The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms

Mark J. Shensa

Abstract—**In a general sense this paper represents an effort to clarify the relationship of discrete and continuous wavelet transforms. More narrowly, it focuses on bringing together two separately motivated implementations of the wavelet transform, the *algorithme à trous* and Mallat's multiresolution decomposition. It is observed that these algorithms are both special cases of a single filter bank structure, the discrete wavelet transform, the behavior of which is governed by one's choice of filters. In fact, the à trous algorithm, originally devised as a computationally efficient implementation, is more properly viewed as a nonorthonormal multiresolution algorithm for which the discrete wavelet transform is exact. Moreover, it is shown that the commonly used Lagrange à trous filters are in one-to-one correspondence with the convolutional squares of the Daubechies filters for orthonormal wavelets of compact support.**

A systematic framework for the discrete wavelet transform is provided, and conditions are derived under which it computes the continuous wavelet transform exactly. Suitable filter constraints for finite energy and boundedness of the discrete transform are also derived. Finally, relevant signal processing parameters are examined, and it is remarked that orthonormality is balanced by restrictions on resolution.

I. Introduction

Wavelets are rapidly finding application as a tool for the analysis of nonstationary signals [1]–[5]. However, with the notable exception of orthonormal wavelets [6]–[9], very little literature has been devoted to linking discrete implementations to the continuous transform. As in the case of the discrete Fourier transform, these implementations (or filter banks) can, and should, stand on their own as abstract decompositions of discrete time series. Their wide sweeping significance, however, lies in their interpretation as wavelet transforms. In a general sense, this paper represents an effort to clarify the relationship of discrete and continuous wavelet transforms. More narrowly, it focuses on bringing together two separately motivated implementations of the wavelet transform. One of them, the *algorithme à trous*[1] for nonorthogonal wavelets [4], [5], was developed for music synthesis [2] and is particularly suitable for signal processing. The other, the multiresolution approach of Mallat and Meyer, originally used in image processing, employs orthonormal wavelets [6]–[10]. The latter algorithm, apart from its wavelet interpretation, was discovered previously in the form of quadrature mirror filter (QMF) filter banks with perfect reconstruction where it finds application in speech transmission and split-band coding [11]–[13].

A glance at these two algorithms suffices to reveal closely related structures. In fact, apart from the constraints on their filters, the decimated à trous [5] and Mallat algorithms are identical. We are thus led to examine the expanded family of algorithms encompassing both types of filters. In this vein, it is shown that the Lagrange interpolation filters commonly employed by the à trous algorithm are actually the squares (in a convolutional sense) of the Daubechies filters for compact orthonormal wavelets. We also derive conditions under which the discrete implementation computes a continuous wavelet transform exactly and find that they bear an intimate relationship to the à trous constraints.

From a more general viewpoint, the situation is as follows: The algorithms to be discussed all are filter bank structures (see Fig. 1). Their only distinguishing feature is the choice of two finite length filters, a low-pass filter f and a bandpass filter g. The low-pass condition, expressed more precisely as $\Sigma f_k = \sqrt{2}$, is necessary to the construction of a corresponding continuous wavelet function. The bandpass requirement, while apparently not essential to all applications, ensures that finite energy signals lead to finite energy transforms (see Section VI). Under these conditions, the filter bank output will be referred to as the discrete wavelet transform (DWT), a terminology which will become clear in the course of the paper.

One class of DWT filter pairs are the Daubechies filters [8] which yield orthogonal wavelet decompositions and constitute, in more conventional terms, a QMF filter bank with perfect reconstruction. Another is that for which the low-pass filter satisfies the à trous condition $f_{2k} = \delta(k)/\sqrt{2}$. Such filters, which simply serve to interpolate every other point, correspond to a nonorthogonal wavelet decomposition. As mentioned above, if they are further restricted to be Lagrangian interpolators, they become the squares of the Daubechies filters, which is quite remarkable in consideration of the totally different derivations. This also implies that a maximally flat filter with the same

Manuscript received May 21, 1990; revised August 5, 1991. This work was supported in part by the ONR and NOSC IR program.

The author is with the Naval Ocean Systems Center, San Diego, CA 92152-5000.

IEEE Log Number 9202367.

[1] Literally, "algorithm with holes," this terminology, taken from [5], refers to the fact that all the even coefficients of the relevant filter (with the exception of the center) are zero.

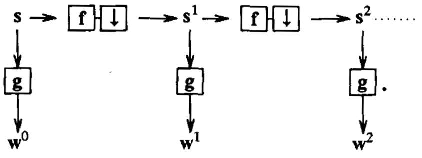

Fig. 1. A wavelet filter bank structure. The down-arrow indicates decimation. The output of the transform is the family of signals w^i, forming the two parameter transform w_n^i in the scale-time plane. Following terminology to be introduced, w^i is the (decimated) discrete wavelet transform.

number of vanishing derivatives at 0 and π is a Lagrangian interpolator; a fact which may aid in the design of maximally flat filters [14].

A fundamental question is, when do these discrete implementations yield exact (i.e., sampled) versions of a continuous wavelet transform? Aside from regularity conditions relating to smoothness [8], we find that if f is à trous, then the DWT coincides with a continuous wavelet transform by a wavelet $\psi(t)$ whose samples $\psi(n)$ form the filter g (i.e., $g_n = \psi(n)$). Even if f is not à trous, the algorithm is exact provided the signal lies in an appropriate subspace; however, in that instance, the sampled wavelet values depend on f as well as g. This is the situation in the orthonormal case where, moreover, the filter g is almost completely determined from f through the constraints of orthogonality.

In the remainder of this introduction we define, and briefly motivate, wavelet transforms at various levels of discretization. Section II contains an abbreviated derivation of the à trous algorithm followed by a description of the Mallat algorithm. (The uninitiated reader is particularly referred to [1], [6], [8].) In Section III we define the undecimated DWT, relate it to the decimated transform, and provide algorithms for its computation. Section IV states and proves several theorems which delineate the relationship between the DWT and the continuous wavelet transform. It may be read independently of the algorithms of Section II although the motivation for the constructions may not be clear. Section V defines the Lagrange à trous filters and proves that they are the squares of the Daubechies filters. In Section VI, we formulate the inversion problem and provide filter constraints which ensure finite energy and bounded operators. It concludes with a short examination of the tradeoffs involved in choosing the bandpass filter, emphasizing the differences of the orthonormal and nonorthogonal cases.

A. Transform Definitions

The continuous wavelet transform of a signal $s(t)$ takes the form

$$W(a, b) = \frac{1}{\sqrt{a}} \int \bar{\psi}\left(\frac{t-b}{a}\right) s(t)\, dt \qquad (1.1)$$

where ψ is the analyzing wavelet, a represents a time dilation, b a time translation, and the bar stands for complex conjugate. The normalization factor $1/\sqrt{a}$ is perhaps most effectively visualized as endowing $|W(a, b)|^2$ with units of power/hertz.[2] Certain weak "admissibility" conditions are usually required on $\psi(t)$ for it to be a candidate for an analyzing wavelet; namely, square integrability and

$$\int \frac{|\hat{\psi}(\omega)|^2}{|\omega|}\, d\omega < \infty \qquad (1.2)$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(t)$. They ensure that the transformation is a bounded invertible operator in the appropriate spaces [8], [16]. If $\hat{\psi}(\omega)$ is differentiable, then it suffices that ψ be zero mean, i.e., that $\int \psi(t)\, dt = 0$, for (1.2) to be satisfied.

In the realm of signal processing, the significance of (1.1) is probably (or, at least, traditionally) best grasped by comparing it to the short-time Fourier transform:

$$F(\omega, b) = \int h(t-b) e^{j\omega t} s(t)\, dt. \qquad (1.3)$$

Thus, to obtain $F(\omega, b)$, one multiplies the signal by an appropriate window h (such as a Gaussian) centered at time b and then takes the Fourier transform. In mathematical terms, (1.3) is an expansion of the signal in terms of a family of functions $h(t - b)e^{j\omega t}$, which are generated from a single function $h(t)$ through translations b in time and translations ω in frequency. In contrast, the wavelet transform (1.1) is an expansion in functions $\psi((t-b)/a)$ generated by translations b in time and dilations a in time.[3] Thus, the continuous wavelet transform resembles a (continuous) bank of short-time Fourier transforms with a different window for each frequency. The significance of this is that, while the basis functions in (1.3) have the same time and frequency resolution (that of $h(t)$ and $\hat{h}(\omega)$) at all points of the transform plane, those of (1.1) have time resolution (that of $\psi(t/a)$) which decreases with a and frequency resolution (that of $\hat{\psi}(a\omega)$) which increases with a. This property can be a great advantage in signal processing since high frequency signal characteristics are generally highly localized in time whereas slowly varying signals require good low frequency resolution.

As originally proposed by Morlet *et al.* [17], ψ was a modulated Gaussian

$$\psi(t) = e^{j\nu_0 t} e^{-t^2/2} \qquad (1.4)$$

and this function is still the prototypical analyzing wavelet for signal processing applications [1]. The window function $\psi(t/a)$ has Fourier transform $\hat{\psi}(a\omega) = ae^{-(\omega - [\nu_0/a])^2 a^2/2}$, which has analysis frequency ν_0/a. We emphasize that ν_0 is simply a parameter which determines the analyzing wavelet; its role should not be confused with that of a even though the scale axis is often expressed in terms of frequency under the transformation $a \rightarrow \nu_0/a$. Observe that (1.4) only satisfies the admissibility condition (1.2) approximately (cf. [16], [18]). As expected, its

[2] The energy density is given by $|W|^2 [(da\, db)/a^2]$, an expression which is intimately linked to representations of the affine group (see [3], [15], [16]). Since a is proportional to bandwidth, $|W|^2$ is power/hertz. See Section VI for a discussion of the discrete case.

[3] Alternatively, dilations in time may be considered contractions in frequency since the Fourier transform of $\psi(t/a)$ is $a\hat{\psi}(a\omega)$.

bandwidth is proportional to $1/a$, thus giving rise to a constant relative bandwidth; i.e., $BW/(\nu_0/a)$ = constant. This feature is also reflected in the narrowing of the time window at higher frequencies; i.e., at smaller a. In general, one's choice of the function $\psi(t)$ is dictated by its time and frequency localization properties [3], [18].

We shall be exclusively concerned with discrete values for a and b. In particular, we assume that a has the form $a = 2^i$ where i is termed the octave of the transform. The integral (1.1) then yields a wavelet series

$$W(2^i, n) \triangleq \frac{1}{\sqrt{2^i}} \int \overline{\psi}\left(\frac{t-n}{2^i}\right) s(t)\, dt. \tag{1.5a}$$

We remark that finite energy for the wavelet transform is not at all equivalent to finite energy for the wavelet series. It depends on the sampling grid as well as the function $\psi(t)$ [3]. Thus, the admissibility condition (1.2) is not necessarily appropriate in the discrete case and shall be replaced with conditions on the relevant filters in Section VI. In addition, we shall often take b to be a multiple of a[4]

$$W(2^i, 2^i n) \triangleq \frac{1}{\sqrt{2^i}} \int \overline{\psi}\left(\frac{t}{2^i} - n\right) s(t)\, dt. \tag{1.5b}$$

A logical step in applying the theory to discrete signals is to discretize the integral in (1.5)

$$w(2^i, 2^i n) \triangleq \frac{1}{\sqrt{2^i}} \sum_k \overline{\psi}\left(\frac{k}{2^i} - n\right) s(k). \tag{1.6}$$

The sample rate has been set equal to one. As indicated by $2^i n$ on the left-hand side, (1.6), as well as (1.5b), are decimated wavelet transforms. Octave i is only output every 2^i samples. In this form the resulting algorithms will not be translation invariant ([7]). This is easily seen by substituting $s(k - r)$ for $s(k)$ which produces $w(2^i, 2^i(n - r/2^i))$, an integer translation of $w(2^i, 2^i n)$ only if r is a multiple of 2^i. However, the invariance, which is lost by decimation, is easily restored by separately filtering the even and odd sequences (see Section III) or by using an equivalent algorithm, also described in Section III. Our major reason for starting with (1.5b) rather than (1.5a) is historical. It delineates the relationship of the DWT to the QMF filter banks and (orthonormal) wavelet structures already found in the literature [6]–[9]. It also simplifies our derivation of the à trous algorithm and readily lends itself to physical interpretation (see Section VI). Note that the original Mallat algorithm [6] was decimated; à trous [4] was not.

Let g be the discrete filter obtained by truncating the sampled wavelet function; i.e., $g_n = \psi(n)$. Then, proceeding from (1.6), we shall be able to arrive at the DWT pictured in Fig. 1,

$$[s^i]_n = \sum_j f_{2n-j}[s^{i-1}]^j$$
$$[w^i]_n = \sum_j g_{n-j}[s^i]^j \tag{1.7}$$

where $[w^i]_n$ corresponds to $w(2^i, 2^i n)$ of (1.6) and s^0 is the original signal s. The mysterious appearance of the filter f in (1.7) will be unraveled in the derivation of the à trous algorithm in Section II. Finally, we shall come full circle in Section IV where we show, under quite general conditions, that given filters f and g there exists a function $\psi(t)$ with $\psi(n) = g_n^\dagger \triangleq \overline{g}_{-n}$ such that the DWT acting on the sampled signal is exactly the sampled output of the continuous wavelet transform (i.e., of the wavelet series).[5] In other words, the DWT with filter g defined by $g_n^\dagger \triangleq \psi_A(n)$, which was originally conceived as an approximation of the (continuous) WT for an arbitrary analyzing wavelet $\psi_A(t)$, is exact for another wavelet function $\psi_B(t)$ where $\psi_B(n) = g_n^\dagger$ for all n. Of course, if there is sufficient regularity, $\psi_A(t)$ and $\psi_B(t)$ will be close since they coincide on the integers up to the length of g.

Before embarking on this voyage, we summarize, and hopefully clarify, the plethora of transforms with a brief analogy to the Fourier transform, Fourier series, the discretized z transform, and the discrete Fourier transform (DFT). The Fourier transform of a continuous signal $s(t)$

$$S(\omega) \triangleq \int_{-\infty}^{\infty} e^{-j\omega t} s(t)\, dt \tag{1.8}$$

is a function of the continuous variable ω. Restricting it to a discrete (one-dimensional) grid results in the coefficients of a Fourier series

$$S(2\pi m) = \int_{-\infty}^{\infty} e^{-j2\pi m t} s(t)\, dt \tag{1.9}$$

which in turn may be computed approximately by

$$s_z(2\pi m\, \Delta t) = \sum_k e^{-j2\pi mk\Delta t} s(k\, \Delta t)\, \Delta t \tag{1.10}$$

the z transform of $s_n \triangleq s(n\,\Delta t)$ output at discrete points $e^{-j2\pi m\Delta t}$. If $s(t)$ is band limited and sampled at an appropriate rate, $\Delta t = 1/N$, then the above may be computed exactly using the DFT

$$\hat{s}_m \triangleq \frac{1}{N} \sum_1^N \exp\left(-\frac{j2\pi mk}{N}\right) s_k. \tag{1.11}$$

These correspond precisely to $W(a, b)$, $W(2^i, n)$, $w(2^i, n)$, and undecimated w_n^i. With wavelets, however, we have the additional difficulty of dealing with a whole class of functions $\psi(t)$ rather than simply $e^{j\omega t}$. Also complicating things are its two-dimensional structure and the decimated versions, which, due to their $2^i n$ dependency on i, play a distinguished role without analogy in the one-dimensional case.

[4]Physically, this reflects a need for less frequency sampling of the transform output at lower frequencies (i.e., larger scales a). Mathematically, $b = 2^i n$ has its roots in the orthonormal wavelets where it suffices for invertibility of the tranform [6]. The general case, however, is much more complex [3]. Too sparse a sampling leads to incompleteness; oversampling results in a redundant set of functions.

[5]The adjoint filter $g_k^\dagger = \overline{g}_{-k}$ is used to simplify future notation. It corresponds to the integrand $\overline{\psi}(-t) * s$ found in (1.1), (1.5), and (1.6).

II. Two Algorithms

A. Notation

Decimation, which appears as a down arrow in Fig. 1, plays a pivotal role in all DWT algorithms. However, it leads to operators which are not time invariant and present a potential source of confusion. It is thus worthwhile to first establish some formal notation.

Signals and filters in boldface type will be treated as vectors, in which case * indicates discrete convolution and yields a vector. The symbol † will be used for the Hermitian adjoint filter $[\boldsymbol{f}^\dagger]_k = \bar{f}_{-k}$. Note that this is the complex conjugate reversal and does not imply a conversion of a column vector to a row vector. The above mentioned decimation operator is represented by a matrix

$$\Lambda_{km} \triangleq \delta(2k - m) = \delta_{2k,m} \tag{2.1}$$

where δ_{km} is the Kronecker delta and $\delta(k) \triangleq \delta_{k0}$. Also of significance is the dilation operator

$$D_{km} \triangleq \delta(k - 2m) = \delta_{k,2m} \tag{2.2}$$

which dilates a vector by inserting zeros. Observe that $\boldsymbol{\Lambda}$ and $\boldsymbol{D}$ are transposes of each other, and that although they are linear, they are not time invariant; i.e., they are not functions of $k - m$.

Convolution followed by decimation becomes $[\boldsymbol{\Lambda}(\boldsymbol{f} * \boldsymbol{s})]_k = \Sigma_m \Lambda_{km}[\boldsymbol{f} * \boldsymbol{s}]_m = [\boldsymbol{f} * \boldsymbol{s}]_{2k} = \Sigma_m f_{2k-m} s_m$. However, a particularly insidious pitfall remains; namely,

$$(\boldsymbol{\Lambda f}) * \boldsymbol{s} \neq \Lambda(\boldsymbol{f} * \boldsymbol{s}). \tag{2.3}$$

This "associativity" problem may be avoided by replacing convolution by $\boldsymbol{f}$ with a matrix $\boldsymbol{F}$ defined by

$$F_{mn} \triangleq f_{m-n} \tag{2.4}$$

which shall occasionally be used in our proofs. A trivial calculation yields $\boldsymbol{\Lambda F s} = \boldsymbol{\Lambda}(\boldsymbol{f} * \boldsymbol{s})$. The symbol † will also be used for the adjoint of matrices. This is consistent with the above notation where $[\boldsymbol{F}^\dagger]_{mn} \triangleq \bar{F}_{nm} = \bar{f}_{n-m} \triangleq f^\dagger_{m-n}$.

We define the Fourier transform $\hat{s}(\omega)$ of a function $s(t)$ by (1.8) and the z transform (on the unit circle) of a discrete signal $\boldsymbol{s}$ by

$$s_z(\omega) \triangleq \sum_n s_n e^{-j\omega n}. \tag{2.5}$$

In the subsequent interplay between continuous and discrete functions one must be careful to distinguish the usage of these two transforms. Ignoring their differences can easily lead to erroneous conclusions. In particular, although the Fourier transform of $s(2t)$ is $\hat{s}(\omega/2)/2$,

$$(\boldsymbol{\Lambda f})_z(\omega) = \sum_n f_{2n} e^{-jn\omega} = \sum_{n \text{ even}} f_n e^{-jn\omega/2} \neq f_z(\omega/2). \tag{2.6}$$

Some comment concerning filter definitions is also appropriate. Usage in the literature is uniform only up to the adjoint. Also, the z transform is sometimes defined with a positive exponential which leads to similar differences in the frequency domain. In keeping with signal processing applications we have chosen (2.5) as above, consistent with the Fourier transform, and we shall define our filters so that adjoints do not appear in convolutions. This produces a minimum of adjoints and greatly simplifies the notation. Unfortunately, it also results in the definition $g^\dagger_n = \psi(n)$ and the introduction of $\boldsymbol{f}^\dagger$ as an interpolation filter whereas $\boldsymbol{g}$ and $\boldsymbol{f}$ would be more natural. Note, also, that our filters are the adjoints of the filters defined in [8], although their z transforms coincide since [8] defines the z transform with a plus sign.

B. The À Trous Algorithm

We take the discretized wavelet series (1.6) as our starting point. The difficulty in implementing (1.6) is that, even for $\psi(t)$ of finite support, as i increases, $\bar{\psi}(t)$ must be sampled at progessively more points, creating a large computational burden. The solution posed by [4] is to approximate the values at nonintegral points through interpolation via a finite filter $\boldsymbol{f}^\dagger$. The resulting recursion is highly efficient and may be implemented with the filter bank structure of Fig. 1.

The interpolation is perhaps best introduced with an example. Let $\boldsymbol{f}^\dagger$ be the filter (0.5, 1.0, 0.5). Then,

$$\sum_k f^\dagger_{n-2k}\psi(k) = \begin{cases} \psi\left(\dfrac{n}{2}\right) & n \text{ even} \\ \dfrac{1}{2}\left(\psi\left(\dfrac{n-1}{2}\right) + \psi\left(\dfrac{n+1}{2}\right)\right) & n \text{ odd} \end{cases} \tag{2.7}$$

approximates a sampling of $\psi(t/2)$. With the help of the dilation operator $\boldsymbol{D}$, this may formalized as a general procedure for dyadic interpolation. The steps are illustrated in Fig. 2. Let $\boldsymbol{g}$ be a filter defined by $g^\dagger_n \triangleq \psi(n)$; i.e.,

$$g_n = \bar{\psi}(-n). \tag{2.8}$$

First we spread $\boldsymbol{g}^\dagger$ to provide space in which to put the interpolated values. The resulting filter is $\boldsymbol{Dg}^\dagger$. Then we apply a filter $\boldsymbol{f}^\dagger$ which leaves the even points fixed and interpolates to get the odd points. This condition, that $\boldsymbol{f}$ be the identity on even points, is sufficiently important to warrant a separate defintion, which follows.

Definition 2.1: The low-pass filter $\boldsymbol{f}$ is said to be an à trous filter if it satisfies

$$f_{2k} = \delta(k)/\sqrt{2}. \tag{2.9}$$

The $\sqrt{2}$ is simply a convenient means of including the normalization factor of (1.6) in the filter. The result of the

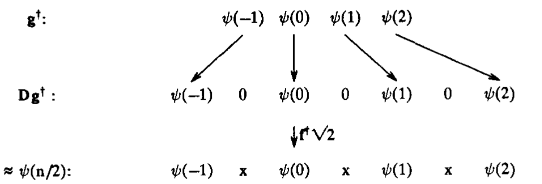

Fig. 2. Diagram illustrating the dilation and interpolation of a function $\psi(t)$: $\psi(n/2) \approx \sqrt{2} f^\dagger * (Dg^\dagger)$.

entire interpolation operation, as pictured in Fig. 2, is thus

$$[f^\dagger * (Dg^\dagger)]_n = [F^\dagger D g^\dagger]_n = \sum_k f^\dagger_{n-2k}\psi(k) \approx \frac{1}{\sqrt{2}}\psi(n/2). \quad (2.10)$$

Noting that $\psi((k/2) - n) = \psi((k - 2n)/2)$ and inserting the approximation (2.10) into (1.6), we obtain

$$w(2, 2n) \approx \sum_{k,m} f^\dagger_{k-2n-2m}\bar{g}^\dagger_m s_k = \sum_{k,m'} g_{n-m'}\bar{f}_{2m'-k}s_k = [g * (\Lambda(\bar{f} * s))]_n \quad (2.11)$$

which is simply w^i_n (1.7) with $i = 1$. Continuing inductively by replacing s in (2.11) with s^{i-1}, we find $w(2^i, 2^i n) \approx w^i_n$ for all i, where w^i_n is given by (1.7), which can be rewritten for real f

$$s^{i+1} = \Lambda(f * s^i) \quad (2.12a)$$

$$w^i = g * s^i. \quad (2.12b)$$

Except for decimation of the output (the undecimated version will be derived in Section III), this is the à trous algorithm described in [4]. Thus, we see that the à trous algorithm is simply a DWT for which the filter f (an interpolator) satisfies condition (2.9) and the filter g is obtained by sampling an *a priori* wavelet function $\psi(t)$.

Remark 1: The definition (1.6) is not so transparent as it might seem. It is, of course, intended to reflect an approximation to (1.5). From this viewpoint one might well consider a change of variables $t \to t/2^i$ before discretizing (1.5). Such a procedure certainly alleviates the computational problem since it dilates $s(t)$ (that is, samples s at 2^i, values which are known) rather than contracting $\psi(t)$. However, unless the original function $s(t)$ was highly oversampled (which begs the computational question), the approximation is poor. More precisely, to accurately approximate $s(t)$, and therefore also (1.5), we must sample at least at the nyquist rate r_{nyq} for s. Then the integral for octave i requires $\psi(t)$ to be sampled at a rate $2^i r_{\text{nyq}}$.

Remark 2: The derivation above, as well as that in [4], of the à trous algorithm make no statements regarding the accuracy of the approximation (2.11) or even of the discretization from (1.5) to (1.6). The former is iterated over i and, hence, to succeed must be numerically stable in some sense. This, in turn depends principally on choice of the filter f. A major step towards treating this question lies in the results of Section IV, as was outlined at the end of the introduction. Since the algorithm is exact for some $\psi_B(t)$, the question reduces to a) the quality of the approximation $\psi \approx \psi_B$ and b) the effect of this approximation on the wavelet integral (1.1). Inasmuch as $\psi(n) = \psi_B(n)$ for the finite set of integers n used to obtain g from $\psi(t)$, it is sufficient that these functions be smooth enough and decrease fast enough at infinity. Conditions on f which achieve regularity of the constructed wavelet ψ_B are found in [8]. Quantifying these statements remains a subject for future study.

C. Multiresolution Algorithm

Mallat's algorithm, illustrated in Fig. 3, has essentially the same tree structure as (2.12) (cf. [6]–[9]). Namely,

$$s^{i+1} = \Lambda(h * s^i) \quad (2.13a)$$

$$d^{i+1} = \Lambda(g * s^i). \quad (2.13b)$$

In keeping with the literature, we have replaced the filter f with the filter h, which also serves to indicate that this class of filters is constrained, as detailed below. We remark that none of these filters are à trous filters. The constraints on h and g which ensure an orthonormal multiresolution analysis [6]–[9] are

$$\sum_j [\bar{h}_{2j-n}h_{2j-m} + \bar{g}_{2j-n}g_{2j-m}] = \delta_{nm} \quad (2.14a)$$

$$\sum_j h_{2n-j}\bar{g}_{2m-j} = 0 \quad (2.14b)$$

$$\sum_n g_n = 0 \quad (2.14c)$$

$$\sum_n h_n = \sqrt{2}. \quad (2.14d)$$

Recalling that $H_{mn} \triangleq h_{m-n}$ and that $\Lambda^\dagger = D$, we may rewrite (2.14a) and (2.14b) as

$$(H^\dagger D)(\Lambda H) + (G^\dagger D)(\Lambda G) = I \quad (2.15)$$

$$(\Lambda H)(G^\dagger D) = 0. \quad (2.16)$$

Furthermore, (2.15) and (2.16) imply (e.g., multiply (2.15) on the left by $\Lambda H^\dagger$)

$$(\Lambda H)(H^\dagger D) = I$$
$$(\Lambda G)(G^\dagger D) = I. \quad (2.17)$$

Thus, $H^\dagger D$ and $G^\dagger D$ are injections and (2.13) is an orthogonal decomposition of the discrete signal s^i. That is, $s^{i-1} = H^\dagger D s^i + G^\dagger D w^i$ with the scalar product $(H^\dagger D s^i) \cdot (G^\dagger D w^i) = 0$. In fact, (2.15) is a paradigm for inverting the transform. These concepts are illustrated in Fig. 4.

Furthermore, from (2.14) it follows that (2.13) represents a wavelet decomposition [6], [8], [9] as follows: There exists a scaling function $\phi(t)$ whose Fourier transform is given by

$$\hat{\phi}(\omega) \triangleq \prod_{j=1}^{\infty} \left(\frac{1}{\sqrt{2}}\bar{h}_z\left(\frac{\omega}{2^j}\right)\right). \quad (2.18)$$

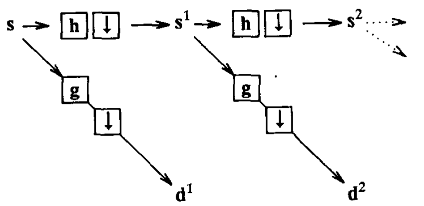

Fig. 3. The Mallat multiresolution algorithm. The down-arrow indicates decimation.

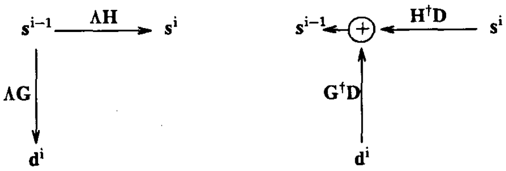

Fig. 4. Illustration of a single stage and its inverse in the multiresolution algorithm for orthonormal wavelets.

Expression (2.18) forms the basis of a recursion

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}} \bar{\boldsymbol{h}}_z \left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \tag{2.19}$$

which in the time domain takes the form

$$\phi(t) = \sum_k \bar{h}_{-k} \sqrt{2}\, \phi(2t - k). \tag{2.20}$$

With some additional computation (see Section IV), one may demonstrate that the translates and dilates of ϕ,

$$\phi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}} \phi\left(\frac{t}{2^i} - n\right) \tag{2.21}$$

have the property

$$\phi_n^{i+1}(t) = \sum_k [\Lambda H]_{nk} \phi_k^i(t). \tag{2.22}$$

Note that the above definitions differ in the sign of i from those of [8].

Finally, define

$$\psi(t) \triangleq \sum_k \sqrt{2}\, \bar{g}_{-k} \phi(2t - k). \tag{2.23}$$

Then, using (2.14) and the above properties of ϕ, one can show that the family of wavelets,

$$\psi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}} \psi\left(\frac{t}{2^i} - n\right) \tag{2.24}$$

are orthonormal ($\int \psi_n^i(t) \bar{\psi}_k^j(t)\, dt = \delta_{ij}\, \delta_{nk}$), and that the $\boldsymbol{d}^i$ are the coefficients of the expansion of $s(t)$ in terms of the ψ_n^i.[6]

More precisely, the translates and dilates $\phi_n^i(t)$ of the scaling function form a basis for $L^2(R)$, and

$$d_n^i = \frac{1}{\sqrt{2^i}} \int s(t) \bar{\psi}\left(\frac{t}{2^i} - n\right) = \int s(t) \bar{\psi}_n^i(t) \tag{2.25}$$

provided

$$s_k^0 = \int s(t) \bar{\phi}(t - k). \tag{2.26}$$

Then, if $s(t)$ is in $L^2(R)$, the expansion of $s(t)$ is

$$s(t) = \sum_{i=-\infty}^{\infty} \sum_n d_n^i \psi_n^i(t). \tag{2.27}$$

This follows from (2.25) and orthonormality since completeness of the $\phi_n^i(t)$ implies completeness of the $\psi_n^i(t)$.

Until recently, the only known orthonormal wavelets with compact support (i.e., zero outside a finite interval) were the Haar functions, generated by $\psi(t) = 1$ for $0 \le t < 1/2$; -1 for $-1/2 \le t < 0$; and 0 elsewhere. Daubechies [8], [19] has uncovered an entire family of finite length filters satisfying (2.14), demonstrating that the corresponding wavelets defined by (2.18), (2.23), and (2.24) are orthonormal as a consequence of (2.15), (2.16) and have compact support since they were generated by finite length filters (see Section IV, eq. (4.10)). The first two of these filters are

$$\boldsymbol{h}^{\dagger} = \frac{1}{\sqrt{2}} (1, 1) \tag{2.28a}$$

and

$$\boldsymbol{h}^{\dagger} = \frac{1}{4\sqrt{2}} (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3}) \tag{2.28b}$$

where the first component of $\boldsymbol{h}$ is on the left. The wavelets corresponding to (2.28a) are exactly the Haar function mentioned above.

Some additional remarks relating the two algorithms are in order. The conditions (2.14c) and (2.14d) effectively make $\boldsymbol{g}$ a bandpass filter and $\boldsymbol{h}$ a low-pass filter (e.g., an interpolation filter) with the sum on g_k analogous to the condition $\int \psi(t)\, dt = 0$. Also, $\boldsymbol{d}^{i+1}$ corresponds to $\boldsymbol{w}^i$. The additional decimation appearing in (2.13b) would not appear in a translation invariant version of Mallat's algorithm (cf. Section III). On the other hand, although the discrete filters $\boldsymbol{g}$ play algorithmically identical roles, the $\psi_n^i(t)$ of the à trous algorithm are not the wavelet vectors of a functional expansion. Rather the $\psi_n^i(t)$ are the duals of a set of vectors for which the coefficients of the signal expansion are w_n^i. That is, they are the coefficients of an expansion of the form $s(t) = \Sigma_{in} \langle s, \psi_n^i \rangle \tilde{\psi}_n^i(t)$ where $\langle\ \rangle$ indicates the L^2 inner product, and $\tilde{\psi}_n^i(t)$ is the dual basis or frame (see [3], [15]) of ψ_n^i. In Mallat's algorithm, since the $\psi_n^i(t)$ are orthonormal, the basis and its dual coincide. Thus, in many senses, the discrete filters $\boldsymbol{g}$ are more fundamental than the wavelets themselves. It is usually the coefficients which are of major interest; the actual wavelets $\psi_n^i(t)$, let alone their duals, are rarely computed.

[6]In contrast, the wavelet functions $\psi_n^i(t) \triangleq (1/\sqrt{2^i})\psi((t/2^i) - n)$ of (1.5) are not generally orthogonal. It is the filter constraints (2.14a) and (2.14b) that ensure orthonormality. Dropping these two constraints in Section IV, we develop a structure identical to (2.21)–(2.24); however, the constructed $\psi_n^i(t)$ need not be orthogonal.

Finally, in anticipation of Section V, let us form the squares of the two previous examples

$$h^\dagger * h = \frac{1}{\sqrt{2}}(1, 1) * \frac{1}{\sqrt{2}}(1, 1) = (1/2, 1, 1/2) \tag{2.29a}$$

and

$$h^\dagger * h = (-\tfrac{1}{16}, 0, \tfrac{9}{16}, 1, \tfrac{9}{16}, 0, -\tfrac{1}{16}) \tag{2.29b}$$

where $*$ indicates convolution, and h given by (2.28a) and (2.28b), respectively. Note that the filters $f = h^\dagger * h/\sqrt{2}$ are à trous filters and that the interpolation equation (2.10) holds exactly if $\psi(t)$ is a polynomial in t of degree one or three, respectively.

III. Wavelet Transforms Without Decimation

As has often been pointed out [7], the recursions (2.12) and (2.13) are not, in general, translation invariant. In contrast, the original undecimated à trous algorithm [4], [5], which is pictured in Fig. 5 and consists of entirely of convolutions, is patently translation invariant. In this subsection we shall use that property to provide a formal definition for an undecimated discrete wavelet transform $\tilde{w}$, then demonstrate that $w_n^i = \tilde{w}_{2^i n}^i$, and also show that $\tilde{w}$ is computed by the algorithm of Fig. 5.

Let T_m be the operation of translation by m; i.e.,

$$(T_m s)_n \triangleq s_{n-m}. \tag{3.1}$$

In order to include the dependency of w^i on s^0, we add it as an argument, writing $w^i(s^0)$. Equations (2.12a) and (2.12b) become

$$w^i(s^0) = G(\Lambda F)^i s^0. \tag{3.2}$$

(Recall that $G_{mn} \triangleq g_{m-n}$.) Finally, we shall need two important identities, which are proved in Appendix A. Namely, for any F of the form $F_{mn} = f_{m-n}$, we have the following lemmas.

Lemma 3.1:

$$[(\Lambda F)^i]_{nk} = [(\Lambda F)^i]_{0,k-2^i n} \tag{3.3}$$

Lemma 3.2:

$$\sum_k [(\Lambda F)^i]_{nk} e^{jk\omega} = e^{j2^i n\omega} \prod_{r=0}^{i-1} f_z(2^r\omega). \tag{3.4}$$

As expected, w^i is not translation invariant,

$$\begin{aligned}[w^i(T_m s^0)]_n &= \sum_k [G(\Lambda F)^i]_{n,k} s^0_{k-m} \\ &= \sum_k [G(\Lambda F)^i]_{n,k+m} s^0_k \\ &\neq \sum_k [G(\Lambda F)^i]_{n-m,k} s^0_k. \end{aligned} \tag{3.5}$$

For example, $[\Lambda F]_{mn} = f_{2m-n} \neq [\Lambda F]_{0,m-n}$. However, if we replace m by $2^i m$ in (3.5) and use Lemma 3.1, the last step becomes an equality so that

$$[w^i(T_{2^i m} s^0)]_n = [w^i(s^0)]_{n-m}. \tag{3.6}$$

Thus, translating s^0 by $2^i m$ translates octave i by m.

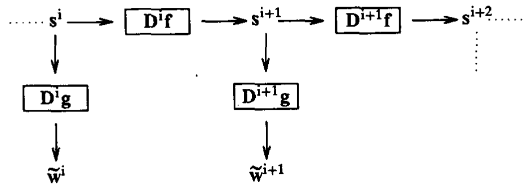

Fig. 5. The (undecimated) discrete wavelet transform. The filters $D^i f$ are obtained from f by inserting $2^i - 1$ zeros between each pair of filter coefficients. The operation of filtering is understood to mean convolution.

Note that the zeroth element of a series is invariant under decimation so that w_n^i and $\tilde{w}_n^i$ should coincide at $n = 0$. Utilizing this fact, we obtain the nth output of the undecimated discrete wavelet transform by translating the signal back by n samples and taking the decimated transform at time zero. More precisely,

Definition 3.1: Define the undecimated discrete wavelet transform $\tilde{w}$ in terms of the decimated transform w by

$$\tilde{w}_n^i \triangleq [\tilde{w}^i(s^0)]_n \triangleq [w^i(T_{-n}s^0)]_0. \tag{3.7}$$

We see that the desired invariance is achieved

$$\begin{aligned}[\tilde{w}^i(T_m s^0)]_n &= w^i(T_{-n}T_m s^0)]_0 \\ &= [w^i(T_{m-n}s^0)]_0 \\ &= [\tilde{w}^i(s^0)]_{n-m}. \end{aligned} \tag{3.8}$$

It is also clear from (3.6) and (3.7) that sampling $\tilde{w}_n^i$ every 2^i points produces exactly w_n^i, that is,

$$w_n^i = \tilde{w}_{2^i n}^i. \tag{3.9}$$

Next, we show that $\tilde{w}$ may be computed by the filter sequence pictured in Fig. 5. The proof is obtained by taking z transforms. From (3.2) and (3.7)

$$\begin{aligned}\sum_n \tilde{w}_n^i e^{-jn\omega} &= \sum_m \sum_n [G(\Lambda F)^i]_{0m} s_{m+n} e^{-jn\omega} \\ &= \sum_m [G(\Lambda F)^i]_{0m} e^{jm\omega} s_z(\omega). \end{aligned} \tag{3.10}$$

Applying Lemma 3.2 to (3.10), we have

$$\begin{aligned}\tilde{w}_z^i(\omega) &\triangleq \sum_n \tilde{w}_n^i e^{-jn\omega} \\ &= g_z(2^i\omega) \prod_{r=0}^{i-1} f_z(2^r\omega) s_z(\omega) \end{aligned} \tag{3.11}$$

where $i = 0$ is understood to mean there are no factors of f_z. As described in the next paragraph, this is exactly the z transform of the algorithm pictured in Fig. 5.

It is eady to see that $D^i f$ is f with $2^i - 1$ zeros inserted between every pair of filter coefficients and that its z transform is $f_z(2^i\omega)$. That is,

$$[D^i f]_n = \begin{cases} f_{n/2^i} & n = 2^i m \\ 0 & \text{other} \end{cases} \tag{3.12}$$

and

$$(D^i f)_z = f_z(2^i\omega). \tag{3.13}$$

Equation (3.11) is then equivalent to

$$s^{i+1} = (D^i f) * s^i \tag{3.14a}$$

$$\tilde{w}^i = (D^i g) * s^i \tag{3.14b}$$

where $s^0 \triangleq s$. This is essentially the original (undecimated) à trous algorithm found in [4] and [5]. However, we would like to emphasize that, since the development in this subsection has not made use of any filter constraints, the general equivalence of the decimated output of the algorithm pictured in Fig. 5 (equations (3.14)) and that of Fig. 1 (equations (2.12)) follows for arbitrary filters.

A. An Alternative Implementation

A second possibility for the implementation of $\tilde{w}$ is to use the algorithm in Fig. 1 and proceed directly from (3.7). That is, to output $\tilde{w}_n^i$, one simply translates the signal by n and then computes w_0^i. As an algorithm, this takes the form a) compute w_0^i; (b) translate the signal by one sample; (c) go to step (a). Moreover, in view of (3.6), one need not reperform the entire recursion (i.e., (2.12)) for every time point n in order to obtain $\tilde{w}^i$. Rather, at each octave, the decimation is replaced by a split into even and odd sequences, each of which is a starting point for the next octave (see Fig. 6). A couple of examples suffice to convince one that if, at octave i, $n \bmod 2^i = 0$, then one takes the upper branch; if $n \bmod 2^i = 1$, then one takes the lower branch. A rigorous derivation follows from the formula (cf. [20])

$$\mathbf{\Lambda} T_m s = \begin{cases} T_{m/2} s_{\text{even}} & m \text{ even} \\ T_{(m-1)/2} s_{\text{odd}} & m \text{ odd} \end{cases}. \tag{3.15}$$

We remark that Figs. 5 and 6 are computationally equivalent provided that the algorithm in Fig. 5 is implemented efficiently. The code must be written so as to omit multiplication by the zero elements of filters $D^i f$. (They are mostly zeros for $i > 2$.) Depending on the number of octaves, the computational burden still remains much greater than that of the decimated algorithm (i.e., Fig. 1); however, there is considerable parallelism which may be sufficiently exploited on suitable hardware to produce a real-time implementation [5].

IV. The DWT as an Exact Wavelet Transform

Regardless of the filters employed, one can, of course, perform the recursions (2.12) or (2.13) on the sampled signal s. Moreoever, provided that f (respectively, h) is low pass and g bandpass, the procedure may be interpreted physically as a bank of proportional bandwidth filters (cf. [21]–[24] also Section VI). In the present section, we examine the mathematical significance of relaxing the filter constraints (2.9) and (2.14). Our goal will be to relate the more general filter bank to the continuous wavelet transform, thus, in a sense, justifying the term DWT (cf. [25]). In this endeavor, the major questions which we shall address are: for what functions $\psi(t)$ is the recursion (2.12) an exact implementation of (1.6) and for which $\psi(t)$ and $s(t)$ do (1.5) and (1.6) coincide? The general answer is that we are able to construct such a ψ provided the discretized signal lies in the appropriate subspace of (cf. (2.26)). A somewhat surprising result is that it is necessary and sufficient for f to be à trous for condition (2.26) to be dropped. Our approach shall be to mimic the construction of orthonormal wavelets outlined in Section II-C.

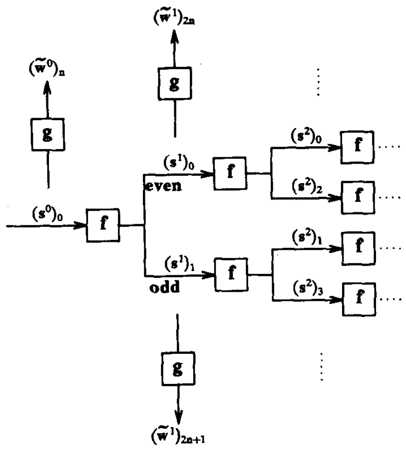

Fig. 6. Diagram of an implementation of the undecimated DWT.

A. Construction of the Scaling Function ϕ

We begin with the stipulation of the existence of a scaling function $\phi(t)$ with Fourier transform

$$\hat{\phi}(\omega) \triangleq \prod_{r=1}^{\infty} \left(\frac{1}{\sqrt{2}} \bar{f}_z \left(\frac{\omega}{2^r} \right) \right) \tag{4.1}$$

where $\bar{f}_z(\omega) = (f^\dagger)_z(\omega)$ is the z transform of $f^\dagger$. To emphasize the nonorthogonality of the corresponding wavelets, we retain the symbol f rather than h. Note that this function ϕ need not have (and in general does not have) all of the properties of the orthonormal ϕ outlined in Section II-C.

For (4.1) to converge to a nonzero function, the factors must approach one. Thus, $f_z(0) = 1$, which implies

$$\sum_k f_k = \sqrt{2}. \tag{4.2}$$

Even though ϕ could be normalized differently by the inclusion of a factor in (4.1), the filter f must necessarily obey the low-pass condition (4.2); i.e., (2.14d). Note also that, under the chosen normalization, $\int \phi(t)\, dt = \hat{\phi}(0) = 1$. However, without some additional conditions, the relationship of $\phi(t)$ to $\hat{\phi}(\omega)$ remains somewhat tenuous. Even under pointwise convergence, the limit may be a highly discontinuous, fractal function [9]. Suitable regularity conditions for the inverse Fourier transform of a product of the form (4.1) to converge to a reasonably behaved (e.g., $L^1(R)$, $L^2(R)$, and/or continuous) function may be found in [8] and [26]. The results are summarized in Appendix B.

A time domain representation of $\phi(t)$ in terms of f can be derived from (4.1). Let χ be the indicator function of $[-1/2, 1/2)$

$$\chi(t) \triangleq \begin{cases} 1 & \text{for } t \in [-1/2, 1/2) \\ 0 & \text{other} \end{cases}. \tag{4.3}$$

From (4.1) and Lemma 3.2, it is not difficult to see that

$$\phi(t) = \lim_{i \to \infty} \sum_k [(\mathbf{\Lambda F})^i]_{0k} \sqrt{2^i} \chi(2^i t - k). \tag{4.4}$$

In fact, the Fourier transform of (4.4) is[7]

$$\lim_{i \to \infty} \sum_k [(\mathbf{\Lambda F})^i]_{0k} e^{-j(\omega/2^i)k} \frac{1}{\sqrt{2^i}} \hat{\chi}\left(\frac{\omega}{2^i}\right) = \lim_{i \to \infty} \left(\prod_{r=1}^{i} \frac{1}{\sqrt{2}} \bar{f}_z\left(\frac{\omega}{2^r}\right)\right) \frac{\sin(\omega/2^{i+1})}{\omega/2^{i+1}} \tag{4.5}$$

which is just (4.1). Note that the existence of the function (4.4) could have been taken as the starting point for our analysis, since our proofs will not make use of (4.1).

We continue our parallel with Section II-C. Define $\phi_n^i(t)$ by definition 4.1, as follows.

Definition 4.1:

$$\phi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}} \phi\left(\frac{t}{2^i} - n\right). \tag{4.6}$$

Then, substitution of $2^i t - n$ for t in (4.4) and the use of Lemma 3.1 yield

$$\begin{aligned} \sqrt{2^i}\phi(2^i t - n) &= \lim_{j \to \infty} \sum_k [(\mathbf{\Lambda F})^j]_{0k} \sqrt{2^{j+i}} \chi(2^{j+i} t - 2^j n - k) \\ &= \lim_{j \to \infty} \sum_k [(\mathbf{\Lambda F})^{j-i}]_{0,k-2^{j-i}n} \sqrt{2^j} \chi(2^j t - k) \\ &= \lim_{j \to \infty} \sum_k [(\mathbf{\Lambda F})^{j-i}]_{nk} \sqrt{2^j} \chi(2^j t - k). \end{aligned} \tag{4.7}$$

On replacing i by $-i$, this becomes

$$\phi_n^i(t) = \lim_{j \to \infty} \sum_k [(\mathbf{\Lambda F})^{j+i}]_{nk} \sqrt{2^j} \chi(2^j t - k). \tag{4.8}$$

An immediate consequence of (4.8) is

$$\phi_n^{i+1}(t) = \sum_k [\mathbf{\Lambda F}]_{nk} \phi_k^i(t) \tag{4.9}$$

paralleling (2.22). Thus, we see that, despite their lack of orthogonality, the $\phi_n^i(t)$ have retained most of their structure. Furthermore, if f is a finite filter, then $\phi(t)$ has finite support [8]. More, precisely, suppose the coefficients of f are zero outside $[-N_-, N_+]$. Let $\eta^j(t) \triangleq [(\mathbf{\Lambda F}^j)]_{0k} \sqrt{2^j} \chi(2^j t - k)$. Then, $\eta^j(t)$ converges to $\phi(t)$, and, as in (4.9), we have

$$\eta^{j+1}(t) = \sum_k f_k^\dagger \eta^j(2t - k). \tag{4.10}$$

We find, for example, that η^j is zero for $t < t_j$ where $t_j = (t_{j-1} - N_-)/2$. With $t_0 = -1/2$, and with a similar calculation for the right half interval, it follows that $\phi(t) = \lim_{j \to \infty} \eta^j(t)$ is zero outside $[-N_-, N_+]$.

B. Exactness

To avoid confusion and stress their differences, let us first recapitulate some definitons. Four different transforms $W(a, b)$, $W(2^i, 2^i n)$, $w(2^i, 2^i n)$, and w_n^i have been mentioned ((1.1), (1.5), (1.6) and (2.12)). We retain a terminology parallel to Fourier transforms, namely, wavelet transform (WT), wavelet series,[8] discretized wavelet series, and discrete wavelet transform (DWT), respectively. The first two transforms involve integrals of a continuous signal; the latter two contain sums of sampled signals. The first three utilize a continuous wavelet function $\psi(t)$, the last one employs the discrete filters g and f. For consistency, we shall continue our development using decimated transforms; however, the results hold without change for undecimated transforms. This follows immediately, since they coincide for $n = 0$, and the undecimated transforms may be obtained at time $n = n_0$, by translating the signal by n_0 samples and taking the transform at $n = 0$. (See Section III where definition 3.1 remains valid for $W(2^i, n)$ and $w(2^i, n)$).

Our starting point shall be a signal $s(t)$ and discrete filters f and g with w^i defined by (2.12) or d^i by (2.13), i.e.,

$$\begin{aligned} s^{i+1} &= \mathbf{\Lambda F} s^i \\ w^i &= \mathbf{G} s^i \\ d^{i+1} &\triangleq \mathbf{\Lambda} w^i. \end{aligned} \tag{4.11}$$

Recall that the matrices F_{mn} and G_{mn} are given by f_{m-n} and g_{m-n}, respectively. Of course, we must also specify an initialization of the recursion (4.11) for some i; for example, for the zeroth octave s^0. The obvious choice is

$$s_n^0 \triangleq s(n) \tag{4.12a}$$

however, we shall also consider

$$s_n^0 \triangleq \sum_k \phi(k - n) s(k) \tag{4.12b}$$

which relates to the discretized wavelet series $w(2^i, 2^i n)$, and

$$s_n^0 \triangleq \int \phi(t - n) s(t)\, dt \tag{4.12c}$$

which corresponds to the sampled WT (wavelet series). For a given g, we shall construct a continuous function $\psi(t)$ such that the DWT of (4.11) is an exact implementation of the discretized wavelet series under (4.12b) and of the wavelet transform under (4.12c).

Define $\psi(t)$ by

$$\psi(t) \triangleq \sum_k \phi(t + k) \bar{g}_k = \sum_k \phi(t - k) g_k^\dagger \tag{4.13}$$

[7]This equality is immediate from Lemma 3.2 and $\hat{\chi}(\omega) = (2 \sin(\omega/2))/\omega$.

[8]At times, we shall prefer the term sampled WT rather than wavelet series in order to emphasize its role as a restriction of the continuous transform.

and $\psi_n^\dagger(t)$ by

$$\psi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}}\,\psi\left(\frac{t}{2^i} - n\right). \quad (4.14)$$

The recursion (4.9) implies that $\phi_n^i(t) = \Sigma_k [(\Lambda F)^i]_{nk}\phi_k^0(t)$. Using this expression, along with (4.6), the definitions (4.13) and (4.14) yield

$$\begin{aligned}\psi_n^i(t) &= \sum_k \bar{g}_k \frac{1}{\sqrt{2^i}}\,\phi\left(\frac{t}{2^i} - n + k\right)\\ &= \sum_k \bar{g}_{n-k}\frac{1}{\sqrt{2^i}}\,\phi\left(\frac{t}{2^i} - k\right)\\ &= \sum_k \bar{g}_{n-k}\phi_k^i(t)\\ &= \sum_k [\bar{G}(\Lambda F)^i]_{nk}\phi_k^0(t). \quad (4.15)\end{aligned}$$

If we take (4.12b) as the definition of s^0, the discretized wavelet series (1.6) takes the form

$$\begin{aligned}w(2^i, 2^i n) &\triangleq \sum_m \bar{\psi}_n^i(m)s(m)\\ &= \sum_{mk} [G(\Lambda F)^i]_{nk}\phi_k^0(m)s(m)\\ &= [G(\Lambda F)^i s^0]_n \quad (4.16)\end{aligned}$$

which, by (4.11) is exactly w_n^i. Furthermore, under (4.12c) we have (again using (4.15))

$$\begin{aligned}W(2^i, 2^i n) &\triangleq \int \bar{\psi}_n^i(t)s(t)\,dt\\ &= \int \sum_k [G(\Lambda F)^i]_{nk}\phi_k^0(t)s(t)\,dt\\ &= \sum_k [G(\Lambda F)^i]_{nk}\int \phi_k^0(t)s(t)\,dt\\ &= [G(\Lambda F)^i s^0]_n \quad (4.17)\end{aligned}$$

again w_n^i.

Finally, let us investigate the significance of the à trous condition; i.e., of the constraint (2.9), $f_{2k} = \delta_{k0}/\sqrt{2}$. We prove the following theorem.

Theorem 4.1: f is an à trous filter $\Leftrightarrow \phi(n) = \delta_{n0}$.

Proof: Letting $i = -1$, $n = 0$, and $t = n$ in (4.9) gives

$$\phi(n) = \sum_k [\Lambda F]_{0k}\sqrt{2}\,\phi(2n - k). \quad (4.18)$$

Then,

$$\phi(n) = \delta_{n0} \Rightarrow \sqrt{2}\,f_{2n} = \delta_{n0}. \quad (4.19)$$

Conversely, suppose that $f_{2n} = \delta_{n0}/\sqrt{2}$. Then, since $\chi(-k) = \delta_{k0}$, (4.8) with $i = 0$ and $t = 0$ implies

$$\begin{aligned}\phi(-n) &= \lim_{j\to\infty} [(\Lambda F)^j]_{n0}\sqrt{2^j}\\ &= \lim_{j\to\infty}\sum_k [(\Lambda F)^{j-1}]_{nk}[\Lambda F]_{k0}\sqrt{2}\,\sqrt{2^{j-1}}\\ &= \lim_{j\to\infty} [(\Lambda F)^{j-1}]_{n0}\sqrt{2^{j-1}}\\ &\cdots = \lim_{j\to\infty} [\Lambda F]_{n0}\sqrt{2} = \delta_{n0}. \quad (4.20)\end{aligned}$$

The import of Theorem 4.1 quickly follows. Equation (4.12b) implies (4.12a) for arbitrary signals if and only if $\phi(n) = \delta_{n0}$. Thus,

Corollary 4.1: The algorithm (4.11) is an exact implementation of the DWT with s^0 set equal to the sampled signal if and only if f is an à trous filter.

Furthermore, g is also related to $\psi(t)$ by sampling:

Corollary 4.2: If f is an à trous filter, then $\psi(t)$ defined by (4.13) satisfies $\psi(n) = g_n^\dagger$.

Wavelet coefficients of the type $d^i = \Lambda w^{i-1}$ are easily obtained by replacing $\psi(t)$ by $\sqrt{2}\,\psi(2t)$. More precisely, let $\psi'(t) \triangleq \sqrt{2}\,\psi(2t) = \sqrt{2}\,\Sigma_k\,\bar{g}_k\phi(2t + k)$, which coincides with (2.23). Then, provided s^0 satisfies (4.12c), (4.11) and (4.17) imply

$$\begin{aligned}d_n^i &= w_{2n}^{i-1}\\ &= \int \bar{\psi}_{2n}^{i-1}(t)s(t)\,dt\\ &= \int \frac{1}{\sqrt{2}}\,\bar{\psi}_n^{\prime i-1}\left(\frac{t}{2}\right)s(t)\,dt\\ &= \int \bar{\psi}_n^{\prime i}(t)s(t)\,dt. \quad (4.21)\end{aligned}$$

Hence, although the $\psi_n^{\prime i}(t)$ are not orthogonal, the Mallat algorithm still computes the wavelet transform. Of course, under (4.12a) and à trous, or for (4.12b), we have $d_n^i = \Sigma_k\,\bar{\psi}_n^{\prime i}(k)s(k)$, the counterpart of (4.16). It is interesting that, in a sense, the decimated wavelet transforms (1.5b) and (1.6) contain superfluous information. That is, they are underdecimated by a factor of two, and, thus, w^i property belongs to octave $i + 1$.

C. Summary

Let us summarize the results of this section. We are given discrete filters f and g such that (4.4) is well defined. Define $\psi(t)$ by

$$\psi(t) \triangleq \sum_k \phi(t - k)g_k^\dagger \quad (4.22)$$

with corresponding transforms sampled WT (wavelet series)

$$W(2^i, 2^i n) \triangleq \int \bar{\psi}_n^i(t)s(t)\,dt \quad (4.23)$$

and discretized wavelet series

$$w(2^i, 2^i n) \triangleq \sum_k \bar{\psi}_n^i(k)s(k). \quad (4.24)$$

Let $\phi_n \cdot s$ stand for the scalar product $\Sigma_k\,\phi_n(k)s_k$, and $\phi_n(t)\cdot s(t)$ for the L^2 scalar product $\int \phi_n(t)s(t)$. Then

$$f \text{ is à trous} \Rightarrow \psi(n) = g_n^\dagger. \quad (4.25)$$

For s discrete:

$$s^0 = s \text{ and } f \text{ is à trous} \Rightarrow w(2^i, 2^i n) = w_n^i \quad (4.26a)$$

$$s_n^0 = \phi_n \cdot s \Rightarrow w(2^i, 2^i n) = w_n^i. \quad (4.26b)$$

For $s(t)$ continuous:

$$s_n^0 = \phi_n(t) \cdot s(t) \Rightarrow W(2^i, 2^i n) = w_n^i. \qquad (4.27)$$

All the above results extend immediately to the undecimated transforms, $W(2^i, n)$, $w(2^i, n)$, and $\tilde{w}_n^i$ by translation invariance and definition 3.1. Also, the corresponding properties hold for $\boldsymbol{d}^i$ with $\psi(t)$ replaced by $\sqrt{2}\,\psi(2t)$. Note that the orthonormal wavelets are only exact under (4.26b) or (4.27) since their filters $\boldsymbol{h}$ are not à trous filters. Furthermore, they do not satisfy (4.25).

V. Lagrange Interpolation Filters

A reasonable class of à trous interpolators to consider for (2.10) are those which are exact for polynomials $P(t)$ of degree $\leq M$ for some M, i.e., for which

$$\frac{1}{\sqrt{2}} P\left(\frac{n}{2}\right) = \sum_k f_{n-2k}^{\dagger} P(k). \qquad (5.1)$$

For reasons which will become clear very shortly, we shall call these filters Lagrange à trous filters. Since the à trous filter $\boldsymbol{f}$ satisfies $f_{2k} = \delta_{0k}/\sqrt{2}$, (5.1) is an identity for n even. Let $\boldsymbol{a}$ contain the odd components of $\boldsymbol{f}^{\dagger}$

$$a_k \triangleq \begin{cases} \sqrt{2} f_{2k-1}^{\dagger} & \text{for } k > 0 \\ \sqrt{2} f_{2k+1}^{\dagger} & \text{for } k < 0. \\ 1 & \text{for } k = 0 \end{cases} \qquad (5.2)$$

Then (5.1) is equivalent to

$$P\left(\frac{n}{2}\right) = \sum_{k>0} a_k P\left(\frac{n+1}{2} - k\right) + \sum_{k<0} a_k P\left(\frac{n-1}{2} - k\right) \qquad (5.3)$$

for n odd and for all polynomials P of degree $\leq M$. (Actually, a single value of n implies (5.3) for all n; see (5.4).) We proceed to express the a_k in terms of Lagrange polynomials, and to show that the above conditions are essentially equivalent to $\boldsymbol{f} = \boldsymbol{h} * \boldsymbol{h}^{\dagger}/\sqrt{2}$ where $\boldsymbol{h}$ is an appropriate Daubechies filter.

A. Construction of $\boldsymbol{a}$

First, we parameterize the family of filters $\boldsymbol{a}$ satisfying (5.3) by $M + 1$, the dimension of the space of polynomials for which it must hold. For such a relationship to exist, one must relate the length of the filter (the number of unknowns) to M. To accomplish this, we shall assume that $\boldsymbol{a}$ has exactly the minimum number of coefficients needed to satisfy (5.3). We further assume that $\boldsymbol{a}$ has symmetric support; i.e., there is an N such that $\boldsymbol{a}_k = 0$ for $|k| > N$ and $\boldsymbol{a}_k \neq 0$ for $|k| = N$. This assumption is not unreasonable, at least for symmetric wavelets $\psi(t)$, since there is no reason *a priori* to distinguish between t and $-t$, and one would even expect $\boldsymbol{a}$ to be symmetric. We shall see, in fact, that the weaker condition of symmetric support joined with the previous constraints implies that $\boldsymbol{a}$ actually is symmetric.

The minimum number of coefficients (unknowns) equals the dimension of the space, i.e., satisfies $2N = M + 1$. Thus, the sums in (5.3) go from 1 to N. Moreover, since for any n, the polynomials $Q(x) \triangleq P(x - (n-1)/2)$ also form a basis, (5.3) is equivalent to

$$P\left(\tfrac{1}{2}\right) = \sum_{k=1}^{N} a_k P(1-k) + \sum_{k=-1}^{-N} a_k P(-k) \qquad (5.4)$$

which must hold for all $P(x)$ of degree $\leq 2N - 1$.

We pick out the kth coefficient by letting P be the Lagrange polynomial

$$L_j^{2N-1}(x) \triangleq \frac{\prod_{i \neq j} (x - i)}{\prod_{i \neq j} (j - i)} \qquad i, j \text{ in } [-N+1, N]. \qquad (5.5)$$

Then, $L_j^{2N-1}(k) = \delta_{jk}$, so that replacing P in (5.4) with L_j^{2N-1}, we get

$$a_j = L_{1-j}^{2N-1}\left(\tfrac{1}{2}\right) \qquad \text{for } j = 1, \cdots, N \qquad (5.6a)$$

$$a_{-j} = L_j^{2N-1}\left(\tfrac{1}{2}\right) \qquad \text{for } j = 1, \cdots, N. \qquad (5.6b)$$

Inasmuch as the Lagrange polynomials L^{2N-1} form a basis for polynomials of degree less than or equal to $2N - 1$, these a_k are in fact the unique solution to (5.4).

It is also straightforward to see from (5.5) and (5.6) that $\boldsymbol{a}$ is symmetric, i.e., for $j > 0$

$$a_j = \frac{\prod_{i \neq 1-j} (1/2 - i)}{\prod_{i \neq 1-j} (-j + 1 - i)} = \frac{\prod_{i \neq j} (-1/2 + i)}{\prod_{i \neq j} (-j + i)} = \frac{\prod_{i \neq j} (1/2 - i)}{\prod_{i \neq j} (j - i)} = a_{-j}. \qquad (5.7)$$

In summary, we have the following theorem.

Theorem 5.1: Let f be an à trous filter, i.e.,

$$f_{2k} = \frac{1}{\sqrt{2}} \delta_{0k}. \qquad (5.8)$$

Assume, furthermore, that f is real with symmetric support described by $k \in [-2N + 1, 2N - 1]$. Then, f is a Lagrange à trous filter, that is, (5.1) holds for all polynomials P of degree $\leq 2N - 1$, if and only if the odd components of f are determined by (5.6). Furthermore, f is necessarily symmetric.

B. Relationship to Daubechies (QMF) Filters

In [8], Daubechies constructs essentially the entire class of finite length filters $\boldsymbol{h}$ which satisfy (2.14) and fulfill suitable regularity conditions on (2.18). Explicitly, they take the form

$$\hat{h}(z) = \frac{1}{\sqrt{2}} \left(\frac{1}{2}(1 + z)\right)^N Q(z) \qquad (5.9)$$

where $\hat{h}(z) \triangleq \Sigma\, h_n z^{-n}$ is the z transforms of $\boldsymbol{h}$, and Q is an appropriately constrained polynomial. (In this section it is convenient express the z transform as a polynomial which we denoted $\hat{h}(z)$ where $h_z(\omega) = \hat{h}(e^{j\omega})$.) Her derivation uses a specific g, which up to a phase factor is given by

$$g_n = (-1)^n h(1-n) \tag{5.10}$$

and, as a consequence, (2.14a) reduces to

$$\Lambda h h^{\dagger} D = I. \tag{5.11}$$

In other words, $[hh^{\dagger}]_{2n} = \delta_{0n}$, which is the à trous condition for $hh^{\dagger}/\sqrt{2}$. Finally, if Q is taken to be of minimal degree (which turns out to be $N-1$), $|Q|^2$ is unique [8]. In other words, the squares of these filters are characterized completely by satisfying (2.14d), (5.11), and being of the form (5.9) with degree $2N-1$. We proceed to show that the $h * h^{\dagger}$ equal the Lagrange à trous filters.

Let h be the Daubechies filter of order $2N$. Since $h * h^{\dagger}$ is symmetric, the above conditions are equivalent to

$$\hat{\beta}(z) \triangleq h * h^{\dagger} = 1 + \sum_{1}^{N} b_k z^{2k-1} + \sum_{1}^{N} b_k z^{-2k+1} \tag{5.12a}$$

and

$$\hat{\beta}(z) = \tfrac{1}{2}(\tfrac{1}{2}(1+z))^N(\tfrac{1}{2}(1+z^{-1}))^N Q(z)Q(z^{-1}). \tag{5.12b}$$

That is, $\sqrt{2}f = h * h^{\dagger}$ if and only if $\hat{f}(z)$ is of the form (5.12a), (5.12b) and is of degree $2N-1$ in z (respectively, in z^{-1}).

Next, we show that the b_k coincide with the a_k of (5.3). We multiply (5.12a) by z^n for an arbitrary integer n,

$$z^n \hat{\beta}(z) = z^n + \sum_{1}^{N} b_k z^{n-1+2k} + \sum_{1}^{N} b_k z^{n+1-2k}. \tag{5.13}$$

The $2N$ zeros at $z = -1$ in (5.12b) imply that

$$\left.\frac{\partial^i (z^n \hat{\beta}(z))}{(\partial z)^i}\right|_{-1} = 0 \qquad i = 1, \cdots, 2N-1 \tag{5.14a}$$

and since $\hat{\beta}(-1) = 0$, we also have

$$2 \sum_{1}^{N} b_k = 1. \tag{5.14b}$$

Next, we note that

$$\left.\frac{\partial^i (z^m)}{(\partial z)^i}\right|_{-1} = m(m-1) \cdots (m-i+1)(-1)^{m-i} \quad \text{for } i > 0. \tag{5.15}$$

Define a set of polynomials P_i by

$$P_i(x) \triangleq 2x(2x-1) \cdots (2x-i+1). \tag{5.16}$$

Finally, setting the ith derivative of the right-hand side of (5.13) at $z = -1$ to zero, utilizing (5.15) and definition (5.16), we obtain

$$P_i\left(\frac{n}{2}\right) = \sum_{k>0} b_k P_i\left(\frac{n-1}{2} + k\right) + \sum_{k>0} b_k P_i\left(\frac{n+1}{2} - k\right) \tag{5.17}$$

where $i = 1, \cdots, 2N-1$. The signs work out since $n - 1 + 2k - i$ and $n + 1 + 2k - i$ have the same parity while $n - i$ differs. Define $P_0(x) \equiv 1$. Then, since the polynomials $P_0(x)$ and $P_i(x)$ for $i = 1, \cdots, 2N-1$ form a basis for polynomials of degree $\leq 2N-1$, and since (5.14b) implies (5.17) for $i = 0$, (5.17) must hold for arbitrary polynomials $P(x)$ of degree $\leq 2N-1$. Replacing P_i by P in (5.17) and setting $a_k = b_{|k|}$ yields (5.3).

Conversely, Theorem 5.1 implies that if a satisfies (5.3) for all polynomials of degree $\leq 2N-1$ and has symmetric support, then it must be symmetric. Clearly (5.17) must be also be satisfied. From (5.2) and reversing the above algebra, this is equivalent to (5.12a) and (5.14) with $\sqrt{2}\hat{f}(z)$ replacing $\hat{\beta}(z)$ where $\hat{f}(z)$ is of degree of $2N-1$ in z and also in z^{-1}. Letting $n = 1$ (or 0) in (5.14a), we see that $\hat{f}(z)$ has $2N$ roots at $z = -1$. Since f is symmetric, $\hat{f}(z)$ must also have the form (5.12b). We conclude that $\sqrt{2}f = h * h^{\dagger}$ where h is a Daubechies filter. Thus,

Theorem 5.2: There is a one-to-one correspondence between the squares of Daubechies orthonormal wavelet filters h of length $2N$ and the Lagrange à trous filters f of length $4N-1$ given by $f = h * h^{\dagger}/\sqrt{2}$.

Note that one can compute the h of length $2N$ by taking all possible square roots of the Lagrange à trous filters f.[9] f is easily computed from (5.6) where its even components given by $f_{2k} = \delta_{k0}/\sqrt{2}$, its odd positive components by $f_{2k-1} = b_k$ for $k = 0$ to N, and for odd negative k by symmetry. Also, the spectrum of f, $\hat{f}(e^{j\omega}) = |\hat{h}(e^{j\omega})|^2$, presents a convenient method of computing the power spectra of the h's. In another vein, since the h are maximally flat filters (i.e., have same number of vanishing derivatives at $z = 1$ and $z = -1$), Theorem 5.2 shows that a maximally flat filter is a Lagrangian interpolator; a fact which may aid in the design of such filters [14].

VI. Wavelet Filters in Signal Processing

This section has its roots in a question which originally motivated the author to undertake this study: Inasmuch as the à trous and Mallat algorithms share the same recursions, why not choose the Daubechies filters since they enjoy the additional advantage of orthonormality? The strongest arguments in favor of orthonormality seem to be mathematical elegance, ease of inversion, and, more subtly, good numerical properties. The major drawback is a lack of flexibility in filter design, in particular, an essentially fixed relative bandwidth. On the other hand, we have seen that the DWT has a firm analytical basis independent of the à trous approximation, even in the nonorthonormal case. In the present section, we shall briefly examine the issues of inversion, boundedness, and adjustable relative bandwidth. (Another fundamental issue, regularity of the associated wavelet functions, appears in Appendix B.) In

[9] During the revision of this paper, it was brought to the author's attention that an implicit relationship between the squared filters and Lagrange interpolation had been independently noted in private conversations between I. Daubechies and Ph. Tchamitchian. Similar observations are to appear in [27].

particular, we wish to establish conditions under which these properties obtain, expressed directly in terms of the filters f and g rather than in terms, for example, of the scale function $\phi(t)$ or wavelet $\psi(t)$. This view is in the spirit of the DWT as an entity in its own right, and it is certainly a necessary element in deciding which filters to use in practice.

Any software realization of the wavelet transform only implements a finite number of octaves. Mathematically, this reduces inversion to an algebraic question, one of finding filters which satisfy certain (not overly restrictive) equations. However, other considerations begin to come into play. Exact inversion requires finite filters, and, even then, exceedingly long filters may not be useful. Moreover, the constrained problem is considerably more difficult to solve. An alternative approach, approximation by truncated infinite filters, might be acceptable, but, once again, practical considerations dictate that the filters decay quickly. Similarly, the behavior of the DWT at infinity (i.e., w^i as i goes to infinity) becomes relevant. For example, the condition $\Sigma_s g_n = 0$, the discrete counterpart of (1.2), is not necessary for inversion of a finite number of stages. However, it is necessary to finite energy and boundedness, which are desirable properties inasmuch as they reflect directly on the numerical stability of the algorithm and/or its inverse (see, for example, [3]).

A. Inversion

To invert either the decimated or undecimated discrete wavelet transform it suffices to invert a single stage (octave); that is, to find s^i, given s^{i+1} and w^{i+1} or $\tilde{w}^{i+1}$. The equations for inverting the decimated algorithm are exactly analogous to those for the Mallat algorithm pictured in Fig. 4. One seeks two filters p and q which invert a single stage of the decimated DWT in Fig. 1; i.e., such that

$$\begin{aligned} s^i &= PDs^{i+1} + QDs^{i+1} \\ &= (PD)(\Lambda F)s^i + (QD)(\Lambda G)s^i. \end{aligned} \tag{6.1}$$

Equivalently,

$$(PD)(\Lambda F) + (QD)(\Lambda G) = I \tag{6.2}$$

where I is the identity matrix. This type of equation, which in the frequency domain may be separated into two equations comparable to (2.14a) and (2.14b), has been treated extensively (but not exhaustively) in the subband coding literature (cf. [12], [13], or even [8]). The QMF filters of the Mallat algorithm satisfy (6.2) with $g_z(\omega) = f_z(\omega + \pi)$, $p = f^\dagger$, and $q = g^\dagger$ (i.e., (5.10) and (5.11)). A less restricted class is just $p = f^\dagger$ and $q = g^\dagger$. The general class of filters satisfying (6.2), so-called biorthogonal filters, are examined in [28] and [29]. It should be emphasized that for perfect reconstruction in applications all filters must be of finite length (FIR). This does not imply that infinite filters (IIR) implemented by their truncations are not worthy of consideration [28].

For the undecimated algorithm the requirements for inversion are much less stringent. In order to invert a stage of the algorithm of Fig. 6, the filters p and q need only satisfy (cf. Fig. 7)

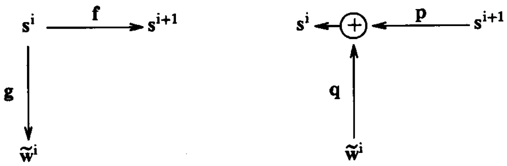

Fig. 7. Illustration of a single stage and its inverse for the undecimated algorithm found in Fig. 5.

$$\begin{aligned} s^i &= p * s^{i+1} + q * \tilde{w}^i \\ &= (p * f + q * g) * s^i. \end{aligned} \tag{6.3}$$

That is,

$$p * f + q * g = \delta \tag{6.4}$$

where the Kronecker delta, $\delta \triangleq \delta_{0,m}$, is the identity for convolution. This is a single equation, and consequently less restrictive than (6.2). If the polynomials formed by the z-transforms $\hat{f}(z)$ and $\hat{g}(z)$ are relatively prime, one may apply the Euclidean algorithm for the greatest common divisor (in this case, one) to find p and q. It has the advantage that finite f and g lead to finite p and q. Another method is simply to solve the equation in frequency space,

$$p_z(\omega) f_z(\omega) + q_z(\omega) g_z(\omega) = 1. \tag{6.5}$$

There is almost too much flexibility in solving this equation, although it becomes much more restrictive if one demands that the filters be finite or rapidly decreasing. Once again, a popular choice [30] is $f_z \bar{f}_z + g_z \bar{g}_z = 2$, which, for example, can be solved for g_z by taking the square root of $2 - |f_z^2|$ as long as $|f_z(\omega)|^2 \leq 2$. (Or, vice versa, it can be solved for f_z.) The Daubechies (QMF) filters $h/\sqrt{2}$ and $g/\sqrt{2}$ certainly satisfy this equation so that inversion for the undecimated version of the Mallat algorithm is immediate. Another case, useful in signal processing, is to choose f to be à trous, $p = f^\dagger/2$, and g any filter with nonvanishing spectrum except possibly where $|f_z(\omega)|$ equals $\sqrt{2}$ (see next subsection). Important questions of numerical stability, filter lengths, etc. certainly remain to be answered, but are well beyond the scope of the present paper.

Finally, before departing from this subject, it should be mentioned that inversion of the undecimated case in the form of Fig. 5 also follows from (6.4). The inverting filters are just $D^i p$ and $D^i q$. It is a simple matter to verify that (6.4) implies that

$$(D^i p) * (D^i f) + (D^i q) * (D^i g) = \delta. \tag{6.6}$$

(Inserting zeros in δ just yields δ.)

B. Finite Energy and Boundedness

The discrete wavelet transform is a mapping of sequences s_n, $n = 1, 2, \cdots$ into the space of doubly indexed sequences w_n^i, $i, n = 1, 2, \cdots$. Finite energy for the signal is simply

$$\sum_n |s_n|^2 < \infty. \tag{6.7}$$

One's initial tendency is to look for the same condition on the wavelet transform, i.e., $\Sigma_{i,n} |w_n^i|^2 < \infty$. However, the actual situation is not quite so transparent. Referring back to the continuous case, we have [16]

$$\int |s(t)|^2 \, dt = \int |W(a, b)|^2 \frac{da\, db}{a^2} \tag{6.8}$$

i.e., the weight (measure) $da\, db/a^2$ results in units of energy. For the DWT, $a = 2^i$ and db is either 1 (the undecimated case) or 2^i (the decimated case). Discretizing and noting that $da/a = d(\ln a)$, we see that

$$\frac{da\, db}{a^2} \rightarrow \frac{(\ln 2)\, \Delta i\, \Delta b}{2^i} = \ln 2 \left(\frac{\Delta b}{2^i}\right) \tag{6.9}$$

since $\Delta i = 1$. Observing that octave i has a bandwidth, up to a constant factor, of $1/2^i$, we assign (6.9) the following physical interpretation: The discrete wavelet transform is in units of power/hertz. Multiplying by the bandwidth $1/2^i$ gives power, and additional multiplication by the time interval Δb gives energy. Note that the decimated version only outputs every 2^i points so that in that case $\Delta b = 2^i$. In summary,

i) $|w_n^i|^2$, $|\tilde{w}_n^i|^2$ are in units of power/hertz;
ii) octave i has bandwidth $\approx 1/2^i$;
iii) $|\tilde{w}_n^i|^2/2^i$ and $|w_n^i|^2$ are in units of energy.

One should take care to note that for w_n^i, the energy weight $da\, db/a^2$ is a constant independent of i so that $|w_n^i|^2$ is discretized in a fashion so as to be both power/hertz and energy/cell.

Let $\|s\|^2 \triangleq \Sigma_n |s_n|^2$ be the squared norm of s, and define

$$\|\tilde{s}^\infty\|^2 \triangleq 2 \lim_{i \to \infty} \sup \frac{1}{2^i} \|s^i\|^2 \tag{6.10}$$

which corresponds to the DC energy (i.e., at $\omega = 0$). Finally, define the energy of the DWT by

$$\tilde{E} = \sum_i \frac{1}{2^i} \|\tilde{w}^i\|^2 + \|\tilde{s}^\infty\|^2 \tag{6.11a}$$

$$E = \sum_i \|w^i\|^2 + \|s^\infty\|^2. \tag{6.11b}$$

Energy conservation takes the form of the following Parseval's relationship for discrete wavelets:

Definition 6.1: A particular choice of filters f and g is said to be energy conserving if, for some constant C,

$$\|s\|^2 = C\left(\sum_i \frac{1}{2^i} \|\tilde{w}^i\|^2 + \|\tilde{s}^\infty\|^2\right). \tag{6.12}$$

One may also specify conservation for decimated transforms, in which case the 2^i is dropped.

Except for some clarifying remarks at the end, we restrict the discussion in the remainder of this subsection to the undecimated DWT. Following the above definitions, we see that the wavelet transform will have finite energy if and only if

$$\sum_i \frac{1}{2^i} \|\tilde{w}^i\|^2 + \|\tilde{s}^\infty\|^2 < \infty. \tag{6.13}$$

This is a necessary condition for the mapping $\tilde{w}$, from $l^2(Z)$ to $l^2(Z^2; 2^{-i}, 1)$ to be bounded. Of course, in practice, implementations never compute an infinite number of octaves. Nevertheless, the property (6.13) of finite energy can be quite important. Unbounded transformations tend to have poor numerical behavior even when truncated. Similarly, the inverse will not be bounded unless the series (6.13) is bounded below. A wavelet representation which has these properties,

$$A\|s\|^2 \le \sum_i \frac{1}{2^i} \|\tilde{w}^i\|^2 + \|\tilde{s}^\infty\|^2 \le B\|s\|^2 \tag{6.14}$$

is called a frame (cf. [3], although here the ambient Hilbert space is l^2 rather than $L^2(R)$.) We proceed to derive conditions on the filters f and g for (6.14) to hold.

Equations (3.14) imply

$$\int_{-\pi}^{\pi} |s_z^{i+1}(\omega)|^2 + \int_{-\pi}^{\pi} |\tilde{w}_z^i(\omega)|^2 = \int_{-\pi}^{\pi} (|f_z(2^i\omega)|^2 + |g_z(2^i\omega)|^2)|s_z^i(\omega)|^2. \tag{6.15}$$

Suppose that

$$\max_\omega \tfrac{1}{2}(|f_z(\omega)|^2 + |g_z(\omega)|^2) \le 1. \tag{6.16}$$

Then, in the time domain, (6.15) and (6.16) imply

$$\frac{1}{2^i}\|s^{i+1}\|^2 + \frac{1}{2^i}\|\tilde{w}^i\|^2 \le \frac{1}{2^{i-1}}\|s^i\|^2. \tag{6.17}$$

Adding $\|\tilde{w}^{i-1}\|^2/2^{i-1}$ to both sides and repeating for decreasing octaves implies that

$$\frac{1}{2^J}\|s^{J+1}\|^2 + \sum_{i \le J} \frac{1}{2^i}\|\tilde{w}^i\|^2 \le \|s\|^2. \tag{6.18}$$

Finally, letting J go to infinity, we get not only (6.13), but also the right inequality of (6.14) with $B = 1$.

However, the condition (6.16) is much too strong. That is, the transformation $g \rightarrow Cg$ for a large enough constant C would cause (6.16) to be violated even though C has no effect other than to multiply the total energy by a constant. In fact, the filters f and g produce finite energy transforms if and only if f and Cg yield finite energy. Thus, to have finite energy, it is sufficient to find a $C > 0$ such that $\max_\omega (|f_z(\omega)|^2 + C|g_z(\omega)|^2 \le 2$. Such a C exists provided that $|f_z(\omega)^2 \le 2$ and $(1/2)|g_z(\omega)|^2/(1 - [1/2]|f_z(\omega)|^2)$ is finite; i.e., is less than some finite $B = 1/C$. A similar argument holds for the lower bound. If

$$\min_\omega \tfrac{1}{2}(|f_z(\omega)|^2 + |g_z(\omega)|^2) \ge 1 \tag{6.19}$$

then (6.18) holds with the inequality reversed and the left inequality of (6.14) holds with $A = 1$. Once again, we apply the trick with the constant C and find that, for the inverse to be bounded, it is sufficient that there exist $A = 1/C > 0$ such that $(1/2)|g_z(\omega)|^2/(1 - [1/2]|f_z(\omega)|^2) \geq A$. In summary,

Theorem 6.1: A sufficient condition for the undecimated DFT $\tilde{w}$ and its inverse to satisfy (6.14) (that is, to be bounded) is that, for all ω, $|f_z(\omega)|^2 \leq 2$ and

$$0 < A \leq \frac{\frac{1}{2}|g_z(\omega)|^2}{1 - \frac{1}{2}|f_z(\omega)|^2} \leq B < \infty. \tag{6.20}$$

To satisfy (6.20), one must have $|f_z(\omega)| = 2 \Leftrightarrow g_z(\omega) = 0$, and the multiplicities of the corresponding roots must be identical. Note, also, that (6.20) can be used to give an estimate of B/A, the so-called tightness of the frame.

Whether these conditions are also necessary remains an open question. One can, however, show from (3.11) and an examination of the power $\tilde{w}_z^i(\omega)$ at $\omega = 0$ that a necessary condition is $g_z(0) = 0$ (equivalently, $\Sigma_n g_n = 0$). This is the discrete analog of the admissibility condition (1.2). The author conjectures that in the discrete case it is not a sufficient condition. (We remind the reader that even in those cases for which the DWT is exactly the sampled WT, finite energy of the continuous wavelet transform does not imply that of the discrete transform.) We do have, however, the following theorem:

Theorem 6.2: A necessary and sufficient condition for energy conservation (6.12) is that for all ω

$$\frac{1}{2}\left(|f_z(\omega)|^2 + \frac{1}{C}|g_z(\omega)|^2\right) = 1. \tag{6.21}$$

To prove this we may, without loss of generality, set $C = 1$ (i.e., redefine g by the constant factor C). Sufficiency follows as above, with inequalities replaced by equalities. To prove necessity, we first note that energy preservation for arbitrary signals implies the energy must be conserved for each stage. (For example, if the signal is s^1, energy must be preserved, and since it is preserved for s^0, the first stage must preserve energy.) From (6.15) this implies (6.21) with C equal to one.

The decimated case seems to present problems. In order for the above proofs to carry over, the even part of the signal would need to compensate for the lack of the factor $1/2^i$ in (6.11b), but $\|s^i_{\text{even}}\|^2 \neq 1/2\|s^i\|^2$. This problem presents yet another area for additional research.

C. Resolution and Relative Bandwidth

Considerable insight may be gained by viewing the algorithm in the frequency domain. One stage of the decimated DWT, illustrating (4.11) from this point of view, is pictured in Fig. 8. Since we are dealing with the discrete wavelet transforms, $\hat{s}(z) = \hat{s}(e^{j\omega})$ is evaluated on the unit circle. For convenience only the positive frequencies are pictured. Briefly, the algorithm is

a) Bandpass filter the upper half of the spectrum to yield w^i.
b) Low-pass filter to obtain the lower half of the spectrum ($[0, \pi/2]$).
c) Decimate to expand the lower half to $[0, \pi]$.
d) Go to a).

In somewhat more detail: We first obtain the high frequency information by using g to filter the upper half of the spectrum of s^i. The filter output is w^i. Then, in preparation for the next octave, s^i is low-pass filtered by f. This retains the, as yet unexamined, low frequency contents and also prevents the upper half of the spectrum from aliasing (i.e., contaminating the low frequency contents) in the dilation which follows. Finally, the operator Λ spreads[10] the remaining energy to fill the spectrum, producing octave $i + 1$. The procedure then repeats itself, s^{i+1} is bandpass filtered to get the spectral contents at frequencies which are, in absolute units, one half the frequencies of the previous octave.

A potential problem is immediately apparent. If the bandwidth of $g_z(\omega)$ is less than $\pi/2$, a portion of the signal energy will be discarded; it never appears in w^i. One possible remedy is to make g_z sufficiently broad; however, that would limit the resolution. Alternatively, we may introduce so-called voices. That is, we can employ a bank of filters of the type g (see Fig. 9) in order to cover the entire upper half of the spectrum.

We formalize some of these concepts using the modulated Gaussian of (1.4) as an example. With the introduction of an additional parameter β, $\psi(t)$ becomes

$$\psi(t) \triangleq e^{j\nu t}e^{-\beta^2t^2/2}. \tag{6.22}$$

Its Fourier transform is given by

$$\hat{\psi}(\omega) = \frac{1}{\beta}e^{-(\omega-\nu)^2/2\beta^2}. \tag{6.23}$$

We define the bandwidth of $\hat{\psi}(\omega)$ as twice the interval between points for which the modulus of (6.23) drops to $1/e$ of its peak value, i.e.,

$$\text{BW} \triangleq 2\sqrt{2}\,\beta. \tag{6.24}$$

The filter g (which, here, equals $g^\dagger$) is the sampled version of (6.22), that is,

$$g_n \triangleq e^{j\nu n}e^{-\beta^2n^2/2}. \tag{6.25}$$

For convenience, we set the sample rate equal to one. The following three restrictions on ν and β are necessary: First, in order that $g_z(\omega)$ lie in the upper half of the spectrum (cf. Fig. 8), we require that

$$\frac{\pi}{2} \leq \nu. \tag{6.26a}$$

Next, in order that $\psi(t)$ is admissible and analytic (see [18]), we demand

$$\beta \leq \frac{\nu}{2\pi}. \tag{6.26b}$$

[10] Λ, which decimates, is a contraction. Thus, in the Fourier domain, it is a dilation.

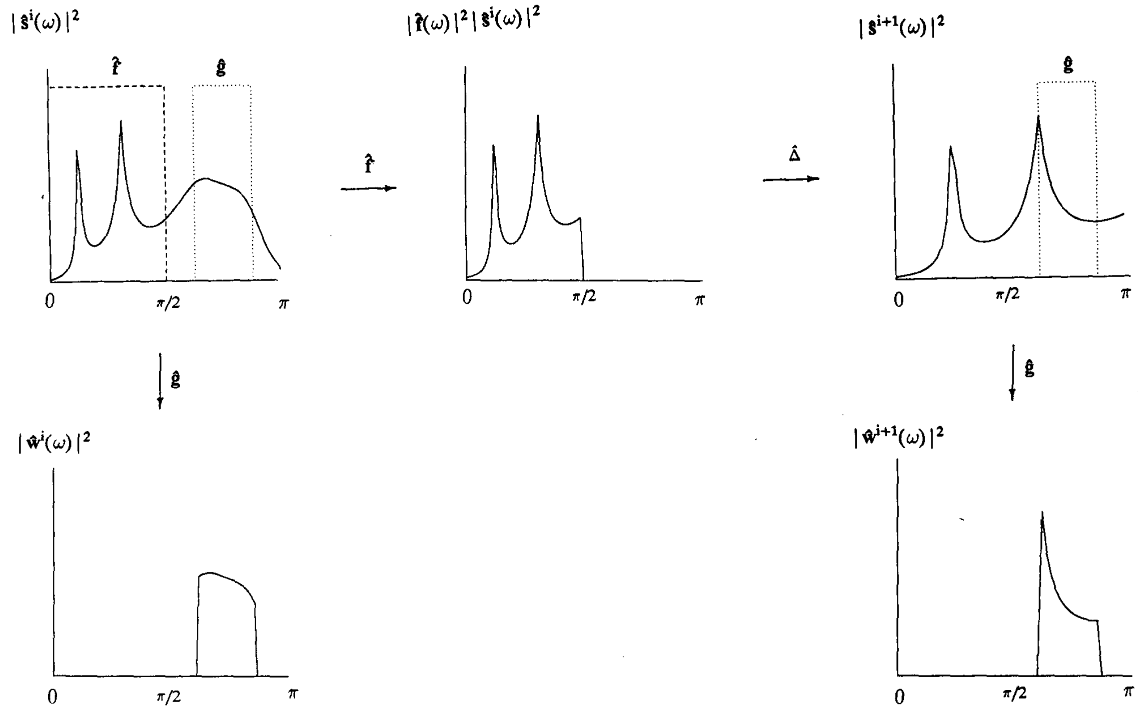

Fig. 8. Illustration of one stage of the algorithm, octaves i and $i + 1$ inclusive, viewed from the frequency domain.

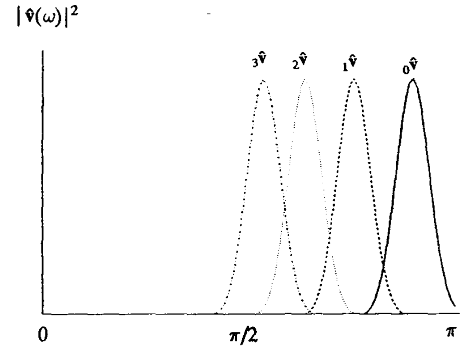

Fig. 9. Plot of the power spectra of the bandpass filters for four voices where $\hat{\psi}(\omega)$ is given by (6.23).

Under (6.26b), $g_z(\omega) \sim 0$ for $\omega \leq 0$. (Reference [18] recommends $\beta \leq \nu/5$ as being sufficient.) Finally, in order that the spectrum not be aliased, we set

$$\nu \leq \pi - \sqrt{2}\,\beta. \tag{6.26c}$$

These may be summarized in

$$\max(2\pi\beta, \pi/2) \leq \nu \leq \pi - \sqrt{2}\,\beta. \tag{6.27}$$

At this point a word of caution is advised. The bandwidth of the discrete filter $\boldsymbol{g}$ (e.g., (6.25)) is $2\sqrt{2}\,\beta$ only *when the sample rate is 1*. Since the ith octave is the result of i decimations by 2, sampling the original signal s^0 at a rate $\Delta t = 1$ results in a Nyquist frequency of $\pi/2^i$ for octave i (i.e., for s^i). Thus, the central frequency of $\boldsymbol{g}$, considered as a filter on octave i is $\nu/2^i$, and its bandwidth is $2\sqrt{2}\,\beta/2^i$. For this reason it is simpler and less ambiguous to speak in terms of a relative bandwidth which is independent of decimation. More precisely, we define

$$\text{RBW} \triangleq \frac{\text{BW}}{\text{mean frequency}}. \tag{6.28}$$

(Any appropriate representative of the "center" of the filter can replace the mean frequency in (6.28).) In the case of $\boldsymbol{g}$, we have

$$\text{RBW} = \frac{(2\sqrt{2}\,\beta/2^i)}{(\nu/2^i)} = \frac{2\sqrt{2}\,\beta}{\nu}. \tag{6.29}$$

Also, since $\pi/2 \leq \nu \leq \pi$, we have $2\sqrt{2}\,\beta/\pi \leq \text{RBW} \leq 4\sqrt{2}\,\beta/\pi$, or approximately

$$\beta \leq \text{RBW} \leq 2\beta. \tag{6.30}$$

In view of (6.30), we shall consider the parameter β as, essentially, the relative bandwidth of $\boldsymbol{g}$.

The number of voices M which we would expect to need to cover the upper half of the spectrum is

$$M \sim \frac{\pi/2}{2\sqrt{2}\,\beta} \sim \frac{1}{2\beta}. \tag{6.31}$$

The filter for voice j, which we denote ${}_j\boldsymbol{v}$, is determined by sampling the function $\psi(t/\alpha^j)$, i.e.,

$$[{}_j\boldsymbol{v}]_n \triangleq \psi(n/\alpha^j) \quad \text{for } j = 0, \cdots, M-1 \quad \text{and } \alpha \triangleq 2^{1/M}. \tag{6.32}$$

The power spectra of the ${}_jv$ are illustrated in Fig. 9. An alternative would be to define the voices as frequency translations of the filter g. However, (6.32) seems to be a more natural definition since it maintains the affine structure. (It is equivalent to taking a nondyadic value for the dilation parameter a in (1.1).) Note that the bandwidths of the voices decrease with j, and the spectral spacing $2\sqrt{2}\,\beta/\alpha^j$ differs somewhat from that assumed in (6.31).

Control over the relative resolution is important because, although the absolute resolution $(\beta/2^i)$ improves at higher octaves, at the same time, the analysis frequency decreases. If one wishes to improve resolution at a given frequency, one has to better the relative resolution, i.e., decrease β. On the other hand, the standard tradeoffs apply to choosing β. Small β increases the relative resolution but also requires more voices and a longer filter g; thus, more computation. (The duration of $\psi(t)$ is on the order of $2\sqrt{2}/\beta$ so that to provide a reasonable approximation to (6.22), the length of g must be proportional to $1/\beta$.) Moreover, the increased length of g implies a worsening of the (relative) time resolution. The time-bandwidth product is bounded below by the uncertainty principle, and no amount of computation will simultaneously produce arbitrarily small time and frequency resolution in a single wavelet transform.

With respect to choosing the low-pass filter f, we note that the spectrum of a longer filter f will generally have a sharper cutoff. This cutoff is relevant because it prevents the energy in the upper half of the spectrum from leaking (aliasing) into the lower half under the decimation Λ. For most applications, a Lagrange à trous filter of length 7 ($N = 2$) is sufficient [5]. One can, of course, also use the asymmetric filters h. Their discrimination of temporal direction seems intriguing, but remains univestigated.

Finally, how do the above considerations relate to orthonormal wavelets? The power spectra of filters h and g obeying (2.14) satisfy [8], [20]

$$|h_z(\omega)|^2 + |g_z(\omega)|^2 = 2 \tag{6.33a}$$

and

$$|h_z(0)| = |g_z(\pm\pi)| = \sqrt{2}. \tag{6.33b}$$

It follows that, for positive frequencies (likewise, for negative frequencies), g and h must each maintain a bandwidth on the order of $\pi/2$.[11] Thus, in exchange for orthonormality one relinquishes control over bandwidth. The relative bandwidth is, essentially, fixed at $(\pi/2)/(3\pi/4) = 2/3$. On the other hand, if one wishes to give time and frequency localization equal weight (e.g., $\beta = 1$ so that bandwidth = duration = $2\sqrt{2}$), a relative bandwidth in the neighborhood of $2/3$ is in a sense optimal.

[11]For the orthonormal wavelets of compact support, the larger the value of N, the more rapid the asymptotic convergence of $\hat{\psi}(\omega)$ to zero; i.e., as $\omega \to \infty$ [8]. This hints at a smaller bandwidth for $\hat{\psi}$ and, hence, also for g_z for large N. However, the speed at which the $\hat{\psi}(\omega)$ fall off near $\omega = 0$ appears to be fairly insensitive to N.

VII. Conclusion

We have seen that the à trous algorithm bears an intimate relationship to Mallat's multiresolution algorithm. Originally devised as a computationally efficient implementation, it is more properly viewed as a nonorthogonal multiresolution algorithm for which the discrete wavelet transform is exact. Moreover, the commonly used Lagrange à trous filters are simply the convolutional squares of the Daubechies filters for compact orthonormal wavelets.

From a broader viewpoint, these two algorithms are instances of the discrete wavelet transform (DWT), which, in more conventional terms, is simply a filter bank utilizing decimation and two filters. There are two basic versions of the DWT one of which is simply the decimated output (octave i is decimated by 2^i) of the other. The decimated DWT is characterized by octaves a) obtained by alternating a low-pass filter f with decimation, and b) tapped by a bandpass filter g to produce the output. The undecimated DWT inserts i zeros between the elements of the filters at octave i in lieu of decimation. (In the case of voices, several g's are used.) Finally, we note that under very general conditions, there exists a function $\psi(t)$ such that the filter bank outputs w_n^i correspond to the sampled wavelet transform

$$\frac{1}{\sqrt{2^i}} \int \bar{\psi}\left(\frac{t}{2^i} - n\right) s(t)\, dt,$$

thus, justifying the terminology discrete wavelet transform.

The personality of a given DWT is distinguished by the choice of filters. If f satisfies the à trous condition $f_{2n} = \delta_{n0}/\sqrt{2}$, then g is the sampled version of $\psi(t)$; i.e., $g_n = \bar{\psi}(-n)$. If finite length f and g obey the constraints of the multiresolution algorithm, then the $\sqrt{2^i}\,\psi(2^i t - n)$ are the compact orthonormal wavelets. A number of fundamental constraints have been discussed. In various combinations they have a bearing on the regularity of the wavelet function, on the energy in the transform domain, and on the boundedness and invertibility of the transform. In particular, we have provided a set of conditions on the filters sufficient for the transform and its inverse to be bounded. The signal processing properties of the discrete wavelet transform depend particularly strongly on the choice of g. The general constraints mentioned above are not restrictive on g; however, there is considerably less freedom in the orthonormal case. In particular, if orthonormality is a requirement, the half bandwidth of g (and, hence, the relative bandwidth of the wavelet) is no longer adjustable. It remains fixed at approximately $\pi/2$.

Many topics remain for investigation. Although considerable work has been done in finding filter pairs which have a complementary set for the inverse transform ([8], [28], [29]), it is far from exhaustive. The equivalence of maximally flat filters (with equal order roots at 0 and π) with Lagrange à trous filters as a design tool is perhaps worthy of investigation. Many of our filter conditions on energy are sufficient but possibly not necessary; a tight set of necessary and sufficient conditions for boundedness would be desirable. Finally, an investigation into the quality of the approximation of the DWT to the sampled WT in the case where it is not exact could be fruitful. It would perhaps lend more insight into the role of the regularity of the wavelet function in particular applications.

Appendix A
Proof of Lemmas 3.1 and 3.2

Lemma 3.1:

$$[(\mathbf{\Lambda F})^i]_{nk} = [(\mathbf{\Lambda F})^i]_{0,k-2^in}. \tag{A.1}$$

Proof: For $i = 1$, we have

$$[\mathbf{\Lambda F}]_{nk} = f_{2n-k} = [\mathbf{\Lambda F}]_{0,k-2n}. \tag{A.2}$$

Then, by induction,

$$\begin{aligned}[(\mathbf{\Lambda F})^i]_{nk} &= \sum_m f_{2n-m}[(\mathbf{\Lambda F})^{i-1}]_{mk} \\ &= \sum_m f_{2n-m}[(\mathbf{\Lambda F})^{i-1}]_{0,k-2^{i-1}m} \\ &= \sum_m f_{-m}[(\mathbf{\Lambda F})^{i-1}]_{0,k-2^{i-1}m-2^in} \\ &= \sum_m f_{-m}[(\mathbf{\Lambda F})^{i-1}]_{m,k-2^in} \\ &= [(\mathbf{\Lambda F})^i]_{0,k-2^in}. \end{aligned} \tag{A.3}$$

Lemma 3.2:

$$\sum_k [(\mathbf{\Lambda F})^i]_{nk} e^{jk\omega} = e^{j2^in\omega} \prod_{r=0}^{i-1} f_z(2^r\omega). \tag{A.4}$$

Proof: For $i = 1$, we have

$$\begin{aligned}\sum_k [\mathbf{\Lambda F}]_{nk} e^{jk\omega} &= \sum_k f_{2n-k} e^{jk\omega} \\ &= e^{j2n\omega} f_z(\omega). \end{aligned} \tag{A.5}$$

Then, by induction,

$$\begin{aligned}\sum_k [(\mathbf{\Lambda F})^i]_{nk} e^{jk\omega} &= \sum_k [(\mathbf{\Lambda F})]_{nk} e^{jk(2^{i-1}\omega)} \prod_{r=0}^{i-2} f_z(2^r\omega). \\ &= e^{j2^in\omega} \prod_{r=0}^{i-1} f_z(2^r\omega). \end{aligned} \tag{A.6}$$

Appendix B
Summary of Filter Constraints

The discrete wavelet transform $\tilde{\mathbf{w}}^i$ and the decimated discrete wavelet transform $\mathbf{w}^i$ (or $\mathbf{d}^i$) are defined for arbitrary filters $\mathbf{f}$ and $\mathbf{g}$ by

$$\mathbf{s}^{i+1} = (\mathbf{D}^i\mathbf{f}) * \mathbf{s}^i \tag{B.1a}$$

$$\tilde{\mathbf{w}}^i = (\mathbf{D}^i\mathbf{g}) * \mathbf{s}^i \tag{B.1b}$$

and

$$\mathbf{s}^{i+1} = \mathbf{\Lambda}(\mathbf{f} * \mathbf{s}^i) \tag{B.2a}$$

$$\mathbf{w}^i = \mathbf{g} * \mathbf{s}^i \tag{B.2b}$$

$$\mathbf{d}^{i+1} \triangleq \mathbf{\Lambda w}^i \tag{B.2c}$$

respectively. Also,

$$w_n^i = \tilde{w}_{2^in}^i. \tag{B.3}$$

In essentially all applications $\mathbf{f}$ is a low-pass filter and $\mathbf{g}$ is high-pass. This rather vague qualification is quantified below.

That is, in addition to the above definitions, it is expedient to impose auxiliary conditions on the filters to ensure a) that the DWT is related to some WT with a reasonably behaved scale function $\phi(t)$; b) that the transform have finite energy and be a bounded transformation; and, often, c) that it be invertible. The algebraic conditions for invertibility are found in (6.2) and (6.4). At the time of this paper no single set of necessary and sufficient conditions exist for the satisfaction of a) and b). Indeed, the definition of "reasonable behavior" of the scale function ultimately depends on the application. In an attempt to provide some degree of organization, we first list the candidate constraints, loosely labeled as low-pass, high-pass, or energy conditions. We then summarize their consequences. If either of the filters is infinite it is assumed to satisfy the decay condition [8], [26]

$$\exists\ \epsilon \text{ such that } \sum_n |f_n| n^\epsilon < \infty. \tag{B.4}$$

1. Candidate Constraints

i) Low pass

$$\sum_n f_n = \sqrt{2} \tag{B.5}$$

(i.e., $(1/2)f_z(0) = 1$).

ii) Energy

$$\tfrac{1}{2}|f_z(\omega)|^2 \leq 1. \tag{B.6}$$

iii) Low pass

$$\frac{1}{\sqrt{2}} f_z(\omega) = (1 + e^{j\omega})^N \gamma(\omega) \tag{B.7}$$

where $|\gamma(\omega)| \leq C < 1/2$ ((B.7) implies that $f_z(\pi) = 0$).

iv) Energy: complementary low-pass/high-pass pair

$$0 < A \leq \frac{\frac{1}{2}|g_z(\omega)|^2}{1 - \frac{1}{2}|f_z(\omega)|^2} \leq B < \infty. \tag{B.8}$$

Equation (B.8) implies

a) high pass

$$\sum_n g_n = 0 \tag{B.9}$$

(i.e., $g_z(0) = 0$)

b) low pass/high pass

$$\tfrac{1}{2}|f_z(\omega)| = 1 \Rightarrow g_z(\omega) = 0 \tag{B.10a}$$

$$g_z(\omega) = 0 \Rightarrow \tfrac{1}{2}|f_z(\omega)| = 1. \tag{B.10b}$$

v) Energy

$$\tfrac{1}{2}(|f_z(\omega)|^2 + C|g_z(\omega)|^2) = 1. \tag{B.11}$$

2. *Implications*

Necessity for pointwise convergence of (4.1) to $\hat{\phi}(\omega)$: (B.5).

Sufficiency for (4.1) and (4.4) to converge in $L^1(R)$, and $L^2(R)$ to continuous $\phi(t)$ and $\hat{\phi}(\omega)$, respectively: (B.5)–(B.7). This is one of the central results of [8], which also includes an examination of the decay of $\phi(t)$ and other regularity properties. Note that an important class of wavelets which does not fall under the domain of this theorem is the Haar wavelets (2.28a). Pointwise convergence still holds for the Haar wavelets, but they are not continuous.

Necessity for finite energy: (B.9).

Sufficiency for finite energy and that the transformation be bounded: (B.6) and $B < \infty$ in (B.8).

Sufficiency for a bounded inverse: (B.6) and $A > 0$ in (B.8).

Necessity and sufficiency for energy conservation: (B.11).

Acknowledgment

The author would especially like to thank D. K. Le for his meticulous reading of the manuscript and penetrating remarks. He is also grateful to D. Stein for several stimulating discussions. Finally, he thanks an anonymous reviewer for greatly simplifying the proof of Theorem 5.2.

References

[1] J. M. Combes, A. Grossman, and P. Tchamitchian, Eds., *Wavelets: Time-Frequency Methods and Phase Space.* Berlin: Springer, IPTI, 1989.
[2] A. Grossman, R. Kronland-Martinet, and J. Morlet, "Analysis of sound patterns through wavelet transforms," *Int. J. Patt. Recog. Artificial Intell.*, vol. 1, pp. 97–126, Aug. 1987.
[3] I. Daubechies, "The wavelet transform, time-frequency localization, and signal analysis," *IEEE Trans. Inform. Theory*, vol. 36, pp. 961–1005, Sept. 1990.
[4] M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets: Time-Frequency Methods and Phase Space.* Berlin: Springer, IPTI, 1989, pp. 286–297.
[5] P. Dutilleux, "An implementation of the algorithme à trous to compute the wavelet transform," in *Wavelets: Time-Frequency Methods and Phase Space.* Berlin: Springer, IPTI, 1989, pp. 298–304.
[6] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 2, pp. 674–693, July 1989.
[7] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 2091–2110, Dec. 1989.
[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, pp. 909–996, 1988.
[9] I. Daubechies, "Orthonormal bases of wavelets with compact support-connection with discrete filters," in *Wavelets: Time-Frequency Methods and Phase Space.* Berlin: Springer, IPTI, 1989, pp. 38–66.
[10] Y. Meyer, "Ondelettes, fonction splines, et analyses graduées," Univ. Torino, 1986.
[11] D. Esteban and C. Galand, "Application of quadrature mirror filters to split-band coding," in *Proc. ICASSP*, May 1977, pp. 191–195.
[12] M. Vetterli, "A theory of multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 356–372, Mar. 1987.
[13] M. J. T. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 434–441, June 1986.
[14] O. Hermann, "On the approximation problem in nonrecursive digital filter design," *IEEE Trans. Circuit Theory*, vol. 18, pp. 411–413, May 1971.
[15] C. Heil and D. Walnut, "Continuous and discrete wavelet transforms," *SIAM Rev.*, vol. 31, pp. 628–666, Dec. 1989.
[16] A. Grossman and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM J. Math. Anal.*, vol. 15, pp. 723–736, 1984.
[17] J. Morlet, G. Arens, E. Fourgeau, and D. Giard, "Wave propagation and sampling theory," *Geophysics*, vol. 47, pp. 203–236, Feb. 1982.
[18] A. Grossmann, R. Kronland-Martinet, and J. Morlet, "Reading and understanding continuous wavelet transforms," in *Wavelets: Time-Frequency Methods and Phase Space.* Berlin: Springer, IPTI, 1989, pp. 2–20.
[19] I. Daubechies, "Orthonormal bases of compactly supported wavelets II: Variations on a theme," preprint.
[20] M. J. Shensa, "The discrete wavelet transform," Naval Ocean Syst. Center, Tech. Rep. 1426, June 1991.
[21] G. Gambardella, "A contribution to the theory of short-time spectral analysis with nonuniform bandwidth filters," *IEEE Trans. Circuit Theory*, vol. 18, pp. 455–460, July 1971.
[22] R. Altes, "Sonar for generalized target description and its similarity to animal echolocation systems," *J. Acoust. Soc. Amer.*, vol. 59, pp. 97–206, Jan. 1976.
[23] M. Vetterli and C. Herley, "Wavelets and filter banks: Relationships and new results," in *Proc. ICASSP*, Apr. 1990, pp. 1723–1726.
[24] A. Kumar and D. R. Fuhrmann, "The Frazier-Jawerth transform," in *Proc. ICASSP*, Apr. 1990, pp. 2483–2486.
[25] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inform. Theory*, Mar. 1992.
[26] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$," *Trans. Amer. Math. Soc.*, vol. 315, pp. 69–87, Sept. 1989.
[27] R. Ansari, C. Guillemot, and J. F. Kaiser, "Wavelet construction using Lagrange halfband filters," *IEEE Trans. Circuits Syst.*, Sept. 1991.
[28] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, Sept. 1992.
[29] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," in preparation.
[30] S. G. Mallat and S. Zhong, "Complete signal representation with multiscale edges," NYU Comput. Sci. Tech. Rep. 483, Dec. 1989.

Mark J. Shensa was born in Cleveland, OH, in 1944. He received the B.A. degree in mathematics from Princeton University in 1966, the M.A. degree in mathematics from the University of California, Berkeley, in 1969, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1971.

From 1971 to 1974, he was an Associate Professor of Applied Mathematics at IME in Rio de Janeiro, Brazil. In 1974 he joined Texas Instruments, Inc. Since 1980 he has been with the Naval Ocean Systems Center in San Diego, CA. He served as an Editor of the IEEE Transactions on Acoustics, Speech, and Signal Processing from 1981 to 1984. He has made numerous contributions to technical journals in the last twenty years. This work includes singular perturbation theory, adaptive signal processing, tracking and localization, and Boolean logic. He is currently interested in wavelet transforms.