \sin \pi t / \pi t$ is a solution that ensures $h_j[n] = 2^{-j/2}\,\psi(2^{-j}n)$ in (11). Equation (12) here holds with $g[n] = 1/\sqrt{2}\;\phi(n/2)$. However, this scheme involves an infinite, ideal low-pass filter $g[n]$, with slow decay as $n \to +\infty$, which makes this choice impractical.

Another solution for $\phi(t)$ is the basic spline interpolating function of some degree k, whose Fourier Transform is [3]

$$\phi(f) = \left(\frac{\sin \pi f}{\pi f}\right)^{k+1}.$$

In this case (13) reduces to a classical curve fitting problem. Rewriting (12) in the Fourier domain, one finds, within time-shifting,

$$g[n] = \tfrac{1}{\sqrt{2}}\binom{k+1}{n}, \qquad n=0, \ldots, k+1,$$

which is refered to as a (FIR) binomial filter [3], [32].

3. *Derivation of the basic algorithm.* Expand (10), (11) into the expression of the WST (4). One obtains

$$\mathrm{WST}\{x(t);\, 2^j, k2^j\} = \int\left(\sum_m x[m]\chi(t-m)\right)\left(\sum_n h_j[n]\phi(t-2^j k-n)\right)^* dt$$

$$= \sum_n\left(\sum_m x[m]\int\chi(t-m)\phi^*(t-n)\,dt\right)h_j^*[n-2^j k]$$

This immediately gives the following WST algorithm.

$$\mathrm{WST}\{x(t);\, 2^j, k2^j\} = \mathrm{DWT}\{x'[n],\, 2^j, k2^j\}, \tag{14}$$

where $x'[n]$ is a corrected version of $x[n]$, namely the discrete convolution

$$x'[n] = x[n] * f[n], \text{ where } f[n] = \int\chi(t)\phi^*(t-n)\,dt. \tag{15}$$

Note that in the case of a perfect sampling, we have $h_j[n] = 2^{-j/2}\,\psi(2^{-j}n)$, and (14) reduces to a simple discretization of the integral defining the WST (4). Compared to the perfect sampling case, our approach allows a greater flexibility in the curve fitting (13) of the wavelets, hence on the accuracy of the computation (14).

Using (10), equation (15) can be rewritten as

$$x'[n] = \int x(t)\phi^*(t-n)\,dt. \tag{16}$$

If $\chi(t) = \phi(t)$ in (10) and if the family of functions $\phi(t-n)$ ($n \in \mathbb{Z}$) is orthonormal, then prefiltering is avoided: $x'[n] = x[n]$ and (14) reduces to the *Mallat algorithm* [4], [19], which was specifically derived for orthonormal wavelets $\psi_{j,k}(t)$. In [19], [22], equation (16) has been further interpreted in terms of multiresolution spaces as being the approximation coefficients of $x(t)$ at the 0th octave (j=0). Similarly the synthesis Mallat algorithm is a special case of the IWST computation derived in section I.*E*.

As a result of (14), a filter bank implementation of the WST *via* a DWT is possible within a much broader framework than the orthonormal case treated by Mallat, provided that a pre-filtering (15) is performed on the data. This result is used throughout this paper to derive various efficient wavelet transform algorithms based on DWT. The DWT is therefore the basic computational structure. However, it may be necessary to compute a WST on a set of points denser than the octave grid in the scale axis a=2^j. This is the aim of the following section, which extends (14) to obtain other points in scale.

D. Finer Sampling in Scale.

So far Wavelet Series' coefficients were computed on a dyadic grid in the time-scale plane $\{b$=$k2^j$, a=$2^j\}$ (see Fig.1). In applications such as signal analysis, it may be desirable to "oversample" this discretization, i.e. to compute a denser set of wavelet coefficients along the scale axis (computation on "M voices per octave" [5]). This means that $a = 2^j$ is replaced by

$$a = 2^{j+m/M},\ m=0,\ \dots,\ M\text{-}1 \ \text{("voices").}$$

Now, for each m, replace $\psi(t)$ by the slightly stretched version $2^{-m/2M}\,\psi(2^{-m/M}t)$ in the expression of $\psi_{j,k}(t) = 2^{-j/2}\,\psi(2^{-j}t\text{-}k)$. The wavelets basis becomes:

$$2^{-(j+m/M)/2}\,\psi(2^{-(j+m/M)}(t\text{-}k2^j)), \qquad j,\, k \in \mathbf{Z},\ m=0,\ \dots,\ M\text{-}1.$$

The obtained points in scale (a) and time-shift (b) parameters are shown in Fig. 2: this corresponds exactly to the desired WST computation on M voices per octave. Since this full set of wavelets is derived from M different (slightly stretched) wavelet prototypes, a WST computation on M voices per octave results from M successive applications of the octave-by-octave algorithm (14), each one corresponding the following prototype

$$2^{-m/2M}\,\psi(2^{-m/M}t),\ m=0,\ \dots,\ M\text{-}1. \tag{17}$$

Of course (13) must be satisfied for each of these M basic wavelets (17), which means that the coefficients of each discrete wavelet must be derived again for each m using the same procedure as explained in section I.*C*. Clearly, the whole algorithm requires about M times the computational load of the octave-by-octave algorithm (14).

Note that the separate computation of several WST's based on prototype wavelets (17) is not the best possible algorithm for an "M voices per octave" WST computation, because these prototype wavelets are related in a simple manner. However, we could not find a method that takes advantage of both time redundancy and scale redundancy when $a=2^{j+m/M}$. This paper uses mostly time redundancy, while the algorithm derived in [2] is mostly based on scale redundancy (see section III.*D*).

E. IDWT Computation of the IWST.

Just as the WST can be computed using a DWT, its inverse transform can be computed using an inverse DWT (8), under the same assumption (11) expressed on *synthesis* wavelets $\tilde{\psi}_{j,k}(t)$ for $j \geq 1$:

$$2^{-j/2}\tilde{\psi}(2^{-j}t) = \sum_n \tilde{h}_j[n]\tilde{\phi}(t-n). \tag{18}$$

Let $c_{j,k} = \text{WST}\{x(t),\, 2^j,\, k2^j\}$. Expand (18) into the expression of an IWST (6) (for $j \geq 1$). One obtains

$$\text{IWST}\{c_{j,k}\} = \sum_n\ y[n]\ \tilde{\phi}(t\text{-}n), \tag{19a}$$

where

$$y[n] = \text{IDWT}\{c_{j,k}\} \tag{19b}$$

The IWST thus results from the IDWT followed by a D/A converter (interpolation) associated to $\tilde{\phi}(t)$. The overall analysis/synthesis WST procedure is depicted in Fig.3: First the analog signal $x(t)$ is "sampled" (A/D converted) according to (10) to provide the discrete signal $x[n]$. This latter is then pre-filtered by $f[n]$ (15) to give the discrete signal $x'[n]$ which feeds the DWT algorithm (the analysis octave-band filter bank of Fig. 1). The synthesis part is obtained by an IDWT, followed the interpolation (D/A conversion) (19a).

Note that in this WST/IWST computation, the analysis and synthesis discrete wavelets do not necessarily form a perfect reconstruction filter bank pair. If by any chance the DWT/IDWT turns out to be a perfect reconstruction filter bank, i.e., if $y[n] = x'[n]$ (16), then it can be shown [3] that analysis and synthesis wavelets are "biorthonormal" [3] (at least for FIR filters $g[n]$, $h[n]$, $\tilde{g}[n]$ and $\tilde{h}[n]$) and we have

$$\int \phi(t-n)\tilde{\phi}^*(t-m)\,dt = \delta_{m,n} \tag{20}$$

Since $y[n] = x'[n]$, it follows that

$$\mathrm{IWST}\{\mathrm{WST}\{x(t)\}\} = \sum_n \left(\int x(u)\phi^*(u-m)\,du\right)\tilde{\phi}(t-n) \tag{21}$$

The right-hand side of (21) is a *projection* of $x(t)$ onto the subspace V spanned by linear combinations of the $\tilde{\phi}(t-n)$: indeed if $x(t)$ belongs to V, i.e., if $x(u) = \sum_n c_k \tilde{\phi}(t-k)$, then using (20), equation (21) reduces to $x(t)$.

Therefore, any general signal $x(t)$ which does not *a priori* belong to V is not exactly recovered in a WST/IWST computation using a DWT/IDWT. Only its projected approximation onto V is recovered. This loss of information is due to the "sampling" operation $x(t) \to x[n]$ (accordingly the DWT/IDWT is performed for $j \geq 1$ only). The only way to reconstruct $x(t)$ is to ensure that $x(t)$ belongs to V from the start (e.g. choose $\chi(t) = \tilde{\phi}(t)$ in (10), in which case $x'[n] = x[n]$).

This precises the fact stated above: the only approximation in our algorithm is in the "sampling", and in the way the continuous-time signal and wavelets are recovered from the discrete samples.

II. FAST ALGORITHMS FOR THE WST AND THE DWT

A. Preliminaries on Filter Bank Implementations of the DWT.

The aim of the following sections is to derive fast algorithms for the DWT/IDWT, and therefore for the WST/IWST *via* (14), (19). We assume real data and wavelets. The complex case can be handled without difficulty (it can be shown that fast algorithms presented in sections II.*B-D* require about twice more complexity in the complex case than in the real case). We also assume that the basic filters involved in a DWT/IDWT implementation (Fig.1) are FIR and have same *even* length L. We may always restrict to this case, padding impulse responses with zeros if necessary (which does not greatly influence the efficiency of the algorithm because the involved filters are of comparable length).

In the following, we restrict to the derivation of fast algorithms for the DWT only. Algorithms implementing the IDWT are easily deduced from any DWT algorithm as follows. If the wavelets form an orthogonal basis, the exact inverse algorithm is obtained by taking the Hermitian transpose of the DWT flowgraph. Otherwise only the structure of the inverse algorithm is found that way, the filter coefficients $g[n]$, $h[n]$ have to be replaced by $\tilde{g}[n]$, $\tilde{h}[n]$. In both cases, any DWT algorithm, once transposed, can be used to implement an IDWT, with exactly the same number of operations (multiplications and additions) per point.

The derivation of fast algorithms is primarily based on the reduction of arithmetic complexity. By "(arithmetic) complexity" we mean the total number of real multiplications and real additions required by the algorithm, per input point. For the DWT/IDWT, this is also the complexity per output point since the input and output rate in a DWT/IDWT are the same (see section I.*B*). Of course, complexity is not the only relevant criterion. For example, regular computational structures (i.e. repetitive application of identical basic computational cells) is also important for implementation issues.

A straightforward computation of the DWT implements (7) exactly as written, all discrete wavelets $h_j[n]$ being precomputed using (9). This method does not take advantage of the dilation property of wavelets (9) *inside* the algorithm, and therefore does not use the filter bank structure of Fig.1. Since the length of $h_j[n]$ is $(L-1)(2^j-1)+1$, we end up, at each octave j, with $(L-1)(2^j-1)+1$ real multiplications and $(L-1)(2^j-1)$ real additions for each set of 2^j inputs. Assuming the DWT is computed on J octaves ($j=1, \ldots, J$), this yields

$$\begin{aligned} &J(L-1)+1 \text{ mults/point.} \\ &J(L-1) \text{ adds/point.} \end{aligned} \qquad (22)$$

This complexity increases linearly with the number of octaves J.

A better implementation uses the DWT filter bank implementation of Fig.1. Owing to the decomposition of the computation into *elementary cells* and to subsequent subsampling operations at each stage, it can be shown that the complexity is significantly reduced compared to the "naive" method above. More precisely, the set of operations to be performed for one elementary cell at the jth octave (Fig. 1(a)) is, on one hand, the "wavelet filtering" by $h[n]$ which directly provides the wavelet coefficients at the considered octave j, and on the other hand, the *down-scaling operation* (filtering by $g[n]$ plus decimating) which is necessary to address the next cell. A direct implementation of the filters $g[n]$ and $h[n]$ plus decimation requires $2L$ multiplications and $2(L-1)$ additions for every set of two inputs, i.e.,

$$L \text{ mults/point/cell}$$
$$L\text{-}1 \text{ adds/point/cell} \qquad (23)$$

for each elementary cell. Since the cell corresponding to the jth octave has input subsampled by 2^{j-1}, the total complexity required by a filter bank implementation of the DWT on J octaves is $(1+ 1/2 + 1/4 + ... + 1/2^{J-1}) = 2(1\text{-}2^{-J})$ times the complexity (23), i.e.,

$$2L(1\text{-}2^{-J}) \text{ mults/point}$$
$$2(L\text{-}1)(1\text{-}2^{-J}) \text{ adds/point} \qquad (24)$$

This is significantly better than (22): the DWT is here roughly equivalent, in terms of complexity, to one filter of length $2L$. The complexity remains bounded when J increases.

The fact that the total complexity of the DWT equals $2(1\text{-}2^{-J})$ times the complexity of one elementary cell is general: it occurs in any filter bank implementation of the DWT, provided that each cell requires the same complexity. In order to derive faster algorithms, it is therefore sufficicent to apply fast computational techniques to one elementary cell only. In the next few sections we propose two classes of fast algorithms: the first one is based on the FFT and the other one is based on short-length FIR filtering algorithms.

Note that in the elementary DWT cell of Fig.1(a), filters do not appear as a whole: they are always followed by subsampling, that discards one output out of two. However it is well known that reducing the arithmetic complexity of an FIR filter implementation is obtained by grouping the computation of several *successive* outputs [24]. Since these successive outputs do not appear in subsampled filters, it is necessary to rewrite the computations in such a way that "true" filters appear. To do that, it is convenient to use z-transform notation

$$X(z) = \sum_n x[n]\, z^{-n},$$

for any sequence $x[n]$. The "biphase decomposition" [32], [35] of $X(z)$ consists in considering separately the odd and even terms of $X(z)$, i.e., the odd and even-indexed samples of $x[n]$:

$$X_0(z) = \sum_n x[2n]\, z^{-n} \text{ and } X_1(z) = \sum_n x[2n+1]\, z^{-n}, \text{ i.e.,}$$
$$X(z) = X_0(z^2) + z^{-1}\, X_1(z^2). \qquad (25)$$

Apply the biphase decomposition (25) to the input $X(z)$ of one cell, and to both filters $G(z)$ and $H(z)$ involved in the computation. The output $Y(z)$ of the elementary cell resulting from the down-scaling operation is obtained by first filtering by $G(z)$, then subsampling. Since we have

$$G(z)\, X(z) = G_0(z^2)X_0(z^2) + z^{-2}\, G_1(z^2)X_1(z^2) + \text{odd terms},$$

selecting the even terms of this expression gives $Y(z) = G_0(z)X_0(z) + z^{-1}\, G_1(z)X_1(z)$. In other words, $Y(z)$ is now obtained as follows. First extract the even and odd-indexed input samples $X_0(z)$ and $z^{-1}X_1(z)$ *as they flow* (hence the delay factor z^{-1} for odd-indexed samples). Then filter by $L/2$-tap filters

$G_0(z)$ and $G_1(z)$, respectively, and finally add the results. The other output of the elementary cell (the one corresponding to the filter $H(z)$) is similarly obtained from $H_0(z)$ and $H_1(z)$.

The resulting flow graph of the basic cell is depicted in Fig. 4 (the corresponding IDWT cell is simply obtained by flow graph transposition). Compare with Fig. 1(a): there are now four "true" filters of length $L/2$, whose impulse responses are the decimated initial filters $G(z)$ and $H(z)$.

B. An FFT-Based WST/DWT Algorithm.

This method simply computes the four $L/2$-tap filters of Fig. 4 using FFT's. The cell input is blocked B samples by B samples, hence the filters' inputs, being decimated by two, are blocked on $B/2$ samples. The filter convolutions are performed in a classical way as the IFFT of a product of the input FFT by the filter FFT. Since the latter can be precomputed once and for all during the algorithm, only one IFFT and one FFT are required per filter, for each input block. However this does not give a true filter convolution, but a *cyclic* convolution [24]. Therefore some time-processing must be done in order to avoid wrap-around effects. There are two well-known methods for this, called overlap-add and overlap-save methods [24]. Both are transposed of eachother and require exactly the same complexity. For one filter of length $L/2$, with input block length $B/2$, wrap-around effects are avoided if the FFT-length N satisfies

$$N \geq L/2 + B/2 - 1. \tag{26}$$

Therefore the block length is determined according to (26) as

$$B = 2N - (L-2). \tag{27}$$

From now on we use the "split-radix" FFT algorithm [8] which has, among practical FFT algorithms, the best known complexity for lengths $N = 2^n$ [8]. For real data, it requires (for both FFT's and IFFT's)

$$\begin{gathered} 2^{n-1}(n-3) + 2 \text{ (real) mults} \\ 2^{n-1}(3n-5) + 4 \text{ (real) adds.} \end{gathered} \tag{28}$$

Each elementary cell is performed with an identical structure shown in Fig.5. The input is first divided into even and odd-indexed sequences. Then a length-N FFT is performed on each decimated input, and four frequency-domain convolutions are performed by multiplying the (Hermitian symmetric) input FFT by the (Hermitian symmetric) filters' FFT. This requires $4N/2$ complex multiplications for the four filters. Finally two blocks are added ($2N/2$ additions) and two IFFT's are performed. Assuming that a complex multiplication is done with three real multiplications and three real additions [24], this gives

$$2\ \mathrm{FFT}_N + 4.3 N/2 \text{ mults} + 4.3 N/2 \text{ adds} + 2N/2 \text{ adds} + 2\ \mathrm{IFFT}_N$$

per cell, for B inputs, i.e.,

$$\frac{n2^{n+1}+8}{2^{n+1}-(L-2)} \text{ mults/point/cell}$$

$$\frac{(3n-1)2^{n+1}+16}{2^{n+1}-(L-2)} \text{ adds/point/cell} \qquad (29)$$

For a given length L there is an optimal value of B=2N-(L-2), i.e., an optimal value of $N=2^n$ that minimizes (29). Table I and II show the resulting minimized complexities for different lengths L.

After one cell is computed, wrap-around effects are eliminated in the time-domain. To enter the next cell, one could immediately feed the previous cell's outputs, whose number would be halved at each stage. But we have seen that FFT-based schemes are most efficient for an optimized value of the block length B (at fixed filter length L). It is therefore advisable to work with the same, optimized degree of efficiency at each cell. This can be done by waiting for another block before entering the next cell, so that each cell has the same input block length B and FFT length N (not only cells have the same structure of computations, but they also are exactly identical!). The resulting total complexity of the DWT is, as shown in section II.A, $2(1\text{-}2^{-J})$ times (29).

Table I and II show that this FFT-based DWT computation is efficient only for medium and large filter lengths (L≥16) compared to the direct method (23), if we choose the criterion to be the total number of operations (multiplications + additions) needed by the algorithm. With today's technology, this criterion is generally more useful than the sole number of multiplications [23], at least for implementations on general purpose computers. A more precise comparison with (23) can be done for large filter lengths by minimizing the criterion (mults+adds) of (29)

$$C(N)=\frac{(4\log_2 N-1)N+12}{N-(L/2-1)}$$

with respect to N. The minimal value of $C(N)$ is obtained for $N = N^*$ such that the first derivative of $C(N)$ vanishes. One has

$$C(N^*)=\min_N C(N) = 4 \log_2 N^* + (4/\ln 2 -1),$$

where N^* satisfies $N^* = (L/2-1)(\ln N^* + 1 - \ln 2/4) + 3\ln 2$. For large filter lengths L this gives $\ln N^* = \ln L + O(\ln \ln L)$, hence

$$\min_N C(N) = 4 \log_2 L + O(\log \log L).$$

This is to be compared with (23), for which the total number of operations (mults+adds) is 2L-1. The FFT-based DWT algorithm therefore a significantly improves the direct method (23) for large lengths L, the gain being about $L/(2\log_2 L)$.

C. A Generalization: Vetterli's Algorithm.

The FFT-based DWT algorithm presented in the preceding section can be generalized by gathering J_0 consecutive stages, using a method due to Vetterli (initially in the filter-bank context [34], and then applied to the Wavelet Transform computation [32]). The idea is to avoid subsequent IFFT's and FFT's by performing the subsampling operation in the frequency domain. This is done by inverting the last stage of a decimation-in-time radix-2 FFT Algorithm [32], [34]. The FFT length is then necessarily halved at each DWT stage, whereas the filter lengths remain constant = $L/2$. Therefore, theses schemes have two disadvantages. First, the structure of computations is less regular than for the simple FFT algorithm of the preceding section because FFT's have different lengths. Secondly, the relative efficiency of an FFT scheme per computed point decreases at each stage. The difficulties brought by this method are easily understood even by evaluating its arithmetic complexity... One finds, assuming the DWT is computed on a multiple of J_0 octaves, an average of

$$\frac{2^{J_0-1}}{2^{J_0}-1}\,\frac{2^n(2n+5-10.2^{-J_0})+2(J_0+3)}{2^{n+1}-(2^{J_0}-1)(L-2)} \text{ mults/point/cell}$$

$$\frac{2^{J_0-1}}{2^{J_0}-1}\,\frac{2^n(6n+5-14.2^{-J_0})+4(J_0+3)}{2^{n+1}-(2^{J_0}-1)(L-2)} \text{ adds/point/cell} \qquad (30)$$

per elementary cell. This complexity was derived such that the total complexity of the DWT algorithm is exactly $2(1\text{-}2^{-J})$ times the average complexity per cell (30), so as to permit a precise comparison with (29). Note that (30) reduces to (29) when $J_0 = 1$. Table I shows the resulting complexities for $J_0 = 2, 3, 4$, when minimized against $N=2^n$. Vetterli's algorithms are efficient only when the filter lengths are large ($L\geq32$) and efficiency is lost in any practical case whenever J_0 is greater than 3.

D. Short-Length WST/IWST Algorithms.

For short filter lengths L, the complexity of the DWT can be significantly reduced compared to direct or FFT-based algorithms, by applying short-length "fast running FIR" algorithms derived in [23], [33]. The class of "fast running FIR algorithms" is interesting because the multiply/accumulate structure of computations is partially retained. These algorithms are in fact very easily implemented [23], [33].

A global description of these short-length algorithms, applied to the computation of one filter of length l is as follows (see [23] for more details). The involved sequences (input, output, and filters) are divided into sub-sequences, decimated with some integer ratio R. Assuming the filter length l is a multiple of R, filtering is done in three steps:

1. The input is decimated and the resulting R sequences are somehow combined, with A_i additions per point to provide M subsampled sequences.

2. These sequences serve as inputs to M decimated *sub-filters* of length l/R.

3. The outputs are recombined, involving A_o additions per point, to give the exact decimated filter outputs.

Fig. 6 provides an example for R=2, A_i=2, M=3. In this paper we also make use of other algorithms derived in [23], with R=3, A_i=4, M=6, A_o=6, and with R=5, A_i=14, M=12, A_o=26.

Further application of the initial algorithm is feasible, since the sub-filters are still amenable to further decomposition. For example, to implement a filter of length 15, one may use either a fast running FIR algorithm with R=3, or with R=5. Another solution is to decompose this filter by a 3x5 algorithm which performs one elementary fast running FIR algorithm with R=3, the subfilters (of length 5) being themselves decomposed using fast running FIR algorithms corresponding to R=5. Similarly, the initial filter can be also decomposed by a 5x3 algorithm. All the above algorithms yield to different complexities, which are treated with some detail in [23].

In this paper we restrict ourselves to at most two successive application of fast running FIR algorithms (as in the above example), in order to retain as much as possible the simplicity of these schemes, even at the cost of a slight loss of efficiency.

To derive a short-length DWT algorithm, we apply fast running FIR algorithms to the four filters of length $l = L/2$ in the elementary cell of the DWT (Fig. 4). Here since two pairs of filters share the same input, all pre-additions (A_i) can be combined together on a single input.

Table II lists the resulting complexities, using the decomposition of fast running FIR algorithms that minimizes the total number of operations (multiplications + additions). When two different decompositions yield the same total number of operations, we have chosen the one that minimizes the number of multiplications. Another choice would have been to minimize the number of multiplication-accumulations. Table II shows that short-length algorithms are more efficient than the FFT-based algorithms for lengths up to L=18. Since DWT's are generally computed with short filter lengths to maintain the complexity at a reasonable level [1], short-length algorithms give the best practical alternative we could find. Compared to the straightforward filter bank implementation, these schemes provide noticeable savings: for example, a DWT cell with filter length L=18 requires a total of 25 operations per output point instead of 35 for the filter-bank scheme (23).

III. Implementing Continuous Wavelet Transforms

A. The Continuous Wavelet Transform (CWT)

In the previous section, we derived a filter bank implementation and various fast algorithms for the computation of Wavelet Series coefficients. The same ideas and techniques can be readily used to implement the Continuous Wavelet Transform (CWT) [10], [12], [13], [18], [29], [36], defined as

$$\mathrm{CWT}\{x(t);\, a, b\} = \int x(t)\, \tfrac{1}{\sqrt{|a|}}\, \psi^*\!\left(\tfrac{t-b}{a}\right) dt. \tag{31}$$

In contrast with the WST (4) for which time-scale parameters were discretized according to $a=2^j$, $b=k2^j$ (j, $k\in\mathbb{Z}$), here a, $b\in\mathbb{R}$ vary continuously. Roughly speaking, the basic wavelet prototype $\psi(t)$ can be any regular, band-pass function of t, and it is generally assumed $\int \psi(t)\,dt=0$ for reconstruction purposes [12], [13], [36]. Note that a can clearly be restricted to positives values $a>0$ when both signal $x(t)$ and wavelet $\psi(t)$ are either both real-valued or both complex analytic (i.e., their Fourier Transforms vanish for negative frequencies). As a two-dimensional time-scale representation of analog signals $x(t)$, the CWT has many applications in Signal Analysis [10], [12], [18], [29], [36].

Inverse CWT. Since the CWT brings a lot of redundancy into the representation of the signal (we represent a 1-D signal into a 2-D plane), there are several possibilities to reconstruct the signal $x(t)$ from its CWT coefficients (31):

-One can use the classical, but computationally expensive inversion formula [12], [13]

$$x(t) = c \iint \mathrm{CWT}\{x(t);\, a, b\}\, \tfrac{1}{\sqrt{a}}\psi\!\left(\tfrac{t-b}{a}\right) \tfrac{da\,db}{a^2},$$

where c is a constant depending only on $\psi(t)$.

-More efficient is to use an Inverse WST on the coefficients $\mathrm{WST}\{x(t);\, 2^j, k2^j\} = \mathrm{CWT}\{x(t);\, 2^j, k2^j\}$, when $\psi(t)$ is carefully chosen [5].

-Still another way is to use Morlet's formula [12], [13], [18]

$$x(b) = c' \int \mathrm{CWT}\{x(t);\, a, b\}\, \tfrac{da}{a^{3/2}},$$

which requires a single integration. This computation is performed from available CWT coefficients, which should be known for enough values of a in order to give an accurate reconstruction.

The above reconstruction methods are straightforward to implement. In the sequel we focus on the computation of CWT coefficients for signal analysis purposes.

B. DWT Computation of the CWT.

The aim of this section is to use the DWT (7) as an intermediate step to compute the CWT (31) on a fine, regular sampling grid in the time-scale plane (b, a), i.e.,

$$a = a_0^j$$
$$b = kT \tag{32}$$

where $a_0 > 1$ is reasonably close to 1, and where T is the sampling period of the discretized signal (we assume T=1 in the following). With discretization (32), one obtains a large number of coefficients, and therefore a nearly continuous representation in the time-scale plane for analysis purposes.

Note that the WST computation we studied in section I and II is nothing but part of the computation required here:

$$\mathrm{CWT}\{x(t);\, 2^j, k2^j\} = \mathrm{WST}\{x(t);\, 2^j, k2^j\} \tag{33}$$

We shall therefore heavily use the results of section I.*C* (WST computation) for implementing the CWT. In particular, we always assume, from now on, the basic assumptions (12), (13). We also assume that the input signal has been pre-filtered in a suitable manner, as explained in section I.*C*. These assumptions ensure that the equivalence between discrete computations and continuous ones hold with sufficient accuracy.

From (33) it is likely to obtain a CWT algorithm by combining several WST computations in a suitable manner. Nevertheless, the algorithm thus obtained would not be the most efficient one. To understand the differences between the WST and the CWT, let us first concentrate on an octave by octave computation, as was done in the WST case.

Octave by octave computation of the CWT ($a=2^j$). Consider the WST computation *via* a filter bank scheme depicted in Fig. 1. More specifically consider the computation performed at the first octave $j=1$. Half the wavelet coefficients required for the CWT at this octave are computed: the missing ones are the outputs of $H(z)$ that are discarded by the decimation process in Fig. 1(a). It is therefore sufficient to remove the subsampling on $H(z)$ to obtain the required wavelet coefficients of the first octave.

The filter $G(z)$ performs the down-scaling part: as explained above (section II.*A*), the output fed into the next cell corresponds to a down-scaled signal, which is used to compute the wavelet coefficients for $j=2$ with even time-shift parameters. The sequence that allows to obtain the coefficients with *odd* time-shift parameters is nothing but the discarded subsampled sequence.

The organization of the resulting (octave-by-octave) CWT cell is provided in Fig.7.(b). All outputs of both filters have to be computed, the outputs of $G(z)$ being used to build two interleaved sequences. However the basic computational cells of the fast DWT algorithms were specifically designed for subsampled outputs. Their structure is therefore not adapted to this new situation, and the operation counts have to be re-worked. This is done in section IV.

At the next octave $j=2$, each of the outputs corresponding to a down-scaled signal has to be processed by the *same* basic cell: the "even output" provides the same points as in the WST computation (round dots in Fig. 7(a)), while the "odd output" allows to start a new computation of the same type, shifted in time, and beginning at the next scale (e.g. squared dots in Fig.7(a)). The whole process is iterated as shown in Fig.7(a).

C. Finer Sampling in Scale.

For Signal Analysis purposes an octave by octave computation of the CWT is generally not enough. It is desirable to obtain more wavelet coefficients, with finer sampling in the scale parameter a, namely $a=2^{j/M}$, where M is the number of "voices per octave" [5], [13],[18].

To do that we apply the same trick as in section II.*D*. An "M voices per octave" CWT computation results from M successive applications of the octave-by-octave algorithm, each correponding to a different basic wavelet prototype:

$$2^{-m/M}\,\psi(2^{-m/M}\,t),\ m=0,\ \ldots,\ M-1. \tag{34}$$

The approximation (13) must be satisfied for each of these slightly stretched wavelet prototypes (34), and the whole algorithm requires about M times the computational load of the octave-by-octave algorithm of Fig.7.

D. Comparison with other algorithms.

The *"à trous" algorithm* of Holschneider *et al.* [9], [17], [30] appears as a special case of the octave by octave CWT filter bank implementation of Fig. 7. The authors have derived the octave-by-octave CWT algorithm using similar assumptions (12), (13), in the case where discrete signals and wavelets are obtained by *perfect sampling* of their analog counterparts.

$$x[n] = x(t{=}n)$$
$$h_j[n] = 2^{-j/2}\,\psi(2^{-j}n). \tag{35}$$

It can be shown that in this case one actually computes a sampled version of the integral defining CWT$\{x(t);\, 2^j,\, k\}$. This condition strongly restricts the choice the interpolating function $\phi(t)$ (12), since we must have $g[n] = \phi(n/2)$, which from (12) implies

$$g[2n] = \delta_{n0} \tag{36}$$

This latter condition is referred to as the "à trous" [with holes] property in [17]. The simple high order spline interpolating functions (see section I.*C*.2) are not usable under the "à trous" restriction. Some examples of $\phi(t)$ meeting (36) are described in [7], and cannot be written in closed form. Therefore the curve fitting problem (13) (the solution on which the CWT algorithm crucially depends) is more fastidious to solve in the "à trous" case.

Another CWT algorithm using DWT's has been proposed recently by Gopinath and Burrus [11], which differs notably from ours. They assume that the signal is completely determined from its WST coefficients, hence the CWT can be computed using only these coefficients by some reproducing kernel equation [11], similar to an interpolation procedure described in [16]. In this paper we "oversample" the discretization $a=2^j$, $b=k2^j$ by computing more coefficients, namely the octave-by-octave CWT coefficients, directly from the signal and not from the sole wavelet coefficients corresponding to $a=2^j$, $b=k2^j$. This should result in a faster implementation compared to [11] which uses a computationally expensive kernel expansion.

An original CWT algorithm has also been proposed by the Bertrands and Ovarlez [2], which uses the scaling property of wavelets $\psi(t) \to a^{-1/2}\,\psi(t/a)$ rather than the convolutional form of (31) (which is $x(t)$ convolved with $a^{-1/2}\,\psi(t/a)$). Let us outline the derivation of this algorithm. Write (31) in the frequency domain, assuming that the signal $x(t)$ and wavelet $\psi(t)$ are complex analytic. This gives

$$\mathrm{CWT}\{x(t);\,a,\,b\} = \int_0^{+\infty} X(f)\,e^{2i\pi fb}\sqrt{a}\,\psi^*(af)\,df \tag{37}$$

where $X(f) = \int x(t)\,e^{-2i\pi ft}\,dt$ and $\psi(f)$ are the Fourier transforms of $x(t)$ and $\psi(t)$, respectively. Then perform the change of variable $\varphi = \ln f$. A correlation form in $\alpha = \ln a$ appears in the integral.

$$\mathrm{CWT}\{x(t);\,a,\,b\} = \int_{\mathbf{R}} X(e^{\varphi})\,e^{\varphi/2}\,e^{2i\pi e^{\varphi}b}\;\psi(e^{\alpha+\varphi})\,e^{\frac{\alpha+\varphi}{2}}\,d\varphi \tag{38}$$

After suitable discretization, this correlation can be performed using FFT's. As stated in [2], the *Mellin Transform* $M_x(\beta)$ of $x(t)$ plays a central role, since it turns out to be exactly the Inverse Fourier Transform of $\sqrt{f}\,X(f)$ in the variable $\varphi = \ln f$:

$$M_x(\beta) = \int_{f>0} X(f)\,f^{-1/2+2i\pi\beta}\,df = \int e^{\varphi/2}X(e^{\varphi})\,e^{2i\pi\beta\varphi}\,d\varphi$$

As a result, the FFT's involved in the computation of (38) are "Discrete Mellin Transforms," which are defined in [2].

This algorithm requires the pre-computation of the entire Fourier Transform of $x(t)$, which makes a running implementation (in case of infinite duration signals) cumbersome. To overcome this difficulty we present a variation of the Bertrands/Ovarlez algorithm, which focuses on the time-domain rather than on the frequency domain. Assuming signals and wavelets are causal (i.e., supported by $t\geq 0$), (31) is rewritten using the change of variable $\tau = \ln t$, as a convolution in $\alpha=\ln a$.

$$\mathrm{CWT}\{x(t);\,a,\,b\} = \int e^{\tau/2}x(e^{\tau}+b)\;e^{(\tau-\alpha)/2}\,\psi^*(e^{\tau-\alpha})\,d\tau \tag{39}$$

The CWT coefficients are obtained, for a given b, by discretizing the convolution (39), resulting in a discrete filtering operation that can be implemented for running data.

Both algorithms (38), (39) have common characteristics:

Some of them can be considered as drawbacks: first, they involve a *geometric sampling* of either the input $X(f)$ (38) or $x(t)$ itself (39). Also, the approximation error made by discretizing (38) or (39) is not easily controlled. Finally, in contrast with the ocatve-by-octave CWT implementation proposed in this paper, the regular structure of time-shifts b has completely disappeared, and one has to recompute the input for each value of b. As a result, the complexity of such algorithms (about two FFT's of length $2JM$ per input point, where J is the number of octaves and M is the number of voices per octave) is found higher than the one obtained for the fast octave by octave CWT algorithms presented in the next sections.

On the other hand, for both algorithms (38), (39), the CWT coefficients are computed for all desired values of $\ln a$ at the same time (for given value of b). This is much more straightforward than in the CWT algorithm described in the preceding sections. This property makes the Bertrands/Ovarlez algorithms very useful if e.g. a "zoom", or a refinement of the wavelet analysis is desired in a short extent around some time location b.

IV. Fast Algorithms for the Continuous Wavelet Transform

A. Filter Bank Implementation of the CWT.

We have seen that when the discrete filters $g[n]$, $h[n]$ have been determined according to (12), (13), the general computation of the CWT reduces to several octave by octave CWT's, whose computational flow graph is depicted in Fig.7. This computation is obtained by combining several identical cells, as was the case for the WST. As in section II, we assume that both filters $g[n]$ and $h[n]$ are FIR filters and have same length L.

Assuming that the filters are directly implemented by an inner product, the octave-by-octave CWT algorithm requires

$$\begin{aligned}&2L \text{ mults/input point/cell}\\&2(L\text{-}1) \text{ adds/input point/cell.}\end{aligned}\qquad(40)$$

In contrast to the DWT filter bank of Fig.1, there are 2^{j-1} elementary cells at the jth octave in Fig. 7. These cells are identical but "work" at a different rate: a cell at the jth octave is fed by an input which is subsampled by 2^{j-1} compared to the original input $x(t)$. Therefore, the total complexity required by an octave-by-octave CWT algorithm on J octaves, is exactly $\sum_{j=1}^{J}\frac{2^{j-1}}{2^{j-1}}=J$ times the complexity of one cell. The complexity of any filter bank implementation of a CWT therefore grows linearly with the number of octaves. In case of a direct implementation (40), the total complexity required by a CWT on J octaves is simply

$$\begin{aligned}&2LJ \text{ mults/input point}\\&2(L\text{-}1)J \text{ adds/input point.}\end{aligned}\qquad(41)$$

This is of course a significant improvement compared to the "naive method" which would consist in directly implementing (31) without taking the dilation property of wavelets into account. (This would require a complexity exponentially increasing with J).

Since the basic cell involves only FIR filters, it is amenable to a reduction of the arithmetic complexity by the same techniques described in sections II.*A-C* for the WST. Moreover in the CWT case, for a given wavelet prototype, filter lengths are larger than in the WST case. We shall see that compared to the WST case, this increases the efficiency of the fast algorithms described below.

B. An FFT-Based CWT Algorithm.

As in section II.B, FFT's can be used to accelerate the two filter computations in the elementary cell of the octave-by-ocatve CWT. The method has been shortly explained in section II.*B*. In the CWT case, wrap-around effects are avoided if the FFT-length N is such that

$$N \geq L+B-1, \tag{42}$$

where B is the input block length, and where L is the filters' length. The block length is therefore chosen according to (42) as $B = N - (L-1)$. Each elementary cell is therefore computed by first performing an FFT of length N on the input, then performing two frequency-domain convolutions by multiplying (Hermitian symmetric) length-N FFT's of $g[n]$ and $h[n]$, and finally applying two inverse FFT's on the results. This requires

$$2\ \mathrm{FFT}_N + 2\ N/2 \text{ complex mults} + \mathrm{IFFT}_N$$

per cell (for B input points). Assuming, as in section II.B, that a complex multiplication requires three real multiplications and three real additions, and that "split-radix" FFT algorithms with complexity (28) are used for real data and filters, we end up with

$$\frac{3.2^{n-1}(n-1)+6}{2^n-L+1} \text{ mults/input point/cell}$$

$$\frac{9.2^{n-1}(n-1)+12}{2^n-L+1} \text{ adds/input point/cell} \tag{43}$$

for each elementary cell. As in the WST case, once a cell is computed, wrap-around effects are eliminated in the time-domain and one waits for one block before entering the next stage, so that each cell has the same input block length B and the same FFT length N. From the discussion of section IV.*A*, the whole CWT algorithm computed on J octaves requires J times this complexity, therefore (43) is also the total complexity per input point per octave. Hence it is also the total complexity per *output* point. Table III shows the obtained complexities, when minimized against N for different filter lengths. Since all cells are computed with FFT's of the same length N, once N is optimized for one cell, it is optimal for the whole algorithm.

Table III shows that compared to the direct implementation (40), the FFT computation becomes more efficient for $L \geq 9$, in terms of total number of operations (multiplications + additions). By deriving this number with respect to N, one finds that the optimal FFT length satisfies $N = (0.69\, n + 0.31)(L-1)$. For large L the corresponding minimized total number of operations per input point is

$$6n + 2.65 = 6 \log_2 L + O(\log \log L).$$

This is a significant improvement compared to (40), for which the total number of operations per input point is about $4L$. The gain of the FFT computation is therefore asymptotically $2L/3\log_2 L$ for large filter lengths L. This gain is asymptotically larger than in the WST case (section II.*B*).

C. A Generalization: Application of the Vetterli's Algorithm

The method shortly presented in section II.C, that Vetterli derived for DWT algorithms, can also be used for the octave-by-ocatve CWT computation. The discussion of section II.C could have been made here as well.

We provide an example, when two octaves are gathered together. Three elementary cells of Fig. 7 are therefore merged into one 1-input, 7-output cell that covers two octaves. Here the FFT length $N=2^n$ must be greater than or equal to $B + 3(L-1)$ to avoid wrap-around effects (compare with (42)). This results in an average of

$$\frac{2^{n-1}(2n-1)+6}{2^n-3L+3} \text{ mults/input point}$$

$$\frac{6.2^{n-1}(n-1)+12}{2^n-3L+3} \text{ adds/input point} \qquad (44)$$

per octave (more precisely, twice this complexity per cell). Table III shows that the resulting complexities, when minimized against N, are significantly better than (43) for large lengths only, although they slightly reduce the complexity as soon as $L \geq 8$. The price to be paid is a more involved implementation, with much larger FFT lengths.

D. Short-Length CWT Algorithms.

Fast running FIR algorithms for decimation ratios R=2, 3, 5, described in section II.*D*, can be easily applied in the CWT case. As in the WST case, in one elementary cell of Fig. 7(b), one input feeds both filters and pre-additions can therefore be combined on the single input.

Table III lists the resulting complexities, using the fast FIR decomposition (of depth at most 2) that minimizes the total number of operations (multiplications + additions). When two different decompositions yield the same number of operations, we have chosen the one that minimizes the number of multiplications.

Table III shows that short-length CWT algorithms are more efficient than the FFT-based algorithm of section IV.*B* for lengths up to 20. It even remains more efficient than the generalized algorithm of section IV.*C* (which gathers two octaves) for lengths up to 12. If the CWT is computed with medium filter lengths so as to maintain the complexity as a reasonable level, short-length algorithms may be a good trade-off both in terms of structure and complexity.

CONCLUSION

This paper has provided fast algorithms for computing various kinds of Wavelet Transforms, from the fully discrete version to the fully continuous one, and for any type of wavelet. Prefiltering the signal allows DWT schemes to be used as an intermediate computation for any type of Wavelet Transform. We have given indications on how the pre-filter can be designed, together with examples. This pre-filter is avoided in the orthonormal case.

The remainder of this paper is devoted to the derivation of fast algorithms to be applied on the prefiltered version of the signal.

The WST is computable by means of a DWT, for which two different fast algorithms have been derived: the first one is based on FFT's: it is efficient for medium to large wavelet prototypes ($L\geq 16$). The second one is based on short-length fast FIR algorithms: it is efficient for small to medium size filters. Compared to the situation encountered for fixed coefficient filtering [23], [24], these fast algorithms are useful for shorter filters, while the reduction of the arithmetic complexity, although substancial, is lower.

We have finally modified these algorithms for use in the computation of the CWT. The resulting algorithms are efficient for even shorter wavelet prototypes than in the WST case, with an improvement which is asymptotically greater.

The availability of both FFT-based and Fast-FIR-based algorithms allows computational efficiency in any case of interest.

ACKNOWLEDGEMENT

Fruitful, passionate discussions with M. Vetterli are greatfully acknowledged.

REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Vector Quantization in the Wavelet Transform Domain," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, Apr.3-6, 1990, pp. 2297-2300.

[2] J. Bertrand, P. Bertrand, and J.P. Ovarlez, "Discrete Mellin Transform for Signal Analysis," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, Apr.3-6, pp. 1603-1606, 1990.

[3] A. Cohen, I. Daubechies and J.C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets," to appear.

[4] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. in Pure and Applied Math.*, Vol.41, No.7, pp.909-996, 1988.

[5] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. on Info. Theory*, Vol. 36, No. 5, pp. 961-1005, Sept. 1990.

[6] I. Daubechies and J.C. Lagarias, "Two-Scale Difference Equations I.Existence and Global Regularity of Solutions" to appear, *SIAM J.Math. Analysis*, 1990.

[7] G. Deslauriers and S. Dubuc, "Symmetric Iterative Interpolation Processes," *Constructive Approximation*, Vol. 5, pp. 49-68, 1989.

[8] P. Duhamel, "Implementation of Split-Radix FFT Algorithms for Complex, Real, and Real-Symmetric Data," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-34, No. 2, pp. 285-295, April 1986.

[9] P. Dutilleux, "An Implementation of the 'Algorithme à Trous' to Compute the Wavelet Transform," in [36], pp. 298-304, 1989.

[10] P. Flandrin, "Some Aspects of Non-Stationary Signal Processing with Emphasis on Time-Frequency and Time-Scale Methods," in [36], pp.68-98, 1989.

[11] R.A. Gopinath and C.S. Burrus, "Efficient Computation of the Wavelet Transforms," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, pp. 1599-1601, Apr.3-6, 1990.

[12] P. Goupillaud, A. Grossmann and J. Morlet, "Cycle-Octave and Related Transforms in Seismic Signal Analysis," *Geoexploration*, Vol.23, pp.85-102, Elsevier Science Publishers, B.V. Amsterdam, Netherlands, 1984/85.

[13] A. Grossmann and R. Kronland-Martinet, "Time and Scale Representations Obtained Through Continuous Wavelet Transforms," in *Proc. Int. Conf. EUSIPCO'88, Signal Processing IV: Theories and Applications*, J.L. Lacoume *et al.* eds., Elsevier Science Publishers, pp. 475-482, 1988.

[14] C.E. Heil and D.F. Walnut, "Continuous and Discrete Wavelet Transforms," *SIAM Review*, Vol. 31, No. 4, pp 628-666, Dec. 1989.

[15] C.E. Heil, "Wavelets and Frames," in *Signal Processing, Part I: Signal Processing Theory*, L. Auslander, T. Kailath, S. Mitter eds., Institute for Mathematics and its Applications, Vol. 22, Springer Verlag, New York, 1990.

[16] A. Grossmann, J. Morlet and T. Paul, "Transforms Associated to Square Integrable Group Representations. I. General Results," *J. Math. Phys.*, Vol. 26, No. 10, pp. 2473-2479, October 1985.

[17] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform," in [36], pp. 286-297, 1989.

[18] R. Kronland-Martinet, J. Morlet, and A. Grossmann, "Analysis of Sound Patterns Through Wavelet Transforms," *Int. J. Pattern Recognition and Artificial Intelligence*, Vol.1, No.2, pp. 273-302, pp.97-126, 1987.

[19] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Trans. on Pattern Analysis and Machine Intell*, Vol.11, No. 7, pp.674-693, July 1989.

[20] S. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. 37, No.12, pp.2091-2110, December 1989.

[21] S. Mallat and S. Zhong, "Signal Characterization from Multiscale Edges," in *Proc. 10th Int. Conf. Pattern Recognition, Pattern Recognition, Systems and Applications*, Los Alamitos, CA, pp.891-896, 16-21 June 1990.

[22] Y. Meyer, *Ondelettes et Opérateurs*, [in French] *Tome I*, Herrmann ed., Paris, 1990.

[23] Z.J. Mou and P. Duhamel, "Short-Length FIR Filters and Their Use in Fast Nonrecursive Filtering," *IEEE Trans. Signal Proc.*, June 1991.

[24] H.J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, Springer Verlag, Berlin, 1981.

[25] O. Rioul, "Structures and Algorithms for the Orthonormal Discrete Wavelet Transform," in *Proc. 1990 Digital Signal Processing Workshop*, New Paltz, NY, pp. 3.3.1-2, September 16-19, 1990.

[26] O. Rioul, "Fast Algorithms for the Continuous Wavelet Transform," *1991 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Toronto, Ontario, Canada, May 14-17, 1991.

[27] O. Rioul and P. Flandrin, "Time-Scale Energy Distributions: A New Class Extending Wavelet Transforms," *IEEE Acoust., Speech, Signal Proc.*, submitted May 1990, revised Nov. 1990. Rapport ICPI TS-8910, URA 346 CNRS, 25, rue du Plat, 69288 Lyon Cedex 02.

[28] O. Rioul, "A Discrete-Time Multiresolution Theory that Unififes Octave-Band Filter Banks, Pyramid and Wavelet Transforms," submitted to *IEEE Trans. Acoust., Speech, Signal Proc.*, June 1990.

[29] O. Rioul and M. Vetterli, "Wavelet Transforms in Signal Processing," submitted to *IEEE ASSP Magazine*, Fall 1991.

[30] M.J. Shensa, "Affine Wavelets: Wedding the Atrous and Mallat Algorithms," submitted to *IEEE Trans. on Acoust., Speech, and Signal Proc.*, May 1990.

[31] M.J.T. Smith and T.P. Barnwell, "Exact Reconstruction for Tree-Structured Subband Coders," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-34, pp.434-441, June 1986.

[32] M. Vetterli, *Analyse, Synthèse et Complexité de Calcul de Bancs de Filtres Numériques*, Thesis [in French], Thèse No. 617, Ecole Polytechnique Fédérale de Lausanne, 1986.

[33] M. Vetterli, "Running FIR and IIR Filtering Using Multirate Filter Banks," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-36, No. 5, pp. 730-738, May 1988.

[34] M. Vetterli, C. Herley "Wavelets and Filter Banks: Relationships and New Results," in *Proc. 1990 IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, pp. 1723-1726, Apr. 3-6, 1990.

[35] M. Vetterli and C. Herley "Wavelets and Filter Banks: Theory and Design," submitted to *IEEE Trans. Acoust., Speech, Signal Proc.*, Aug 1990.

[36] *Wavelets, Time-Frequency Methods and Phase Space*, Proc. Int. Conf. Marseille, France, Dec. 14-18, 1987, J.M. Combes, A. Grossmann, Ph. Tchamitchian eds., Inverse Problems and Theoretical Imaging, Springer Verlag Berlin Heidelberg, 315 pp., 1989.

FIGURE CAPTIONS

Fig. 1. Basic computational cell of the DWT (a) and the IDWT (b). The organization of cells shown in (c) provides the wavelet coefficients corresponding to a dyadic grid in the time-scale plane (d). The signal may be reconstructed using the scheme (e).

Fig. 2. A "three voices per octave" WST output in the time-scale plane (b,a), showing the imbrication of the computation. Points labelled by circles, squares and crosses are computed separately by a DWT-based algorithm.

Fig. 3. A full analysis/synthesis WST scheme. Under suitable assumptions on $x(t)$, this scheme may have exact reconstruction, i.e. $y(t) = x(t)$. (see text).

Fig. 4. A rearrangement of the DWT cell of Fig. 1(a) which avoids subsampling, thereby allowing the application of fast algorithms.

Fig. 5. FFT-based implementation of the DWT cell of Fig. 4. The overlap-add (or overlap-save) procedure is not explicitly shown.

Fig. 6. A simple example of fast FIR algorithm with decimation ratio R=2. Subscripts 0 and 1 indicate biphase decomposition (25).

Fig. 7. Basic computational cell (a) for computing ocatve by octave a CWT. (b) shows the connection of the cells and the corresponding position of the outputs in the time-scale plane.

TABLE CAPTIONS

TABLE. I. ARITHMETIC COMPLEXITY PER OUTPUT (OR INPUT) POINT PER OCTAVE OF FFT-BASED WST ALGORITHMS. Each entry gives the number of operations in the form *mults+adds*, and the corresponding initial FFT length.

TABLE. II. ARITHMETIC COMPLEXITY PER OUTPUT (OR INPUT) POINT PER OCTAVE OF VARIOUS WST ALGORITHMS. Each entry gives the number of operations in the form *mults+adds*, and either the FFT size or the type of fast running FIR algorithm used (see text).

TABLE. III. ARITHMETIC COMPLEXITY PER INPUT POINT PER OCTAVE (OR: PER COMPUTED POINT) OF VARIOUS CWT ALGORITHMS. Each entry gives the number of operations in the form *mults+adds*, and either the FFT size or the type of fast running FIR algorithm used (see text).

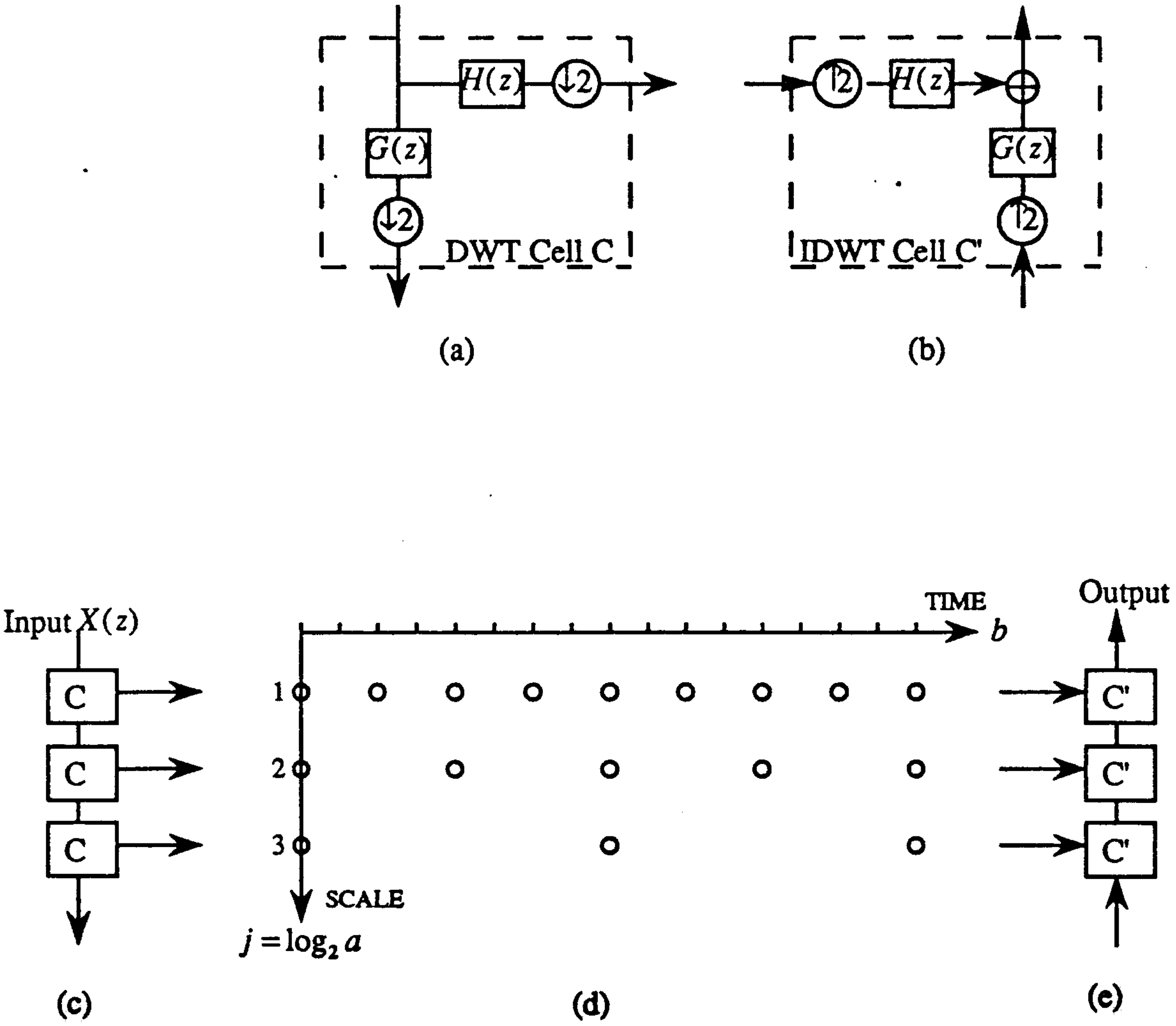

H(z)
↓2
G(z)
↓2
DWT Cell C
↑2
H(z)
G(z)
↑2
IDWT Cell C'
(a)
(b)
Input X(z)
C
C
C
TIME
b
1
2
3
SCALE
j = log₂ a
Output
C'
C'
C'
(c)
(d)
(e)

Fig. 1

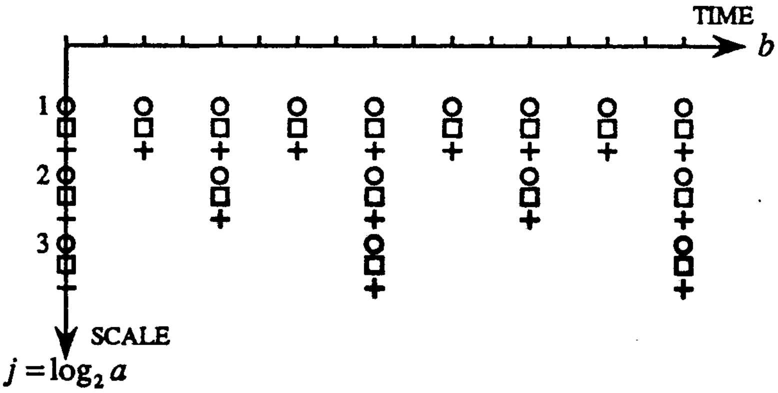


Fig. 2

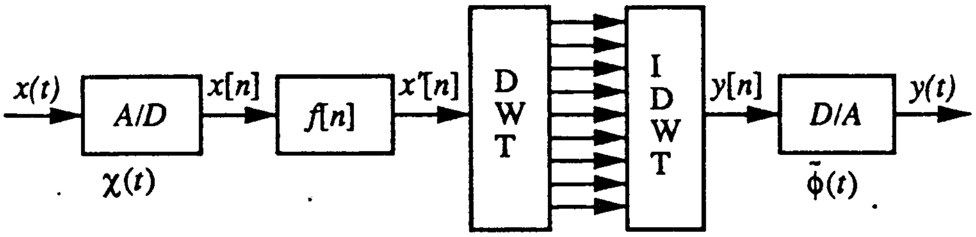


Fig. 3

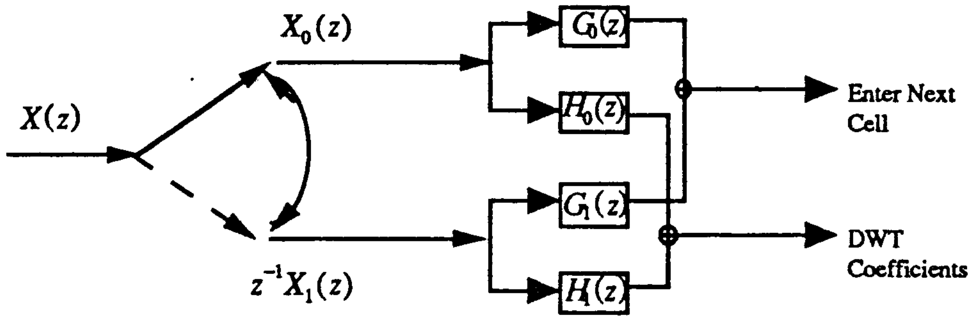

$X_0(z)$
$X(z)$
$z^{-1}X_1(z)$
$G_0(z)$
$H_0(z)$
$G_1(z)$
$H_1(z)$
Enter Next Cell
DWT Coefficients

Fig. 4

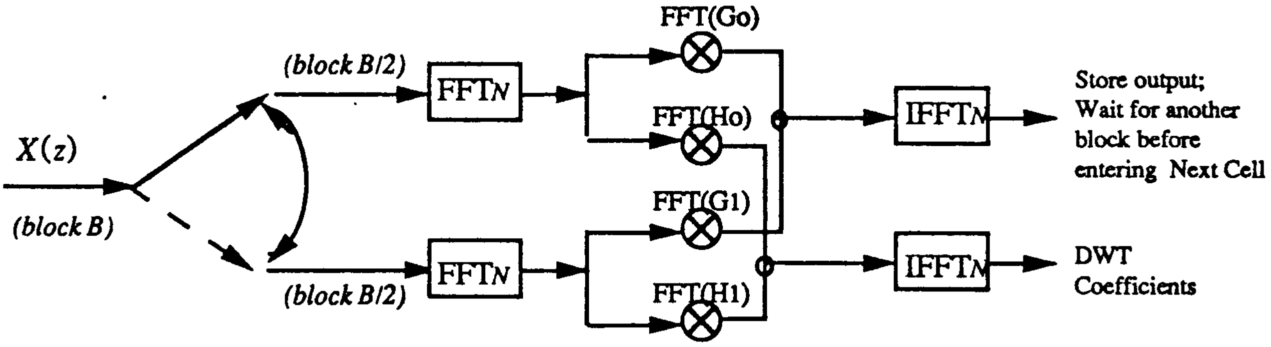

X(z)
(block B)
(block B/2)
(block B/2)
FFTN
FFTN
FFT(Go)
FFT(Ho)
FFT(G1)
FFT(H1)
IFFTN
IFFTN
Store output;
Wait for another
block before
entering Next Cell
DWT
Coefficients

Fig. 5

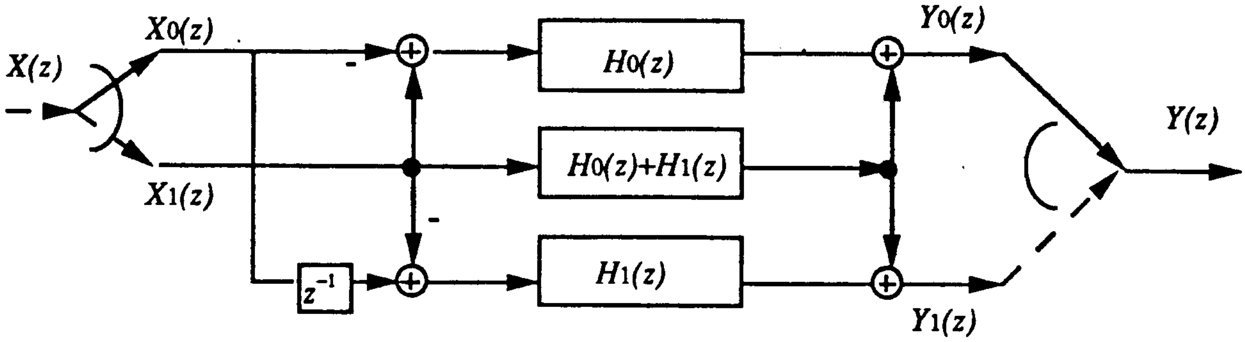


Fig. 6

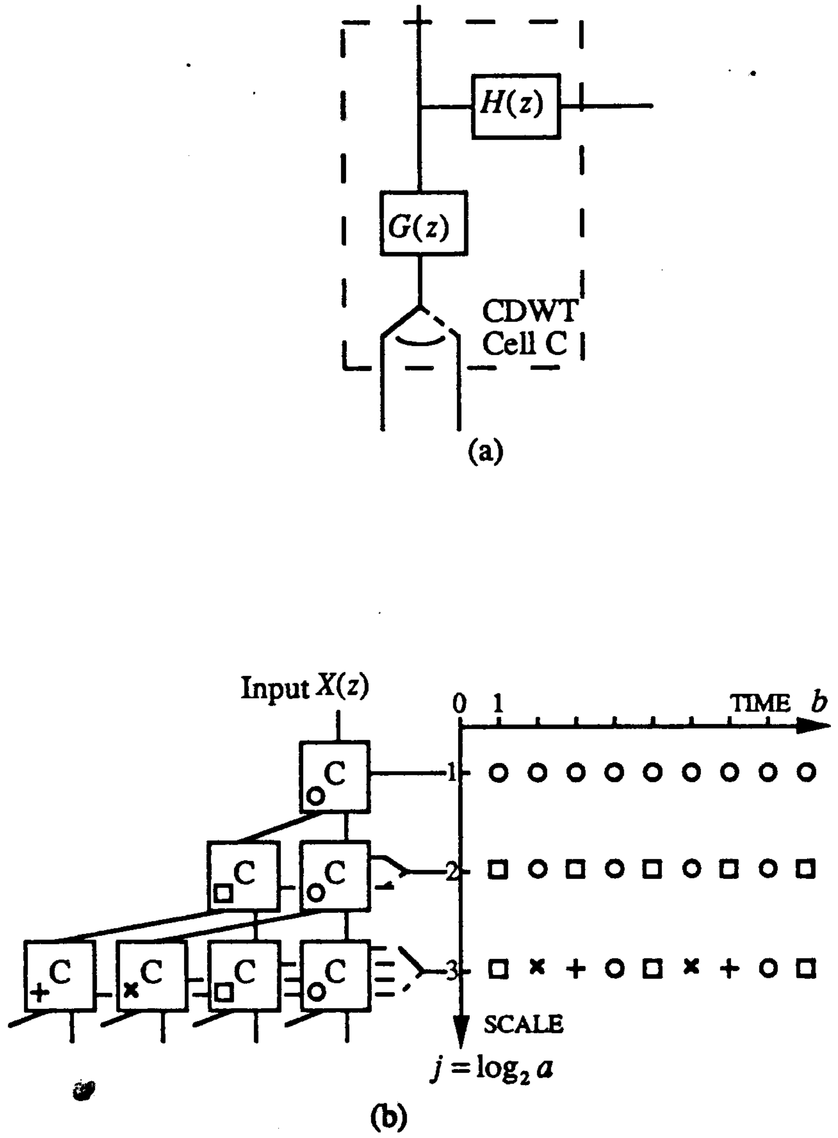


Fig. 7

Filter Length L	Straightforward Filter Bank (section II.A)	FFT-Based Algorithm (section II.B)	Vetterli's Alg. (2 octaves merged) (section II.C)	Vetterli's Alg. (3 octaves merged) (section II.C)	Vetterli's Alg. (4 octaves merged) (section II.C)
2	2 + 1	3 + 6 (2)	3.17 + 5.83 (2)	3.07 + 6.07 (4)	3.17 + 6.17 (4)
4	4 + 3	4 + 9.33 (4)	4.56 + 10.97 (16)	5.17 + 12.43 (32)	5.58 + 14.00 (128)
8	8 + 7	5.23 + 14.15 (16)	5.68 + 14.67 (64)	6.10 + 15.53 (128)	6.61 + 16.90 (256)
16	16 + 15	6.56 + 18.24 (32)	6.61 + 17.41 (128)	6.88 + 18.10 (512)	7.25 + 19.06 (1024)
32	32 + 31	7.92 + 22.37 (64)	7.50 + 20.05 (256)	7.56 + 20.14 (1024)	7.90 + 21.01 (2048)
64	64 + 63	9.12 + 26.20 (256)	8.25 + 22.55 (1024)	8.23 + 22.13 (2048)	8.54 + 22.90 (4096)
128	128 + 127	10.27 + 29.67 (512)	9 + 24.79 (2048)	8.89 + 24.10 (4096)	9.16 + 24.76 (8192)

TABLE I

Filter Length *L*	Straightforward Filter Bank (section II.A)	FFT-Based Algorithm (section II.B)	Short-Length Algorithm (section II.C)
4	4 + 3	4 + 9.33 (4)	3 + 4 (2)
6	6 + 5	4.67 + 12 (8)	4 + 6.3 (3)
8	8 + 7	5.23 + 14.15 (16)	4.5 + 8.5 (2x2)
10	10 + 9	5.67 + 15.33 (16)	4.8 + 14.2 (5)
12	12 + 11	6.18 + 16.73 (16)	6 + 12 (2x3)
16	16 + 15	6.56 + 18.24 (32)	9 + 13 (2x2)
18	18 + 17	6.83 + 19 (32)	8 + 17 (3x3)
20	20 + 19	7.13 + 19.83 (32)	7.2 + 21.4 (5x2)
24	24 + 23	7.32 + 20.68 (64)	12 + 18 (2x3)
30	30 + 29	7.76 + 21.92 (64)	9.6 + 27 (5x3)
32	32 + 31	7.92 + 22.37 (64)	18 + 22 (2x2)

TABLE II.

Filter length *L*	Straightforward Filter Bank (section IV.A)	FFT-Based Algorithm (section IV.B)	FFT-Based (two octaves merged) (section IV.C)	Short-Length Algorithm (section IV.D)
2	4 + 2	4 + 10 (4)	4.8 + 12 (16)	3 + 3 (2)
3	6 + 4	5 + 14 (8)	5.8 + 15.2 (32)	4 + 5.3 (3)
4	8 + 6	6 + 16.8 (8)	6.5 + 17.2 (32)	4.5 + 7.5 (2x2)
5	10 + 8	6.5 + 19 (16)	6.9 + 18.7 (64)	4.8 + 13.2 (5)
6	12 + 10	7.1 + 20.7 (16)	7.3 + 19.8 (64)	6 + 11 (2x3)
8	16 + 14	7.9 + 23.5 (32)	7.8 + 21.6 (128)	9 + 12 (2x2)
9	18 + 16	8.2 + 24.5 (32)	8.1 + 22.3 (128)	8 + 16 (3x3)
10	20 + 18	8.6 + 25.6 (32)	8.3 + 22.9 (128)	7.2 + 20.4 (5x2)
12	24 + 22	9.2 + 27.4 (64)	8.6 + 24.2 (256)	12 + 17 (2x3)
15	30 + 28	9.7 + 29 (64)	9 + 25.2 (256)	9.6 + 26 (5x3)
16	32 + 30	9.9 + 29.6 (64)	9.1 + 25.5 (256)	18 + 21 (2x2)
18	36 + 34	10.3 + 30.9 (64)	9.4 + 26.3 (256)	16 + 24 (3x3)
20	40 + 38	10.6 + 31.8 (128)	9.6 + 27 (512)	14.4 + 27.6 (5x2)
24	48 + 46	11 + 33 (128)	9.8 + 27.8 (512)	24 + 29 (2x3)
25	50 + 48	11.1 + 33.3 (128)	9.9 + 27.9 (512)	11.5 + 44.9 (5x5)
27	54 + 52	11.3 + 34 (128)	10 + 28.3 (512)	24 + 32 (3x3)
30	60 + 58	11.7 + 35 (128)	10.2+ 28.9 (512)	19.2 + 35.6 (5x3)
32	64 + 62	11.9 + 35.7 (128)	10.4+ 29.4 (512)	36 + 39 (2x2)
64	128 + 126	13.7 + 41.1 (512)	11.6+ 33.1 (2048)	72 + 75 (2x2)
128	256 + 254	15.4 + 46.2 (1024)	12.7+ 36.4 (4096)	144 + 147 (2x2)

TABLE III.