Tilings of the Time-Frequency Plane: Construction of Arbitrary Orthogonal Bases and Fast Tiling Algorithms

Cormac Herley, Jelena Kovačević, *Member, IEEE*, Kannan Ramchandran, and Martin Vetterli, *Senior Member, IEEE*

***Abstract*—We consider expansions which give arbitrary orthonormal tilings of the time–frequency plane. These differ from the short-time Fourier transform, wavelet transform, and wavelet packets tilings in that they change over time. We show how this can be achieved using time-varying orthogonal tree structures, which preserve orthogonality, even across transitions. The method is based on the construction of boundary and transition filters; these allow us to construct essentially arbitrary tilings. Time-varying modulated lapped transforms are a special case, where both boundary and overlapping solutions are possible with filters obtained by modulation. We present a double-tree algorithm which for a given signal decides on the best binary segmentation in both time and frequency. That is, it is a joint optimization of time and frequency splitting. The algorithm is optimal for additive cost functions (e.g., rate-distortion), and results in time-varying best bases, the main application of which is for compression of nonstationary signals. Experiments on test signals are presented.**

I. Introduction

RECENTLY, there has been a renewal of interest in linear expansions of signals, particularly using wavelets and some of their generalizations [1]–[3]. It is well known that the short-time Fourier transform (STFT), or Gabor transform, and the more recent wavelet transform (WT), are just two of many possible tilings of the time–frequency plane. These are illustrated in Fig. 1(a) and (b).

We use the term *time–frequency tile* of a particular basis function to designate the region in the plane which contains most of that function's energy. The tiling also shows the sampling in time and frequency, since it indicates where basis functions are localized in time and frequency. The rectangular representation for a tile is purely symbolic, since no function can have compact support in both time and frequency. Fig. 2 shows, for a Gaussian window $w(t)$, the product $T(t, \omega) = |w(t)|^2 \cdot |W(\omega)|^2$, where $W(\omega)$ is the Fourier transform of $w(t)$ ($W(\omega) = \int w(t) e^{-j2\pi\omega t}\, dt$). As can be seen, the shape of $T(t, \omega)$ is not rectangular, but it is clear where most of the energy is actually localized.

For a function $x(t)$, we can define its time spread (as-

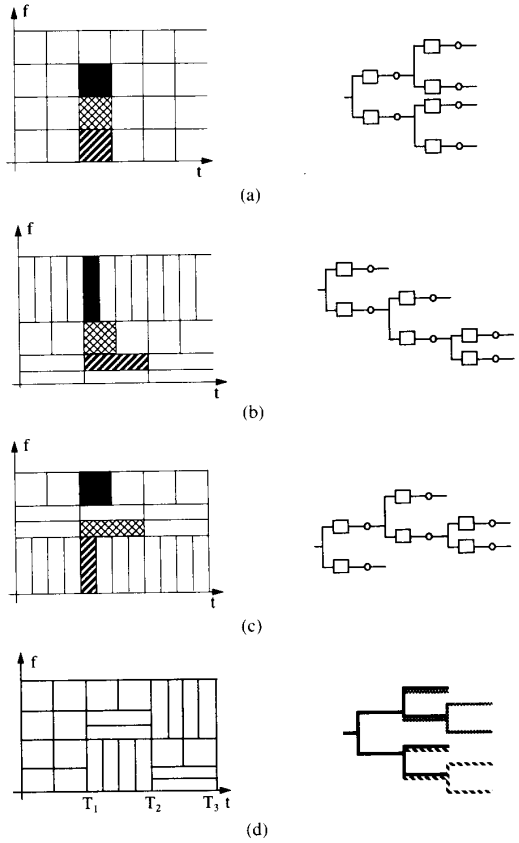


Fig. 1. Tilings of the time-frequency plane. (a) Short-time Fourier transform tiling. (b) Wavelet tiling. (c) Wavelet packet tiling. (d) Generalized tiling which adapts in time as well as in frequency. From $(0, T_1)$ the tiling is as in (a) and is denoted by the solid black line one the left. From (T_1, T_2), the tiling is as in (c) and is denoted by the gray line, while from (T_2, T_3), the tiling is as in (b) and is denoted by the dashed line on the left.

Manuscript received September 8, 1992; revised June 8, 1993. The Guest Editor coordinating the review of this paper and approving it for publication was Dr. Pierre Duhamel. This work was supported in part by the National Science Foundation under Grants ECD-88-11111 and MIP-90-14189, and in part by the New York State Science and Technology Foundation's CAT.

The authors were with the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, NY 10027-6699.

C. Herley and J. Kovacevic are now with AT&T Bell Laboratories, Murray Hill, NJ 07974-0636.

K. Ramchandran is now with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

M. Vetterli is now with the Electrical Engineering and Computer Science Department, University of California at Berkeley, Berkeley, CA 94720.

IEEE Log Number 9212171.

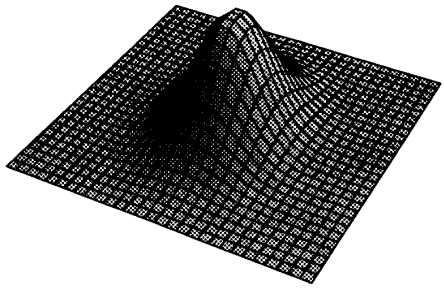


Fig. 2. An elementary tile corresponding to a Gaussian window.

suming that $x(t)$ has unit norm, that is, $\int |x(t)|^2\, dt = 1$) as

$$\Delta_t = \int_{-\infty}^{\infty} t^2 |x(t)|^2\, dt. \tag{1}$$

Similarly, its frequency spread, or bandwidth, can be defined as

$$\Delta_\omega = \int_{-\infty}^{\infty} \omega^2 |X(\omega)|^2\, d\omega \tag{2}$$

where $X(\omega)$ is the Fourier transform of $x(t)$. Then, the well-known uncertainty principle [4], imposes the following lower bound on the product of time and frequency spreads:

$$\Delta_t^2 \cdot \Delta_\omega^2 \geq \frac{\pi}{2}. \tag{3}$$

That is, one cannot have arbitrarily fine time and frequency resolutions, but can trade one for the other. In other words, the area of a time–frequency tile is roughly constant.

Consider, for example, Fig. 1(a) and (b) where two specific tilings are given, namely the short-time Fourier and the wavelet tilings. Note how in the former case, all tiles are of the same shape and size (i.e., time and frequency resolutions are constant), while in the latter case, the tiles are of different shapes (but of constant area) and trade frequency resolution for time resolution and vice versa.

One elegant generalization that contains, at least conceptually, Gabor and wavelet transforms as special cases, is the idea of wavelet packets (WP) [5]–[7], or arbitrary subband coding trees. An example of a wavelet packet tiling is given in Fig. 1(c). The main characteristic of the wavelet packet tiling is that it produces an arbitrary frequency split, which can be adapted to the signal. While wavelet packets create arbitrary binary slicing of frequencies (with associated time resolution), they do not change over time. Often a signal is first arbitrarily segmented, and then, the wavelet packet decomposition is performed on each segment in an independent manner. An obvious question is whether we can find a wavelet packet decomposition that changes over time, that is, an arbitrary tiling of the time–frequency plane. An example of such a generalized tiling is shown in Fig. 1(d). We use the term "arbitrary" somewhat freely, since the tiling is restricted to those produced by binary tree structures (possibly other trees if multiband filter banks are used).

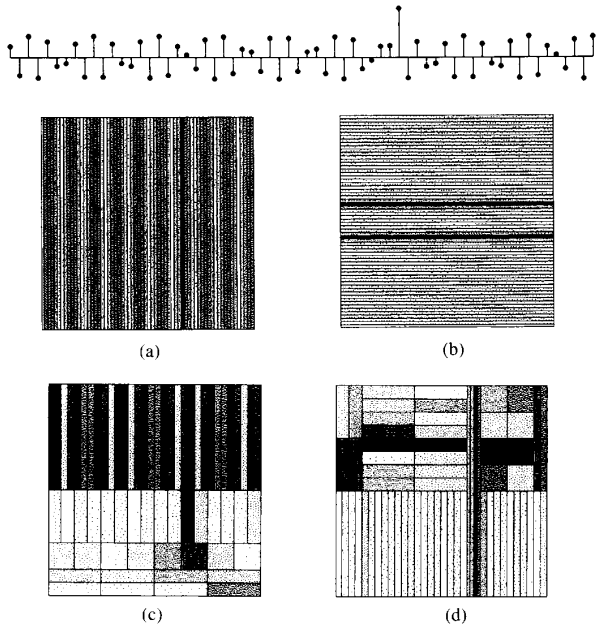


Fig. 3. Tilings from expansions using different bases. (a) Expansion using the unit sample functions as basis functions. (b) Expansion using the discrete Fourier transform as the basis. (c) Expansion using a discrete wavelet transform. (d) Expansion derived from time-varying filter banks.

The advantage of considering time-varying structures is illustrated in Fig. 3; here the signal consists of just 64 points, and so is sufficiently simple that we can display the energy contributed by the coefficient of each individual basis function. The signal consists of a high-frequency sinusoid and a spike, and is expanded using four different bases. In Fig. 3(a) the basis functions are the unit sample sequences, so only time-domain information is displayed: the spike is clearly seen, but the sinusoid is less obvious. In Fig. 3(b) we have used the discrete Fourier transform; as one might expect the frequency of the sinusoid is very evident, but the spike is not apparent. Fig. 3(c) shows the expansion using a discrete wavelet transform, which here has done quite badly, since it has very poor frequency resolution at high frequencies. Finally, Fig. 3(d) shows that a parsimonious representation of this signal is possible using a time-varying basis. Bases of this kind are constructed in Section III.

It is important to note that wavelet packet expansions, and the time-varying bases we will consider, are usually signal dependent. That is, given a signal, an algorithm finds the best set of basis functions for the expansion of this particular signal (out of a library of possible bases). This is similar to the Karhunen–Loève transform, which is a signal-dependent linear transform.

In this paper, we develop a generalized wavelet packet decomposition, in the sense that the signal segmentation and the wavelet packet decomposition in each segment are

found jointly. We call this an *adaptive wavelet packet decomposition*. Just as in regular wavelet packets, an orthonormal decomposition is performed, that is, for a given signal x, we want to find a complete set of basis functions such that we can write x as

$$x = \sum_k \langle x, \phi_k \rangle \phi_k \tag{4}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, and the set of basis functions is orthonormal

$$\langle \phi_k, \phi_l \rangle = \delta_{k-l}. \tag{5}$$

The aim of this paper is twofold. First, we would like to construct more general orthonormal sets of basis functions than the ones in the current literature (short-time Fourier transform, wavelet transform, wavelet packet transform). For example, in an adaptive wavelet packet scheme, we want basis functions which have overlaps between adjacent, but different, wavelet packet decompositions in order to avoid discontinuities at the boundaries. Such time-varying bases, which retain orthogonality as well, will be constructed.

Second, given an input signal, we want to find the "best" set of basis functions for that particular signal. A key question is what criterion should be used. Since we have orthogonal basis functions and a possible application is compression, we choose a rate-distortion framework, although any additive cost criterion can be used. Rate-distortion criterion is a typical example where one can trade one cost for the other. One-sided criteria (such as distortion only or entropy only [5]), are a particular case of the more general, two-sided additive criterion framework used here. For a rate-distortion approach to wavelet packets, see [7]. Since the number of bases is extremely large, the design of efficient algorithms to search for the best bases is an important problem, and will be addressed in the paper as well.

We will start by constructing discrete-time bases, which can be used for compression applications. Then, we explore the relationship between these discrete-time constructions and continuous-time ones, along the lines of wavelet constructions based on filter banks [1]. These lead to "arbitrary" continuous-time tilings, which are mostly of theoretical interest.

The relation of the results presented here with previous work is as follows: our work is an extension of wavelet packets [5], [6] and their construction in a rate-distortion sense [7]. Some preliminary results were given in [8]. For deriving our discrete-time results, we take a filter bank point of view [9]–[11]. Time-varying filter banks were constructed in [12] using a numerical optimization procedure but neither orthogonality nor algorithms to find the best basis were addressed. A development of continuous-time bases from discrete-time ones follows the work of Daubechies [1]. The particular case of boundary filters leads to wavelet constructions on the interval, a problem solved also in [13] using a different method. Finally, time-varying modulated lapped transforms (MLT's) have been considered briefly in [6]. The application of the double-tree algorithm to image compression, has shown promising results [14].

The outline of the paper is as follows: In Section II, we start by discussing signal expansions in the three-dimensional space of (time, frequency, scale). Recall, that the short-time Fourier transform and the wavelet transforms are expansions in (time, frequency) and (time, scale), respectively. Next, we construct adaptive wavelet packets, that is, time-varying forms of arbitrary subband coding trees. Two approaches are shown. The first, relies on boundary filters, and amounts to segmenting the signal without overlaps between adjacent wavelet packet decompositions. The second, and more general construction, leads to decompositions which change in time while avoiding abrupt transitions by using overlaps, either between different filters or different filter bank topologies. In Section IV, we investigate a special case of the general constructions from Section III, namely, time-varying modulated lapped transforms, and we present both boundary and overlapping solutions, where all the filters involved are obtained by modulation. In Section V, we describe algorithms to find best bases in a rate-distortion sense. After reviewing the algorithm for wavelet packets [7] which can be used for time-varying modulated lapped transforms as well, we develop the *double-tree algorithm* which we employ to jointly find the best segmentation of the signal together with the best wavelet packet expansion for every segment. Section VI presents experimental results both on synthetic and real signals. Improvements over nonadaptive wavelet packets are shown.

II. Signal Expansions Over Time, Frequency, and Scale

Given a basis function, denote by I_t the time interval which contains 90% of the function's energy. Similarly, call I_ω the frequency interval containing 90% of the energy of the Fourier transform of the basis function. Then, when choosing a basis function for a signal expansion, there are three essential parameters:

- its time location, given by τ, which corresponds to the center of the time interval I_t;
- its frequency location, given by ω, corresponding to the center of the frequency interval I_ω; and
- its scale, given by s, corresponding to the size of the basis function, or length of the interval I_t.

The scale is related to the time and frequency resolutions of the basis function, since the larger the scale, or size, of the basis function, the better its frequency localization (and inversely, the poorer its time localization). Thus, at least conceptually, one can think of analyzing a signal with what we could call a *three-indexed transform*, as shown in Fig. 4, where the expansion space (τ, ω, s) is shown. For a constant scale s_0, we obviously have a short-time Fourier transform, which is indicated by the vertical plane in Fig. 4. Along the curve $s \cdot \omega = \alpha$, the basis functions are scaled versions of each other (within

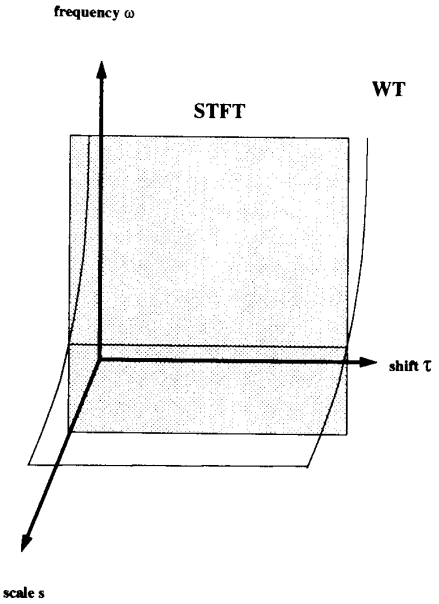


Fig. 4. Three-indexed transform. The expansion space is (τ, ω, s).

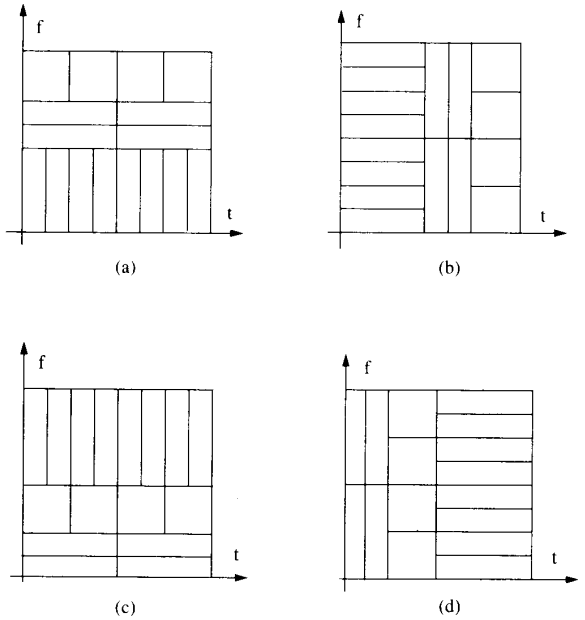


Fig. 5. Wavelet packets have their dual in time-varying modulated lapped transforms. (a) Wavelet packet tiling. (b) The dual time-varying MLT tiling. (c) Wavelet transform tiling. (d) The dual time-varying MLT tiling.

a scalar factor and a time shift). Thus, for various α_i's, we have different wavelet transforms on the surface obtained by translating in time the curve $s \cdot \omega = \alpha_i$ (where α_i is proportional to the number of periods of the modulation function present in the wavelet). One such wavelet family is indicated by the surface obtained from a hyperbole and its translation along the τ axis in Fig. 4.

While the three-indexed transform is useful as a conceptual framework to think about frequency and scale, it is not practical. Instead, we will turn to the discrete-time case and to filter banks, where the question of orthonormal bases is easy to study since it is more constrained. First, the sampling density question is easily solved. Each time the spectrum is divided into N equal pieces, the sampling is reduced by N as well.

Consider the time–frequency tiling obtained by a wavelet packet decomposition over a time interval, as shown in Fig. 5(a). It is clear that the slicing of the frequency axis does not change over time. Consider now Fig. 5(b), which is conceptually a dual to Fig. 5(a), since now time is sliced up, and this slicing does not change over frequency. What system produces such a dual time–frequency tiling? A uniform filter bank, where the number of channels changes over time. This can be achieved with time-varying modulated lapped transforms which are discussed in Section IV. Mixing the two special cases shown in Fig. 5(a) and (b) will produce more general, arbitrary tilings, and this is the topic of the constructions in Section III, which we call adaptive wavelet packets. Note, that since the wavelet tiling is the special case of the wavelet packet tiling, its dual can be obtained as a special case of time-varying modulated lapped transforms, as shown in Fig. 5(c) and (d).

Due to the relative ease of constructing these discrete-time bases, it is desirable to use them to derive continuous-time counterparts. The well-known construction of wavelets from octave band filter banks indicates that there are continuous-time orthogonal bases in the (time, frequency, scale) space (namely on the $(s \cdot \omega = \alpha, \tau)$ surface in Fig. 4). Wavelet packets give other possible orthogonal bases, and we will show that adaptive wavelet packets can be used as well to construct fairly general orthogonal bases in the (time, frequency, scale) space.

III. Arbitrary Trees Based on Two-Channel Filter Banks

Having set the goals that we wish to achieve, we now give our attention to two-channel filter banks and examine how to change between orthogonal trees based on such structures. The ability to change trees and filters will allow the construction of time-varying bases. This leads to general tilings of the type shown in Figs. 1(d) and 3(d). In this section, we indicate the various constructions and give examples involving small filters. For greater detail on the constructions, we refer the interested reader to [15], [16].

One convenient way to look at multirate filter banks is in terms of the time-domain operator notation. Here the action of the analysis filter pair $H_0(z)$ and $H_1(z)$ on the infinite signal column vector $\boldsymbol{x} = [\cdots x(-1), x(0), x(1), x(2), \cdots]^T$, as depicted in Fig. 6(a), can be expressed as $\boldsymbol{y} = \boldsymbol{T} \cdot \boldsymbol{x}$ where $\boldsymbol{y} = [\cdots y_0(0), y_1(0), y_0(1), y_1(1), \cdots]^T$ and $\boldsymbol{T}$ is the doubly-infinite block Toeplitz matrix [11],

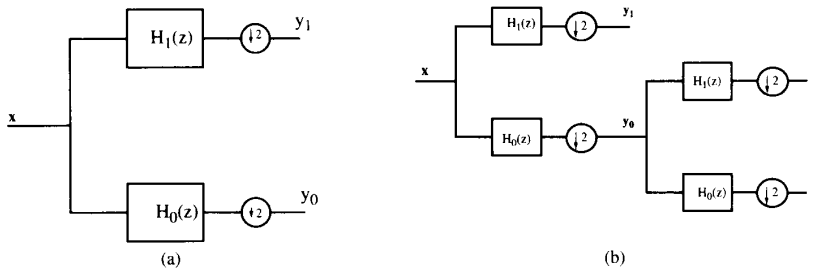


Fig. 6. Splitting of the input signal x using orthogonal two-channel filter banks. (a) Single division. (b) Iterated division.

[17], [18]

$$T = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & h_0(0) & h_0(1) & h_0(2) & \cdots & \cdots & h_0(N-1) & 0 & 0 & \cdots \\ \cdots & h_1(0) & h_1(1) & h_1(2) & \cdots & \cdots & h_1(N-1) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & h_0(0) & \cdots & \cdots & h_0(N-3) & h_0(N-2) & h_0(N-1) & \cdots \\ \cdots & 0 & 0 & h_1(0) & \cdots & \cdots & h_1(N-3) & h_1(N-2) & h_1(N-1) & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{6}$$

The rows of T are made up of the impulse responses, with even shifts, of the filters $h_0(n)$ and $h_1(n)$; it is readily verified that if these are filters from an orthogonal filter set then

$$TT^T = T^T T = I$$

where filter coefficients are assumed to be real. Further, in order to have an orthogonal FIR filter set, $H_0(z)$ must be of even length [17], and $H_1(z) = z^{2k-1}H_0(-z^{-1})$; we shall assume this form from now on, and take $k = N/2$ without loss of generality.

A. Boundary Filters

The assumptions of stationary filter bank analysis are that we are operating over an infinite signal (the matrix in (6) is infinite), and that the filters do not change over time (T is block Toeplitz, so all blocks are the same).

Relax the first of these assumptions. Consider the following simple example: suppose we wish to split a signal x using length-4 orthogonal analysis filters $H_0(z)$ and $H_1(z)$, but only in the time interval $0 \le n \le n_1$. Since we wish to retain orthogonality, the truncated matrix must be square; so consider the following truncation:

$$M = \begin{bmatrix} h_0(1) & h_0(2) & h_0(3) & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & h_0(0) & h_0(1) & h_0(2) \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & -h_0(3) & h_0(2) & -h_0(1) \end{bmatrix} \tag{7}$$

We have made use of the identity $H_1(z) = z^{N-1}H_0(z^{-1})$ in writing the coefficients of $H_1(z)$. For convenience label the rows of this matrix as $M_0, M_1, \cdots, M_{n_1}$. An important point is that this matrix is still of full rank; that is, the rows containing the truncated filters (M_0, M_1, M_{n_1-1}, and M_{n_1}) are linearly independent of each other, and of the other rows (M_i, $2 \le i \le n_1 - 2$) as well. To see this examine, for example, M_0 and M_1; these are clearly orthogonal to M_i for $i \ge 2$ since the truncation has not affected this orthogonality (there is no overlap); thus the truncated filters are linearly independent of the others. To show that M_0 and M_1 are linearly independent of each

other, suppose that they are not, i.e., $\alpha M_0 = M_1$ for some $\alpha \in R$. Thus,

$$\alpha(h_0(1), h_0(2), h_0(3)) = (h_0(2), -h_0(1), h_0(0)).$$

This implies $\alpha h_0(1) = h_0(2)$ and $\alpha h_0(2) = -h_0(1)$; the only solution is $\alpha = \pm j$ which is clearly inadmissible, since we require both filters to be real. The truncated matrix, of course, is no longer unitary, since M_0 and M_1 are not orthogonal. Since we have a full set of linearly independent vectors, however, we can restore orthogonality using the Gram–Schmidt procedure. Obviously,

$$\langle M_i, M_j \rangle = 0, \quad i \in \{0, 1\}, \quad j \in \{2, 3, \cdots n_1 - 1, n_1\} \tag{8}$$

this is inherited from the matrix (6). To orthogonalize the left boundary, start by normalizing the first vector $M_0' = M_0/\|M_0\|$, and then

$$M_1' = M_1 - \langle M_1, M_0' \rangle M_0' - \sum_{j=3}^{n_1} \langle M_1, M_j \rangle M_j, \tag{9}$$

$$= M_1 - \langle M_1, M_0' \rangle M_0'. \tag{10}$$

The simplification is a consequence of (8). Finally, set $M_1'' = M_1'/\|M_1'\|$. Note that since M_1 and M_0' each have only three nonzero entries, so does M_1'' from (10). The same procedure is applied to the other boundary vectors M_{n_1-1} and M_{n_1}. A new matrix $\boldsymbol{M}''$ which has rows

$$\{M_0'', M_1'', M_2, M_3, \cdots, M_{n_1-2}, M_{n_1-1}'', M_{n_1}''\} \tag{11}$$

is then obviously unitary. What is important to note is that $\boldsymbol{M}''$ has exactly the same zero entries as $\boldsymbol{M}$; i.e., the orthogonal boundary filters have the same support as the truncated filters.

One might imagine that this is a peculiarity of the case involving length-4 filters, but we can use exactly the same arguments for the length-6 case, taking a truncation where the top left corner is

$$\boldsymbol{M} = \begin{bmatrix} h_0(2) & h_0(3) & h_0(4) & h_0(5) & 0 & 0 & \cdots & \cdots & \cdots \\ -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & \cdots & \cdots & \cdots \\ h_0(0) & h_0(1) & h_0(2) & h_0(3) & h_0(4) & h_0(5) & \cdots & \cdots & \cdots \\ -h_0(5) & h_0(4) & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & \cdots & \cdots & \cdots \\ 0 & 0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & h_0(4) & h_0(5) & \cdots \\ 0 & 0 & -h_0(5) & h_0(4) & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{12}$$

Taking the same labeling convention as before, M_0 and M_1 are clearly orthogonal to all the other rows, and are linearly independent of them, and each other. Hence, again applying the Gram–Schmidt procedure produces a unitary matrix with the same zero entries as the truncated matrix.

Obviously, the orthogonalization given by Gram–Schmidt is not unique. If we choose a different set of linearly independent vectors as the input to the orthogonalization, we get a different set of orthogonal boundary vectors out. To explore the whole space of possible solutions $\{M_0'', M_1''\}$ and $\{M_{n_1-1}'', M_{n_1}''\}$ we can premultiply $\boldsymbol{M}$ by the matrix

$$\begin{bmatrix} \boldsymbol{U}_l & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_1-4} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{U}_r \end{bmatrix} \tag{13}$$

where $\boldsymbol{U}_l$ and $\boldsymbol{U}_r$ are 2×2 unitary matrices (rotations). Note that the particular form of the truncation was carefully chosen in the above two examples to preserve the rank of the truncated matrix. For longer filters there will be more than two boundary vectors, and proving that one can always choose a truncation which retains full rank becomes nontrivial. Nonetheless, for any FIR orthogonal filter bank, the orthogonal boundary filters always remain localized in the region of the boundary. We state this formally as follows.

Proposition 3.1 [16]: To apply a two-channel orthogonal filter bank, with length-N filters, to a finite length signal, requires a set of $((N-2)/2 + d)$ vectors at each boundary, each of which has only $(N - 2 + d)$ nonzero values, for any $d \geq 0$.

Since it is somewhat lengthy, the proof is given in [16]. Here, we are mostly interested in applications of the result, and thus, we merely communicate the main idea behind it. Note that boundary conditions have been studied before, for example, in [19]. However, here we are interested in solutions where the boundary filters are themselves orthogonal, and our method is constructive and complete.

Recall, once more, that N, the filter length, is always even. In the case where $N = 4k$ we choose $d = 0$, and when $N = 4k - 2$ we choose $d = 1$; this gives an even number of boundary filters in each case. Let us consider the $N = 4k - 2$ case and assume that we want to apply the filter bank to a signal of length L, where L is even. First define

$$\boldsymbol{Q} = [\boldsymbol{0}_1 \quad \boldsymbol{G} \quad \boldsymbol{0}_1] \tag{14}$$

where $\boldsymbol{G}$ is the $2k \times N + 2(k-1)$ matrix containing the shifted filter impulse responses of the filters $h_0(n)$ and $h_1(n)$, and $\boldsymbol{0}_1$ is a $2k \times 1$ column vector of zeros. It is easily shown that $\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{Q}^T\boldsymbol{Q})$ is the orthogonal projection onto the space orthogonal to the row space of $\boldsymbol{Q}$.

Hence, any row output from the Gram–Schmidt procedure must have the form

$$e_i' = (I - Q^T Q) \cdot e_i$$

for some e_i. The critical observation is that $(I - Q^T Q)$ has the following form

$$(I - Q^T Q) = \begin{bmatrix} I - U_0 & 0 & & \\ 0 & 0 & 0 & \\ & 0 & I - L_0 & \end{bmatrix}$$

where the blocks U_0 and L_0 are of dimension $(N - 2 + d) \times (N - 2 + d)$ each.

Thus, a vector e_i input to the orthogonalization procedure will always produce an output that has nonzero entries only in the first and/or last $N - 1$ positions. When the signal length L is large enough (roughly twice the filter length) the two nonzero portions of the Gram–Schmidt outputs e_i' are nonoverlapping. In this case, we can choose half of the inputs to the procedure to have zeros in the *last* $N - 1$ positions. Then, the corresponding outputs will have nonzero values only in the *first* $N - 1$ positions.

TABLE I
COEFFICIENTS OF THE BOUNDARY FILTERS IN THE NORMALIZED VERSION OF (7), WHERE $h_0(n)$ IS THE LENGTH-4 DAUBECHIES FILTER

n	$h_0'(n)$	$h_1'(n)$	$h_0''(n)$	$h_1''(n)$
0	0.9390708015	-0.3437237693	0.4034491106	0.2953452472
1	0.2976735161	0.8132591701	0.6987943579	0.5115529740
2	-0.1718618846	-0.4695354007	0.5906904945	-0.8068982213

lutions which have nonzero values at both boundaries correspond to the circulant solutions and variations.

B. Transitions Between Filter Banks with the Same Number of Channels

We can use boundary filters to make a nonoverlapping orthogonal transition between one filter set and another, or between two filter trees. For example, suppose we wished to change from the length-4 filter set $\{h_0(n), h_1(n)\}$ to length-6 filter set $\{g_0(n), g_1(n)\}$ at some time n_0. We could let T_0 be the half-infinite block Toeplitz matrix containing the impulse responses of the filters $\{h_0(n), h_1(n)\}$ in the region $(-\infty, n_0)$, and T_1 be the half-infinite block Toeplitz matrix containing the impulse responses of $\{g_0(n), g_1(n)\}$ in the region $(n_0 + 1, \infty)$; both matrices must contain the appropriate boundary filters. In this case we can use the operator

$$T = \begin{bmatrix} T_0 & 0 \\ 0 & T_1 \end{bmatrix} \tag{15}$$

which gives the appropriate transition between the filters at time n_0. This transition is of the form

$$\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_0(0) & l_0(1) & l_0(2) & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_1(0) & l_1(1) & l_1(2) & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & l_2(3) & l_2(4) & l_2(5) & l_2(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & l_3(3) & l_3(4) & l_3(5) & l_3(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & g_0(0) & g_0(1) & g_0(2) & g_0(3) & g_0(4) & g_0(5) & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & -g_0(5) & g_0(4) & -g_0(3) & g_0(2) & -g_0(1) & g_0(0) & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \tag{16}$$

Similarly, the second half of the Gram–Schmidt inputs can be chosen such that the outputs have nonzero values only in the last $N - 1$ positions. The $N = 4$ case constructed first was of this form, so that there were two filters each with three nonzero values at each boundary. An example of such a set of boundary filters for the length-4 Daubechies filters is given in Table I. The boundary filters $h_0'(n)$, $h_1'(n)$, $h_0''(n)$, and $h_1''(n)$ contain the three nonzero coefficients of M_0'', M_1'', M_{n_1-1}, and M_{n_1}'', respectively. So-

We noted before that it was possible to explore all possible boundary filters for the length-4, and length-6 filter banks by premultiplying by 2×2 unitary matrices, as in (13). If we premultiply (16) by a matrix

$$\begin{bmatrix} I & 0 & 0 \\ 0 & U_4 & 0 \\ 0 & 0 & I \end{bmatrix} \tag{17}$$

where U_4 is an arbitrary 4×4 unitary matrix, we get that the resulting matrix is still unitary. In general, however, all of the vectors will have support on both sides of the boundary

$$\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_0(0) & l_0(1) & l_0(2) & l_0(3) & l_0(4) & l_0(5) & l_0(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_1(0) & l_1(1) & l_1(2) & l_1(3) & l_1(4) & l_1(5) & l_1(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_2(0) & l_2(1) & l_2(2) & l_2(3) & l_2(4) & l_2(5) & l_2(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & l_3(0) & l_3(1) & l_3(2) & l_3(3) & l_3(4) & l_3(5) & l_3(6) & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & g_0(0) & g_0(1) & g_0(2) & g_0(3) & g_0(4) & g_0(5) & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & -g_0(5) & g_0(4) & -g_0(3) & g_0(2) & -g_0(1) & g_0(0) & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \tag{18}$$

We refer to the $l_i(n)$ as transition filters. Note that the boundary filter solution is contained as a special case: $l_0(n)$, $l_1(n) = 0$ for $n = 3, 4, 5, 6$, and $l_2(n)$, $l_3(n) = 0$ for $n = 0, 1, 2$. This clearly gives a nonoverlapping transition as in (15). An example of an overlapping solution is given in Table II.

C. Growing and Pruning the Tree

A main application is where we wish to change the structure of the tree over time; for example, if we wish to add a stage in going from the structure of Fig. 6(a) to that of (b) at some time n_0. To achieve this we would leave the output $y_1(n)$ alone, and process $y_0(n)$ with a further two-channel division, beginning at n_0. In operator notation the operation on $\boldsymbol{y}_0$ could be written

$$\boldsymbol{T}_1 \cdot \boldsymbol{y}_0 = \begin{bmatrix} \boldsymbol{I}_l & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_r \end{bmatrix} \cdot \boldsymbol{y}_0$$

where $\boldsymbol{I}_l$ is a half-infinite identity matrix extending infinitely to the left and $\boldsymbol{T}_r$ is a half-infinite matrix containing a set of boundary filters on the left, and then the filter impulse responses extending infinitely to the right. Then the overall operation can be expressed

$$\boldsymbol{T}_2 \cdot \boldsymbol{x} = \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \cdot \boldsymbol{P} \cdot \boldsymbol{y} \tag{19}$$

$$= \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \cdot \boldsymbol{P} \cdot \boldsymbol{T} \cdot \boldsymbol{x} \tag{20}$$

where $\boldsymbol{P}$ is the orthogonal permutation matrix that reorders the elements of $\boldsymbol{y}$

$$\begin{aligned} \boldsymbol{P} \cdot \boldsymbol{y} = [\cdots\ & y_0(0), y_0(1), y_0(2), \cdots, y_1(0), y_1(1), \\ & y_1(2), \cdots]^T \\ = [\boldsymbol{y}_0^T\ & \boldsymbol{y}_1^T] \end{aligned}$$

and $\boldsymbol{T}$ is as in (6). Clearly $\boldsymbol{T}_2$ is also orthogonal, and is again doubly infinite. Any binary tree based on orthogonal two-channel filter banks can thus be written as a cascade of such matrices.

D. Continuous-Time Bases from Discrete-Time Ones

So far we have considered only the construction of discrete-time orthogonal bases. In [1], [20], it was shown how discrete-time bases for $l^2(Z)$ could be used, under certain conditions, to generate continuous-time ones for $L^2(R)$; we will now suggest how the novel time-varying bases presented above may be used to derive continuous-time ones. This will lead to wavelet bases for interval regions, wavelet bases where the analyzing wavelet varies with time, and wavelet-like bases where the segmentation changes with time. Using a different approach wavelet bases for interval regions have also been investigated by Cohen, Daubechies, and Vial [13].

Again, we will rely heavily on matrix notation, and we examine the length-4 case for illustration. Consider the half-infinite block Toeplitz matrix which contains as rows the shifted impulse response of $h_0(n)$, and has the low-pass boundary filter in the first row

$$\boldsymbol{H}_0 = \begin{bmatrix}
h_0'(0) & h_0'(1) & h_0'(2) & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}. \tag{21}$$

TABLE II
COEFFICIENTS OF THE TRANSITION FILTERS IN (17), WHERE $h_0(n)$ AND $g_0(n)$ ARE THE LENGTH-4 AND LENGTH-6 DAUBECHIES FILTERS, RESPECTIVELY

n	$l_0(n)$	$l_1(n)$	$l_2(n)$	$l_3(n)$
0	4.8341612202e-01	1.2770611953e-01	0	0
1	8.3730128454e-01	2.2119348747e-01	0	0
2	2.2435420301e-01	-8.4926579221e-01	4.7788264955e-01	-6.4014861015e-03
3	-1.1503507360e-01	4.3545140498e-01	8.2771702907e-01	-1.1087699171e-02
4	3.5159731296e-02	-1.3309292472e-01	-2.4617419606e-01	5.1193184389e-01
5	1.9200602790e-02	-7.2681567455e-02	-1.4881697564e-01	-7.9408837886e-01
6	-7.9161511473e-03	2.9965635968e-02	6.1355244174e-02	3.2739199389e-01

Denote by $L_{ik}(z)$ the z-transform of the coefficients of the ith row of $\boldsymbol{H}_0^k$

$$L_{ik}(z) = \sum_{n=0}^{\infty} \boldsymbol{H}_0^k(i, n) z^{-n}.$$

Clearly $L_{11}(z) = H_0'(z)$, $L_{21}(z) = z^{-1}H_0(z)$, $L_{31}(z) = z^{-3}H_0(z)$. To find the $L_{i2}(z)$ we must examine the product $\boldsymbol{H}_0^2$. We find

$$L_{i2}(z) = z^{-1-4(i-2)}H_0(z)H_0(z^2), \qquad i > 1$$

and, in general

$$L_{ij}(z) = z^{-1-2^j(i-2)} \prod_{k=0}^{j-1} H_0(z^{2^k}) \qquad i > 1. \quad (22)$$

The function $L_{2j}(z)$ can easily be recognized as the z-transform of the "graphical iteration" [1] to find the scaling function $\phi(t)$ of a compactly supported wavelet scheme [17]. That is, if we define from $L_{ij}(z)$ a continuous-time function

$$f^{(j)}(t) = l_{2j}(n) \qquad n/2^j \le t < (n+1)/2^j \quad (23)$$

it can be shown that $f^{(j)}(t)$ converges to the scaling function $\phi(t)$ as $i \to \infty$ [under some constraints on $h_0(n)$]. Similarly, the other rows $L_{ij}(z)$, $i > 1$, can be used to converge to $\phi(t - i - 2)$. For the case of $L_{1j}(z)$ we can define a continuous-time function in a manner similar to (23). If this converges as $j \to \infty$ we call it the left-boundary scaling function. The wavelet $\psi(t)$ and the left-boundary wavelet function are found by examining the convergence of the rows of $\boldsymbol{H}_1 \cdot \boldsymbol{H}_0^k$. Right boundary functions are found using the right boundary filters. The left and right boundary wavelet functions along with $\psi_2(t)$ and $\psi_2(t - 1)$ are shown in Fig. 7. When we deal with Daubechies' filters we use the subscript to denote the order of the filters involved; i.e., $\phi_n(n)$ and $\psi_n(t)$ are the scaling function and wavelet derived from the length-$2n$ Daubechies' filters.

In a similar way, we can consider iterates of the matrix containing the transition filters. For example, if $\boldsymbol{H}_0$ is the doubly infinite matrix containing the even-indexed rows of $\boldsymbol{T}$ from (18) then the rows of $\boldsymbol{H}_0^k$ give the iterates of the graphical recursion that converges to $\phi_2(t - n)$ for rows $n < 0$, to $\phi_3(t - n)$ for rows $n \ge 3$, and to transition functions for $n = 0, 1, 2$. This is illustrated in Fig. 8, which shows the appropriate scaling and transition functions. It should be noted that the number of transition functions is not necessarily equal to the number of transition filters. Finding transition functions by considering iterates of matrices containing the transition filters between different filter bank topologies is investigated in [16].

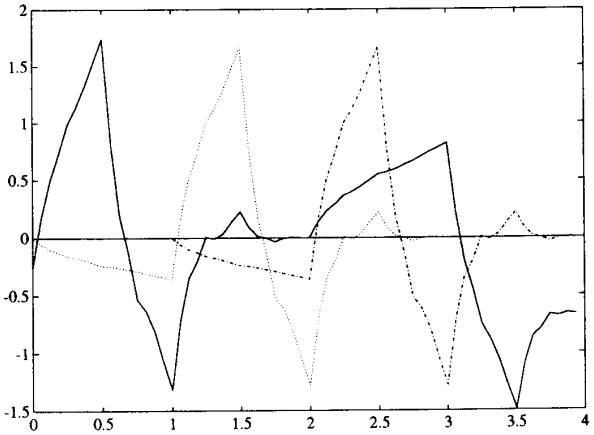


Fig. 7. Boundary functions for wavelet basis for the interval [0, 4). Left and right boundary functions at one scale are shown; these are supported on [0, 2), and [2, 4), respectively. The wavelets $\psi_2(t)$, $\psi_2(t - 1)$ are also shown.

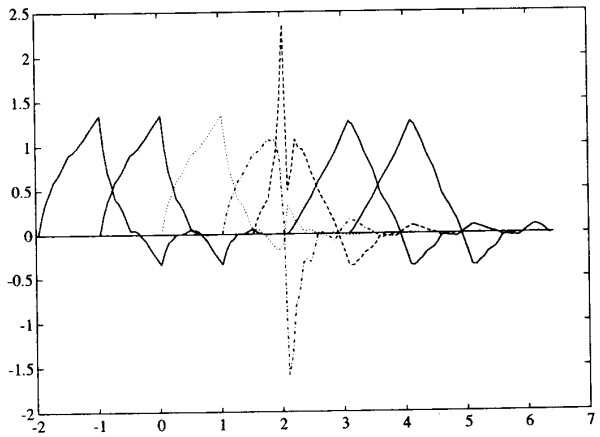


Fig. 8. Transition functions for the transition between $\phi_2(t)$ and $\phi_3(t)$. There are three orthogonal transition functions shown, which span the null space, at one scale, between $\{\phi_2(t + 1), \phi_2(t + 2), \cdots\}$ and $\{\phi_3(t - 2), \phi_3(t - 3), \cdots\}$.

The orthogonality properties that we desire of a wavelet basis (i.e., with respect to shift and scale) still hold for systems containing boundary and transition filters. As an illustration observe

$$\boldsymbol{H}_0^k \cdot (\boldsymbol{H}_0^*)^k = (\boldsymbol{H}_0 \cdot \boldsymbol{H}_0^*)^k = \boldsymbol{I}$$

and

$$\boldsymbol{H}_0^k \cdot (\boldsymbol{H}_1^*)^k = (\boldsymbol{H}_0 \cdot \boldsymbol{H}_1^*)^k = \boldsymbol{0}.$$

This ensures that the graphical recursions that converge to the wavelet and scaling function are orthogonal with respect to translates at the kth iteration; and this holds true in the limit as $k \to \infty$ also.

Equally, the kth iteration to the wavelet is orthogonal to the $(k-1)$st (i.e., its stretched version)

$$\boldsymbol{H}_1 \cdot \boldsymbol{H}_0^k \cdot (\boldsymbol{H}_0^*)^{k-1} \cdot \boldsymbol{H}_1^* = \boldsymbol{H}_1 \cdot \boldsymbol{H}_0 \cdot \boldsymbol{H}_1^* = \boldsymbol{0}. \quad (24)$$

In the limit this gives orthogonality across scales. More complete and formal statements of these arguments can be found in [15], [16].

E. Discussion

We have shown how to obtain arbitrary decompositions that can change over time. This can be achieved using time-varying orthogonal tree structures, which preserve orthogonality, even across transitions. The method is based on the construction of both boundary and transition filters; these allow us to construct essentially arbitrary tilings. The time-varying discrete-time bases were shown to generate time-varying continuous-time ones. A special case of the constructions presented in this section are the time-varying modulated lapped transforms given in the next section. We will show that boundary and overlapping solutions are possible there as well.

IV. Time-Varying Modulated Lapped Transforms

We have mentioned previously that the wavelet packets and time-varying *modulated lapped transforms* (MLT's) can, at least conceptually, be seen as duals of each other [see Fig. 5(a) and (b)]. By the same token, the wavelet tiling, being a special case of the wavelet packet tiling, has a dual in a particular time-varying modulated lapped transform tiling [see Fig. 5(c) and (d)].

Our aim in this section is to offer ways of constructing these dual tilings, by using either boundary or overlapping modulated lapped transforms. By boundary modulated lapped transforms, we denote the time-varying tiling of the time-frequency plane where the basis functions do not overlap, while the opposite is true for overlapping modulated lapped transforms, that is, the basis functions from adjacent decompositions do overlap. Although the tilings obtained via time-varying modulated lapped transforms are somewhat restricted as compared to those that can be obtained using the general theory presented in the last section (see Fig. 5), they offer the advantage that all the filters, both at transitions, and within decompositions, are obtained by modulation. Time-varying lapped transforms are also studied in [24].

A. Modulated Lapped Transforms

By modulated lapped transforms [21], [22], we will denote a class of perfect reconstruction filter banks which uses a single prototype filter, window, $w(n)$ of length $2N$ (where N is the number of channels and is even) to construct all of the filters $h_0, \cdots, h_{N-1}$ as follows:

$$h_k(n) = \frac{1}{\sqrt{N}}\, w(n) \cdot \cos\left(\frac{2k+1}{4N}(2n - N + 1)\pi\right) \quad (25)$$

with $k = 0, \cdots, N-1$, $n = 0, \cdots, 2N-1$, and where the prototype low-pass filter $w(n)$ is symmetric $(w(n) = w(2N-1-n), n = N, \cdots, 2N-1)$ and satisfies the following [22]:

$$w^2(n) + w^2(N-1-n) = 2, \qquad n = 0, \cdots, N-1. \quad (26)$$

This last condition, imposed on the window, ensures that the resulting modulated lapped transform is orthogonal. The two symmetric halves of the window are called "tails."

We have seen in the last section, a convenient way of analyzing filter banks in the time domain, via infinite matrices, such as the one given in (6). For modulated lapped transforms, the matrix $\boldsymbol{T}$ can be written as

$$\boldsymbol{T} = \begin{bmatrix} \ddots & & & & \\ & \boldsymbol{A}_0 & \boldsymbol{A}_1 & & \\ & & \boldsymbol{A}_0 & \boldsymbol{A}_1 & \\ & & & & \ddots \end{bmatrix} \quad (27)$$

where blocks $\boldsymbol{A}_0, \boldsymbol{A}_1$ are of sizes $N \times N$, and contain the impulse responses of the filters. Note that the filter length is twice the number of channels. For example, the jth row of $\boldsymbol{A}_i$ is $[h_j(2N-1-iN) \cdots h_j(N-iN)]$ for $i = 0, 1$. For an orthogonal, perfect reconstruction solution, the matrix $\boldsymbol{T}$ has to be unitary, which is equivalent to the following [11]:

$$\boldsymbol{A}_0\boldsymbol{A}_0^T + \boldsymbol{A}_1\boldsymbol{A}_1^T = \boldsymbol{I}, \qquad \boldsymbol{A}_1\boldsymbol{A}_0^T = \boldsymbol{0}, \qquad \boldsymbol{A}_0\boldsymbol{A}_1^T = \boldsymbol{0}. \quad (28)$$

The second and third conditions in the above are called the "orthogonality of tails" conditions [11].

We want to point out an interesting fact here, that will be used in later constructions. In the matrix $\boldsymbol{T}$, denote by w_A the window function corresponding to the block $\boldsymbol{A}_0$. Take the same block, that is, $\boldsymbol{A}_0$, and change the window function to w_B. Call this newly obtained block $\boldsymbol{B}_0$. Then

$$\boldsymbol{B}_0\boldsymbol{B}_0^T = \boldsymbol{A}_0\boldsymbol{A}_0^T \quad (29)$$

that is, the product $\boldsymbol{A}_0\boldsymbol{A}_0^T$ does not depend on the window (the proof of which is given in the Appendix). The same is true of the other block $\boldsymbol{A}_1$. An interesting consequence of the above, is that

$$\boldsymbol{B}_0\boldsymbol{B}_0^T + \boldsymbol{A}_1\boldsymbol{A}_1^T = \boldsymbol{A}_0\boldsymbol{A}_0^T + \boldsymbol{B}_1\boldsymbol{B}_1^T = \boldsymbol{A}_0\boldsymbol{A}_0^T + \boldsymbol{A}_1\boldsymbol{A}_1^T = \boldsymbol{I} \quad (30)$$

or, in other words, any combination of blocks with different window "tails" will be unitary, or, we will have a nonsymmetric window. Note, however, that this is not sufficient for a valid orthogonal transform, since the "orthogonality of tails" conditions from (28) do not hold, namely $\boldsymbol{A}_0\boldsymbol{B}_1^T \neq \boldsymbol{0}$ (similarly for $\boldsymbol{B}_0\boldsymbol{A}_1^T$). Nevertheless, the fact that $[\boldsymbol{A}_0 \quad \boldsymbol{B}_1]$ is unitary despite $\boldsymbol{A}_0$ and $\boldsymbol{B}_1$ having dif-

ferent windows, is going to be very useful in constructing boundary and overlapping modulated lapped transforms.

B. Boundary Modulated Lapped Transforms

Let us now try to construct a set of boundary filters for the modulated lapped transforms. From the results in the last section, we know that we can always do this by applying the Gram–Schmidt procedure to the appropriately truncated matrix $\boldsymbol{T}$. However, our approach here is different; namely, we want the boundary filters to be obtained by modulation as well. Therefore, consider the following matrix:

$$\boldsymbol{T}_b = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{A}_1 & & & \\ & \boldsymbol{A}_0 & \boldsymbol{A}_1 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{A}_0 & \boldsymbol{A}_1 \\ & & & & \boldsymbol{A}_0 \quad \boldsymbol{B}_1 \end{bmatrix} \tag{31}$$

where $\boldsymbol{A}_i$ are size-$(N \times N)$ blocks as introduced in (27), and $\boldsymbol{B}_i$ are size-$(N \times N/2)$ blocks with the associated window

$$w_B(n) = \begin{cases} \sqrt{2} & n = N/2, \cdots, 3N/2 - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

For example, the ith row of $\boldsymbol{B}_j$ is given by $[h_{jB}((3/2)N - 1) \cdots h_{jB}(N)]$, where $h_{jB}(n)$ is as given in (25) with the window $w_B(n)$. Note how the truncation was performed here as compared to that in (7). In $\boldsymbol{T}_b$, instead of using $\boldsymbol{A}_0$ at the left boundary, for example, and then truncating it, we have performed the truncation by using a shorter window (of length N instead of $2N$). The net result is a square matrix, for which we only have to check whether it is unitary. Using (28) and (30), it can be easily checked that $\boldsymbol{T}_b \boldsymbol{T}_b^T = \boldsymbol{I}$, holds.

Therefore, we have demonstrated how to construct boundary filters for a given modulated lapped transform, so as to be able to change decompositions over time. The transform is still orthogonal, as demonstrated by the fact that $\boldsymbol{T}_b \boldsymbol{T}_b^T = \boldsymbol{I}$. There are still N boundary filters, of length $3N/2$. The way they are obtained is by using a nonsymmetric window (which was shown to be possible in the last subsection).

C. Overlapping Modulated Lapped Transforms

Suppose now that we want to be able to switch between two different modulated lapped transforms, but with filters (basis functions) that overlap. We will first show how to change between two same-size modulated lapped transforms but with different windows, and then we will examine the case of switching between different-size transforms.

1) Same-Size Transforms: Let us now demonstrate how one can switch between two same-size modulated lapped transforms with different windows. Consider the following matrix:

$$\boldsymbol{T} = \begin{bmatrix} \ddots & & & & & & \\ & \boldsymbol{B}_0 & \boldsymbol{B}_1 & & & & \\ & & \boldsymbol{B}_0 & \boldsymbol{B}_1 & & & \\ & & & \boldsymbol{B}_0 & \boldsymbol{A}_1 & & \\ & & & & \boldsymbol{A}_0 & \boldsymbol{A}_1 & \\ & & & & & \boldsymbol{A}_0 & \boldsymbol{A}_1 \\ & & & & & & \ddots \end{bmatrix}. \tag{33}$$

Here, blocks $[\boldsymbol{B}_0 \quad \boldsymbol{B}_1]$ represent the N-channel modulated lapped transform with the window w_B, while $[\boldsymbol{A}_0 \quad \boldsymbol{A}_1]$ represent the N-channel transform with the window w_A. It is easily checked that the above matrix is unitary, leading to an orthogonal transform (even across the transition). Note how the transition $[\boldsymbol{B}_0 \quad \boldsymbol{A}_1]$ uses a nonsymmetric window (or half of the window w_B and half of the window w_A) to achieve a transition that is still orthogonal.

2) Different-Size Transforms: We will now explain how to switch between different-size transforms. Note that a similar result, without details of the construction or the proof, was mentioned in [6].

Here, we want to show that one can switch from an N_1-channel modulated lapped transform to an N_2-channel modulated lapped transform (where $N_1 < N_2$ and $N_1 = 2^{m_1}$, $N_2 = 2^{m_2}$) as follows.

a) Since the maximum overlap between the two modulated lapped transforms is N_1, one has to adjust the size of the window of the N_2-channel modulated lapped transform accordingly, that is, its length has to be reduced to $N_2 + N_1$, or the outer $(N_2 - N_1)/2$ coefficients of each window tail have to be zero. This means that the N_2-channel transform can be expressed as follows:

$$[\boldsymbol{A}_0 \quad \boldsymbol{A}_1] = \begin{bmatrix} \underbrace{\mathbf{0}}_{(N_2-N_1)/2} & \underbrace{\mathbf{A}_{00}}_{N_1} & \underbrace{\mathbf{A}_{01}}_{(N_2-N_1)/2} & \underbrace{\mathbf{A}_{11}}_{(N_2-N_1)/2} & \underbrace{\mathbf{A}_{10}}_{N_1} & \underbrace{\mathbf{0}}_{(N_2-N_1)/2} \end{bmatrix} \tag{34}$$

where the blocks $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ have been split into three subblocks each, to account for the fact that some of the coefficients of the window are set to 0. Therefore, the blocks $\boldsymbol{A}_{01}$ and $\boldsymbol{A}_{11}$ are those where the window is constant.

b) Then, construct the N_1-channel modulated lapped transform $[\boldsymbol{A}'_{00} \quad \boldsymbol{A}'_{10}]$ from the N_2-channel one as follows:

$$\boldsymbol{T}_o = \begin{bmatrix} \ddots & & & & & & \\ & \boldsymbol{A}'_{00} & \boldsymbol{A}'_{10} & & & & \\ & & \boldsymbol{A}'_{00} & \boldsymbol{A}'_{10} & & & \\ & & & \boldsymbol{A}_{00} & \boldsymbol{A}_{01} & \boldsymbol{A}_{11} & \boldsymbol{A}_{10} \\ & & & & & & \boldsymbol{A}_{00} \\ & & & & & & \ddots \end{bmatrix} \tag{35}$$

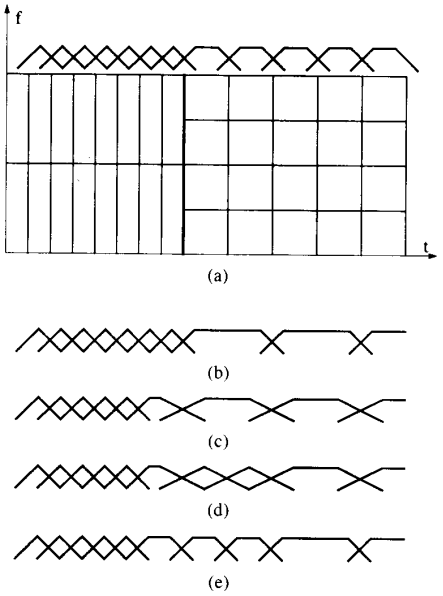


Fig. 9. (a) Switching from a 2-channel to a 4-channel modulated lapped transform. The thick line denotes the switching point. On top of the tiling, basis functions with appropriate overlaps are given. (b) Example of switching from a 2-channel to an 8-channel modulated lapped transform. Windows with appropriate overlaps are given. Direct switching according to the theorem resulting in only one degree of freedom for designing the window. (c) Use of a transition (4-channel modulated lapped transform) allowing for longer window in the 8-channel modulated lapped transform. (d) Use of transition and 4-channel modulated lapped transforms. (e) Avoiding the transition by using 4-channel modulated lapped transforms. All the windows are symmetric and have the same tails.

where the blocks A'_{00} and A'_{10} are of size $N_1 \times N_1$ and are obtained from the blocks A_{00} and A_{10} by scaling by $\sqrt{N_2/N_1}$ and retaining N_1 rows with the indexes in the following set:

$$S = \left\{ i \cdot \frac{2N_2}{N_1}, (i+1) \cdot \frac{2N_2}{N_1} - 1 \right\},$$
$$i = 0, \cdots, \frac{N_1}{2} - 1. \tag{36}$$

Therefore, if we denote by h_k, g_k the basis functions of the N_2/N_1-channel modulated lapped transforms, respectively,

$$g_k(n) = \sqrt{\frac{N_2}{N_1}}\, h_{k'}\left(n + \frac{N_2 - N_1}{2}\right) \tag{37}$$

for $n \in [0, N_1 - 1]$ and

$$g_k(n) = \sqrt{\frac{N_2}{N_1}}\, h_{k'}\left(n + \frac{3N_2 - N_1}{2}\right) \tag{38}$$

for $n \in [N_1, 2N_1 - 1]$ and $k' \in S$.

The proof that the above construction indeed leads to an orthogonal transform with overlapping basis functions, is given in the Appendix.

As a simple example, refer to Fig. 9(a), where we want to switch between a 2-channel and a 4-channel modulated lapped transform. Thus, we first shorten the window of the 4-channel modulated lapped transform to length 6 as follows:

$$w(n) = \begin{cases} 0 & n \in \{0, 7\}, \\ \sqrt{2} & n \in \{3, 4\}, \\ a & n \in \{1, 6\}, \\ \sqrt{2 - a^2} & n \in \{2, 5\}, \end{cases} \tag{39}$$

and choose $S = \{0, 3\}$. Then one obtains

$$g_k(n) = \begin{cases} \sqrt{2}\, h_{k'}(n+1) & n \in \{0, 1\}, \\ \sqrt{2}\, h_{k'}(n+5) & n \in \{2, 3\} \end{cases} \tag{40}$$

with $k = 0, 1$ and $k' = 0, 3$. Fig. 9(b)–(e) shows another simple example, that is, a switch from a 2-channel to an 8-channel modulated lapped transform. A few possibilities are explored: 1) switch directly ending up with 1 degree of freedom; 2) use a nonsymmetric 4-channel transform to obtain a transition; 3) the same as in 2) except that the transition is followed by 4-channel transforms for a while; and 4) avoid a nonsymmetric transform and use 4-channel transforms instead.

D. Discussion

We have shown how to construct bases which can be seen as duals both of wavelets and wavelet packets. This was achieved using time-varying modulated lapped transforms, with both overlapping and nonoverlapping basis functions. The constructions we gave are, however, more restricted than in the wavelet packets case. In particular, the filter length (basis functions) is restricted to twice the number of channels. On the other hand, the advantage of these time-varying modulated lapped transforms is the existence of a fast algorithm together with the fact that all the filters involved are obtained by modulation.

V. Fast Algorithms for Arbitrary Time-Frequency Tilings

We have so far considered the construction of time-varying bases. An obvious remaining question is how we may find the "best basis" [5] for a given input signal, i.e., the orthonormal expansion which best matches the signal's characteristics. This question has been studied for wavelet packet bases [5], [7]. However, the "best basis" of [5] gives the "best" (binary) tiling for the *unsegmented* signal. Here, we seek to generalize this to *jointly* find the best split in time and in frequency. It must be noted that the tasks of finding the best signal time split and the best frequency split for each time segment are not sequential operations, and are found jointly, using the *double-tree* fast algorithm, to be described in Section V-E.

This algorithm, like the "best basis" scheme of [5], relies on the orthogonality of the decomposition basis sets for its optimality. This was a major motivation for the

construction of orthogonal boundary filters in the previous section. It is important to emphasize that, although we resort to a rate-distortion measure here, the double-tree algorithm is valid for *any additive cost measure* over the set of sequences (as is the "best basis" method of [5]). Before we describe the double-tree algorithm we will first try to motivate the relevance of the R–D cost criterion we use in this work (as in [7]), and briefly describe the best basis algorithm using this measure, and show its use in finding adaptive modulated lapped transform (MLT) decompositions.

A. Motivation for the Rate–Distortion Measure

Among the various cost measures that one can pick for finding adaptive time–frequency decompositions, we pick rate–distortion. The benefits of this are twofold. Firstly, since the rate–distortion (R–D) measure is two-sided, it is more general than one-sided measures like entropy only (as used in [5]) or distortion only. In fact, these are special cases of the more general R–D measure, corresponding to the endpoints of the operational R–D characteristic, which can be generated *in its entirety* if a two-sided measure were employed. Secondly, the R–D measure is the most sensible for compression applications.

We pick that basis (from the entire wavelet packet family of orthonormal bases) which minimizes a combination of: 1) the number of bits (or rate R) required to represent the signal; and 2) the distortion (D) representing the squared error between the original signal and the approximation using R bits. The trade-off between rate and distortion is given by the R–D function, which represents the minimum achievable distortion for a maximum target rate or bit budget, or conversely the minimum rate needed to achieve a maximum tolerable distortion level. Note that each combination of a basis and quantization choice represents a single R–D operating point. Thus the constrained optimization problem of seeking the best basis in an R–D sense involves the search for that basis which minimizes the average distortion for a target average bit rate. This "hard" constrained problem is converted to an "easy" equivalent unconstrained problem by "merging" rate and distortion through the minimization of the Lagrangian cost function defined as

$$J(\lambda) = D + \lambda R. \tag{41}$$

For example, if R represents the entropy of the signal, and D the mean-squared-error, then the entropy-only and mean-squared-error-only cost cost measures of [5] become special cases of this more general Lagrangian cost measure corresponding to $\lambda = \infty$ and $\lambda = 0$, respectively. We now summarize the solution to (41), briefly outline the R–D optimal wavelet packet construction [7], and then present its generalization to time-varying bases.

B. Rate–Distortion Optimality

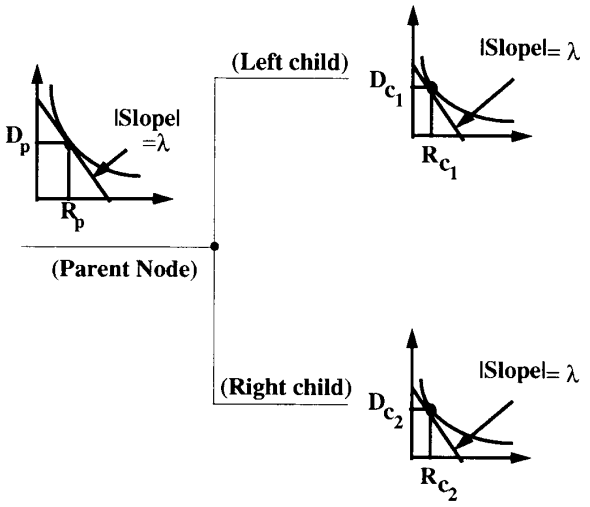
We consider a wavelet packet framework with independent coding of the nodes of the wavelet packet tree and a mean squared error (MSE) distortion metric. Due to additivity of the rate and distortion measures over tree nodes, the quantizer search for the nodes can be done independently using the Lagrangian cost measure of (41). It can be shown that at R–D optimality, all nodes *must operate at a constant slope point* λ *on their R–D curves* [23].

The intuition for this result can be seen from the argument that at optimality, each allocated bit must do an equal amount of "good" (in the distortion sense), for otherwise one could redistribute bits from "unprofitable" to "profitable" elements. It is obviously cost-effective to keep doing this until an optimum is reached, where no element can spend the next available bit any more profitably than any other, i.e., until all subsignals have the same slope (λ) on their rate–distortion function. The desired optimal constant slope value λ^* is not known *a priori* and depends on the particular target budget or quality constraint. Fortunately, however, λ^* can be obtained relatively painlessly via a fast convex recursion in λ using the bisection algorithm [7].

C. "Best Basis" Wavelet Packet: The Single-Tree Algorithm

We now summarize the best wavelet packet basis search in a rate–distortion sense, described in [7] as an extension of the one-sided entropy only or distortion only criterion used in [5]. The problem is to determine the optimal combination of *best basis* wavelet packet subtree (from among the entire basis family for a given wavelet set of filters) and its *best quantizer* (from among the admissible set of quantizers). This problem is solved for arbitrary coding scenarios (within the constraints of independent coding of the wavelet packet tree nodes and an additive distortion measure) with arbitrary quantizers assumed at each node.

As a first step to finding a fast algorithm, the constrained R–D problem is transformed to an unconstrained one using the Lagrange multiplier λ. Then, a fast dynamic programming algorithm is employed to find the best basis/best quantizer choice for the fixed λ. Finally, the optimal value of λ for the given budget is solved. The first step involves converting all (R, D) points of the full-depth wavelet packet tree to their associated Lagrangian costs $J = D + \lambda R$. The second step involves: 1) populating each internal wavelet packet tree node with the minimum (over all quantization choices) Lagrangian cost $J(node) = D(node) + \lambda R(node)$ to get *the best quantizer* at each node; and 2) "pruning" the full-depth tree (using a fast algorithm based on Bellman's optimality principle) into that subtree which has minimum total sum-of-leaves cost, to get *the best basis*. The pruning criterion applied at each node is that of deciding in favor of the parent or its children based on which has the lower Lagrangian cost (for the fixed quality factor λ). See Fig. 10. Finally, the "correct" value of λ, λ^*, which results in the desired convex-hull operating point for the target bit budget of the original constrained problem is found by a fast convex recursion in λ.



For "quality factor" λ, prune if

$$(D_{c_1}+D_{c_2}) + \lambda(R_{c_1}+R_{c_2}) > (D_p + \lambda R_p)$$

Fig. 10. Lagrangian cost pruning criterion for "quality criterion" λ for each parent node of the wavelet packet tree. This condition is used recursively to do fast pruning from the complete tree depth towards the root to find the optimal subtree for a given λ.

The best basis wavelet packet construction summarized above will be referred to as *the single-tree algorithm* because a single tree is pruned into the R–D optimal wavelet packet basis subtree.

D. Optimal Time-Varying Modulated Lapped Transforms

We now consider tiling using the time-varying modulated lapped transforms designed in Section IV retaining the constraint of seeking R–D optimality. We will consider two cases: 1) the special case where the tails of the prototype window have the same "slope" in the transition region between the different size trees [see Fig. 9(e)]; and 2) the general case where the tails for the different size MLT's have arbitrary slopes [see Fig. 9(d)].

1) See Fig. 9(e) for the case where the tails of the prototype windows of the MLT filter-bank transition region have the same shape or slope. Fig. 11 shows a family tree of such orthogonal windows, where the different tree levels correspond to different size MLT's. Due to the independence in the optimal orthonormal decomposition splits for adjacent signal segments of the MLT tree, it is obvious that this MLT tree is the time-domain dual of the wavelet packet tree described earlier and can therefore be pruned optimally using the single-tree algorithm. To be specific, in Fig. 11, the decision on whether or not to split node B1 into nodes C1 and C2 is independent of the optimal split/merge decision for node B2, *due to the constant shape property* of the transition tails. The single-tree algorithm is thus optimal.

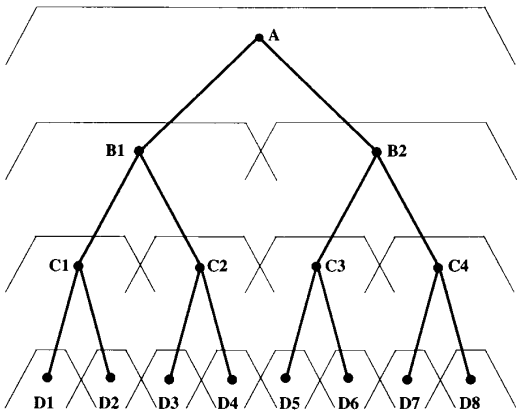


Fig. 11. Variable-size modulated lapped transform tree of adjacent orthogonal windows where the slopes of the window tails are constant. The optimal tree rooted at *B*1 is obtained independently of the optimal tree rooted at *B*2 and the single tree algorithm is therefore optimal here.

2) See Fig. 9(d) for the more general case where arbitrary tail shapes are permitted. Here, changing between MLT trees becomes a "dependent" problem. Fig. 12 shows an example of changing at time instant T from an 8-channel MLT to both a 16-channel MLT (with a more "gradual" slope to mitigate segmentation discontinuity) and a "sharper" 4-channel bank. Here, clearly the single-tree wavelet packet algorithm does not work due to the loss of independence between signal decompositions before and after the transition boundary, i.e., one cannot find an optimal split for the segment to the left of T without knowing the optimal split to its right, and the optimal algorithm becomes a difficult dependent problem for the general case. Note, however, that when the transition boundary between MLT's is a negligible proportion of the signal (as is likely for most cases), then the boundary effect which causes loss of independence (between points T_1 and T_2 in Fig. 12) becomes negligible. Thus, barring the small disturbance effect at the transition boundaries, *the single-tree algorithm would be nearly always optimal.*

E. Optimal Arbitrary Binary Tiling: The Double-Tree Algorithm

We now proceed to find the R–D optimal time-varying binary tiling of the time-frequency plane, obviously desirable from a coding perspective. We emphasize that the framework in which we seek optimality is dependent on the coding environment. Thus, the optimal tiling for one application corresponding to a particular choice of a wavelet filter set, quantizer set, and compression scheme need not be optimal for another. We solve the arbitrary tiling problem by extending the fast wavelet packet single-tree algorithm to a *double-tree* algorithm. This finds the optimal binary segmentation of the input signal, and the optimal wavelet packet tree to use over each segment. The time split and the frequency split are found *jointly*,

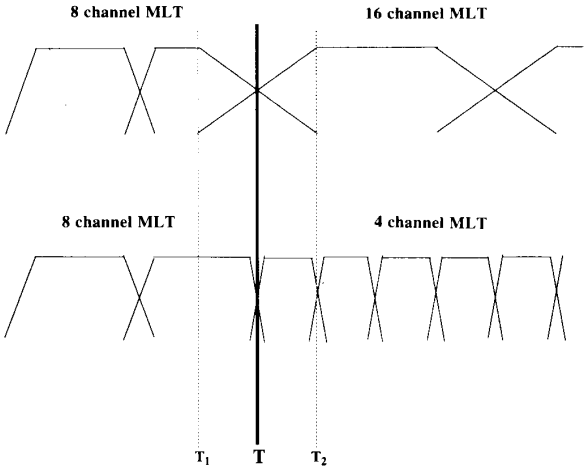


Fig. 12. Transition between different size MLT-trees where the tails of the transition windows do not have the same shape. The split to the left of transition time T is no longer independent of the split to its right, as the MLT size (4 or 16 channel) to the right of T affects the split to its left (8 channel). However, barring a small disturbance effect between times T_1 and T_2, the single tree algorithm is nearly optimal.

so this is not a sequential optimization. Within the constraints of the tree structure imposed by the chosen set of wavelet filters and their associated boundary filters designed in Section III, this gives us, for a given signal and set of quantizers, the *optimal tiling of the time-frequency plane* in the R–D sense.

Although we summarize the flowchart of the double-tree algorithm later, the basic idea is easiest explained through a "toy" example. See Fig. 13. Assume a length-4 input signal [1, 2, 3, 4] and a Haar basis as the wavelet filter set (i.e., $\{[1/\sqrt{(2)} \quad 1/\sqrt{(2)}], [1/\sqrt{(2)} \quad -1/\sqrt{(2)}]\}$). Since no boundary filters are required for the Haar basis, the analysis is the simplest possible. To find the optimal split, optimal wavelet packet subtrees are found for all possible binary signal subsets: {[1, 2, 3, 4], [1, 2], [3, 4], [1], [2], [3], [4]} as shown in Fig. 13(a). A scalar quantizer of step size 4 has been picked for this example to quantize all wavelet packet coefficients. As before, a Lagrangian cost criterion corresponding to λ is used for the optimal tree pruning operation. Then, the costs associated with the best bases determined in the first step are used to populate a second tree called *the splitting tree* as shown in Fig. 13(b). The root of the splitting tree is populated with the cost associated with the best basis wavelet packet for the [1, 2, 3, 4] signal split, the first tree level with the two costs corresponding to the [1, 2] and [3, 4] splits, respectively, etc. The splitting tree is pruned using the same single-tree fast algorithm as that used to find the best bases whose costs populate its nodes.

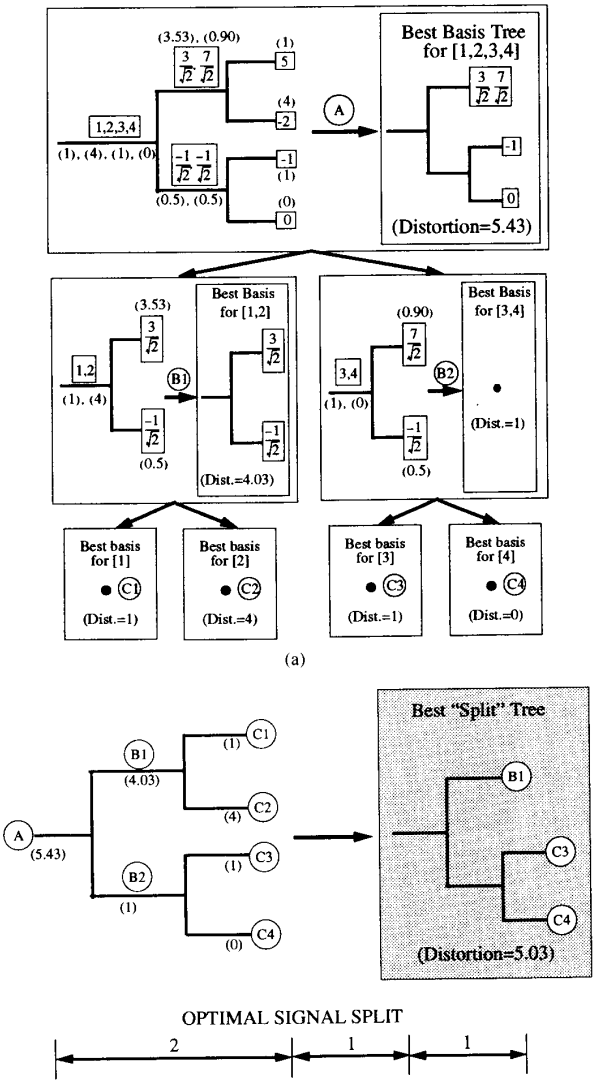


Fig. 13. The double-tree optimal orthonormal splitting algorithm on the input signal (1, 2, 3, 4) in R^4 for the Haar kernel and a scalar quantizer of step size 4. Lagrangian costs are shown in brackets ($\lambda = 0$ used here). (a) The best basis wavelet packet subtrees corresponding to all feasible signal subsets. (b) The splitting tree whose nodes are populated from the best basis cost of (a).

F. Complete Tiling Algorithm

Having established the main idea, we now summarize the complete algorithm. As in [7], this will be done in two phases. First, the optimal algorithm for a given operating slope λ will be given, followed by a summary of how to hunt for the optimal operating slope λ^*.

1) Initialization: Prior to the "pruning" operation, a single fixed cost of gathering the statistics enlisted in *Step 0* below must be paid.

Step 0: Generate the wavelet packet coefficients and the quantizer set dependent (R, D) values for the complete tree for *each binary subset* of the signal to be coded. Thus, for a length-N signal, this entails growing one length-N wavelet packet basis tree, two length-$N/2$ trees, four

length-$N/4$ trees, etc., up to the desired maximum tree depth. Note that the operation of populating the wavelet packet trees for the binary subsets of the signal uses the appropriate boundary filters (as designed in Section III) between signal segment boundaries. This guarantees orthonormality of the bases, a requirement of the fast pruning algorithm.

2) Phase I: Optimality for a Fixed Operating Slope: Phase I of the algorithm is run for a fixed slope value of λ, and could be considered a subroutine called by Phase II.

Step 1: For the λ of the current iteration, solve the wavelet packet problem for each binary subset of the signal to be coded; i.e., *run the single-tree algorithm on each binary subset*. This requires finding the minimal Lagrangian costs associated with the best basis/quantizer choice for each of the halves, quarters, eighths, sixteenths, etc., of the original signal.

Step 2: Populate the global splitting tree with the minimal Lagrangian costs obtained from Step 1 for each binary signal segment (see Fig. 13 of the toy example).

Step 3: Prune the splitting tree of Step 2 exactly as was done for each tree in Step 1, i.e., using the single-tree pruning algorithm [7]. This gives the "best binary split" tree for the signal (see Fig. 13 of the toy example). For the picked λ, we now have the R–D optimal binary tiling of the time–frequency plane for the given signal and coding parameters.

3) Phase II: Iterating Towards the Optimal Operating Point: The optimal value of λ, λ^*, which solves the given budget constraint R_{budget} is found using the convex-search bisection algorithm as described in [7]. The basic idea is that starting from a known initial interval engulfing the desired operating slope, the search intervals are made successively smaller (exploiting the convex relationship of both global rate and global distortion with respect to the operating slope λ [23]) until convergence is achieved. If $\lambda_l^{(i)}$ and $\lambda_u^{(i)}$ are the lower and upper bounds to λ^* at iteration i, then the convexity property is exploited in tightening either the upper or the lower bound at the $(i+1)$th iteration to $\lambda^{(i+1)} = |[(\Delta D/\Delta R)]^{(i)}|$, where $[(\Delta D/\Delta R)]^{(i)}$ (the ratio of the difference in distortion and rate associated with the slopes $\lambda_l^{(i)}$ and $\lambda_u^{(i)}$ of the ith iteration) provides a closer approximation to λ^* than available at the ith iteration. Phase I of the algorithm is thus called recursively for converging values of λ in Phase II, and λ^* solves the given budget constrained problem to within a convex hull approximation.

G. Complexity

We now address the complexity of the double-tree algorithm. As a prelude, it is important to note that the complexity of the single-tree wavelet packet algorithm for a signal of size N (assumed to be a power of 2) is $O(N \log N)$. This follows from the complexity being of $O(N)$ at each level of the wavelet packet tree, and there being $\log N$ levels in the complete tree. Now, consider the double-tree with the splitting tree grown to the complete depth of $\log N$. The splitting tree is populated with the results of the wavelet packet single trees grown on each of its nodes. A single wavelet packet tree of length N is grown at the root [of complexity $O(N \log N)$], two $N/2$-size trees are grown at depth 1 [of complexity $O(2 * (N/2) \log (N/2)) = O(N \log (N/2))$], four $N/4$-size trees [of complexity $O(N \log (N/4))$] at depth 2, etc. Thus, the total complexity is bounded by $O(N \log N)$ at each level of the splitting tree, to give a total complexity of $O(N(\log N)^2)$.

Thus, the increase in complexity between the single wavelet packet tree algorithm and the double-tree adaptive wavelet packet algorithm is from $O(N \log N)$ to $O(N(\log N)^2)$. This implies that if one considers nonoverlapping 1024-point segments of an input signal, for an order of magnitude increase in computational complexity over that of the wavelet packet best basis single-tree algorithm, one can find the optimal adaptive wavelet packet tree using the double-tree algorithm.

VI. Experimental Results

The single-tree algorithm was used to find the optimal time-varying modulated lapped transform decomposition. A description of the experiments involving time-varying MLT's follows next.

A. MLT Experiments Using the Single-Tree Algorithm

The single-tree algorithm was used to find the optimal tiling for a synthetic 512 point signal consisting of a cascade of a 256-point first-order autoregressive signal of variance 10 and a correlation coefficient of 0.1 (i.e., nearly uncorrelated) and a 256-point signal with a correlation of 0.9 (i.e., highly correlated). See Fig. 14(a). The single-tree algorithm was used to select adaptively from MLT filter banks of sizes 8, 16, and 32, with the switching between banks being done so as to preserve orthonormality, as explained in Section IV. Fig. 14(b) shows the time–frequency tiling of the signal using this time-varying MLT decomposition for a quantizer step size of 3 and $\lambda = 0$. As expected, the first part of the signal gets finer time-resolution, while the latter part gets better frequency-resolution. Note how the maximum frequency resolution (afforded by the 32-channel MLT bank) is picked for the latter half of the signal, which is more "stationary" than the first half.

B. Adaptive Wavelet Packet Experiments Using the Double-Tree Algorithm

The double-tree algorithm was used to find the optimal tiling for several test signals. As expected, it beats the single-tree wavelet packet algorithm (of which it is a generalization) for many signals, both real and synthetic. We must emphasize the sensitivity (in terms of the picked segmentation and trees for each segment) of the algorithm to such parameters as the quantization step size, the smallest signal split desired, and the coding scheme (fixed

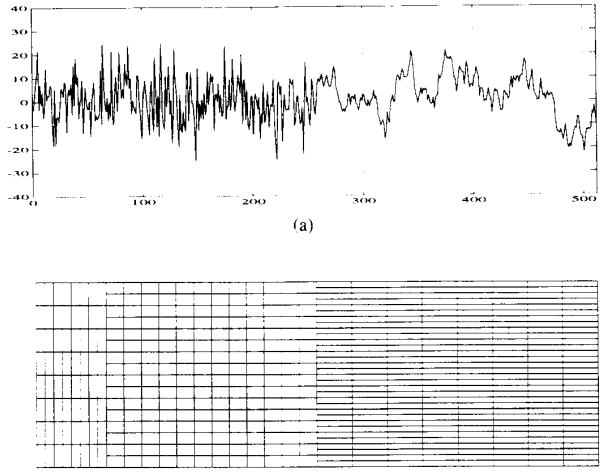


Fig. 14. Optimal tiling using adaptive modulated lapped transforms for a synthetic 512-point test signal: (a) Input signal consisting of two parts: the first half is an autoregressive first-order Markov sequence with variance = 10 and $\rho = 0.1$, while the second half has $\rho = 0.9$. (b) Optimal tiling using adaptive MLT banks of sizes 8, 16, and 32. Switching between MLT banks is accomplished as explained in Section IV. Note, how the second half of the signal, which is more "stationary" gets better frequency resolution, while the first part, which is more uncorrelated gets better time resolution. (Shown for a quantizer step size of 3 and $\lambda = 0$).

bit rate or entropy coded). Also, the time-variance of the decomposition will be highlighted, as evidenced by the algorithm's sensitivity to phase shifts of the input signal. In the experiments that follow, uniform scalar quantizers were used along with first-order entropy and MSE as the rate and distortion measures for all cases. The Daubechies length-4 filter (called D_2 filter) and its boundary filters were used as the wavelet packet tree kernel. Note that the overhead of sending the splitting map has been included in the results.

1) Speech Input: The optimal split for a 512-point speech segment to a maximum tree depth of 7 (i.e., leaf size of 8) is shown in Fig. 15. A scalar quantizer of step size 20 was used. The optimal split using the double-tree algorithm achieves 1.99 bits/sample with 27.98 MSE, while the best basis wavelet packet (single) tree algorithm needs 2.44 bits/sample with 28.33 MSE, highlighting the usefulness of efficient binary tiling.

2) Synthetic Signal: Sum of an Impulse and a Sinusoid: Fig. 16(a) shows the second test input consisting of an impulse and a sinusoid, with Fig. 16(b) and (c) showing the optimal tiling representations for the double-tree and the single-tree algorithms, respectively. A scalar quantizer of step size 0.1 is used (to ensure fine quantization) with the tree grown to a smallest split of 16, for $\lambda = 1$. As seen from Fig. 16(b), the double-tree split "adapts" well to the input signal, finding a split with good time resolution around the impulse, and good frequency resolution around the sinusoid. The wavelet packet single-tree tiling of Fig. 16(c), on the other hand, fails to isolate the impulse.

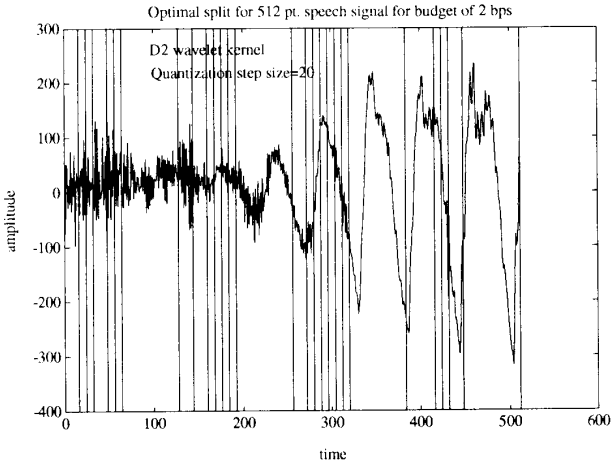


Fig. 15. Optimal split of a 512-point speech signal for the Daubechies D_2 filter, a scalar quantizer of step size 20, and maximum tree depth of 7. First-order entropy and MSE are used as the rate and distortion measures.

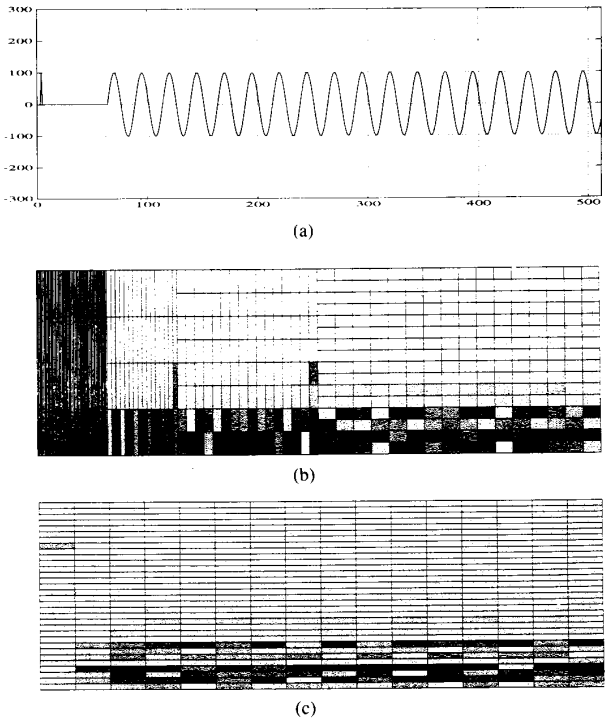


Fig. 16. Optimal tiling for a synthetic test signal consisting of a Dirac and a sinusoid: (a) Input signal with time impulse at $t = 4$, and sinusoid starting at $t = 65$. (b) Optimal split using the double-tree algorithm. Note how both the Dirac in time and Dirac in frequency are detected with good resolution. (c) Optimal single tree (WP) split. Note how the impulse in time is not detected. (Shown for the Daubechies D_2 filter, quantizer step size of 0.1 (fine quantization), $\lambda = 1$, and the tree grown to a maximum depth of 6, i.e., to a leaf size of 16. First-order entropy and MSE are used as the rate and distortion measures.)

3) Basis Signal Input: The wavelet basis function (of nonzero length 22) from a tree depth of 3 derived from the length-4 Daubechies D_2 filter set is analyzed next (see

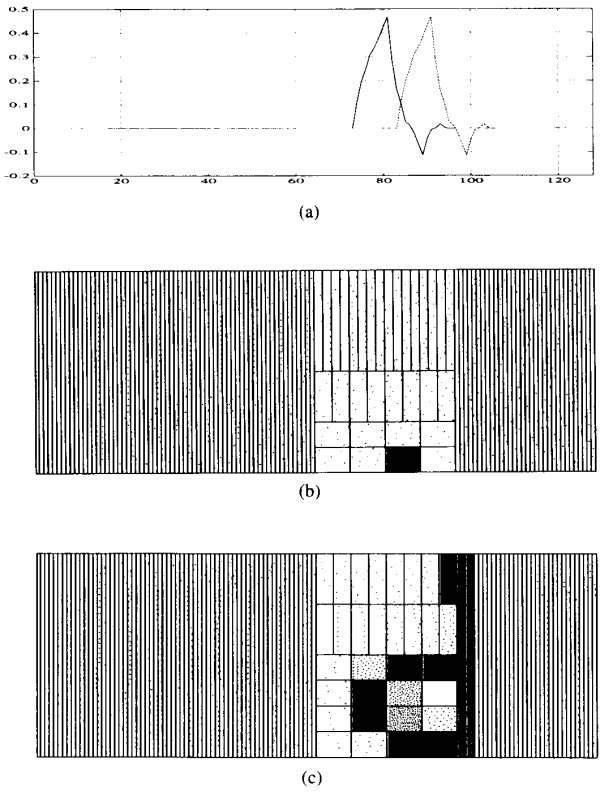


Fig. 17. Time variance of the double-tree algorithm shown for basis input signal: (a) Synthetic test input signals consisting of different phases of the depth-3 wavelet basis of the Daubechies D_2 filter: i) "in-phase" ($t = 74$ to $t = 95$) and ii) "out-of-phase" ($t = 84$ to $t = 105$). (b) Tiling of signal i). Note how the wavelet basis is located. (c) Tiling of signal ii). Note that the wavelet tiling is lost.

Fig. 17). This experiment highlights the sensitivity of the algorithm to the phase of the input signal. When the input basis function is positioned at the correct phase (starting at $t = (k * 8 + 2)$ for positive integer values of k), the optimal split is verified to correspond to the depth-3 wavelet basis function with a single nonzero wavelet packet coefficient. Fig. 17(b) depicts how the algorithm "finds" the input basis signal corresponding to the in-phase signal i) of Fig. 17(a) (corresponding to $t = 74$). The time variance of the algorithm should be evident from Fig. 17(c) corresponding to the split for the same signal shifted by 10 samples (signal ii), starting at $t = 84$). This phase-sensitivity is common to bases which are not shift-invariant.

4) Discussion: The segmentation in time performed by our adaptive wavelet packet construction makes it less sensitive to the phase of the signal than wavelets or wavelet packets, as borne out by experiments. For example, in the impulse and sinusoid signal example of Section VI-B-2, we found that the adaptive wavelet packet construction tracked the phase of the impulse with remarkable precision when it was changed, something that the wavelet packet single-tree algorithm failed to do.

Appendix
Time-Varying Modulated Lapped Transforms: Proofs

A. Proof of (29)

To prove (29), denote by $\boldsymbol{a}_i$ the ith row of $\boldsymbol{A}_0$. Then,

$$\begin{aligned}\boldsymbol{a}_i\boldsymbol{a}_j^T &= \frac{1}{N}\sum_{n=N}^{2N-1} w_A^2(n)\cdot\cos\left(\frac{2i+1}{4N}(2n-N+1)\pi\right)\\ &\quad\cdot\cos\left(\frac{2j+1}{4N}(2n-N+1)\pi\right)\\ &= \frac{1}{N}\sum_{n=N}^{(3/2)N-1}\left[w_A^2(n)\cdot\cos\left(\frac{2i+1}{4N}(2n-N+1)\pi\right)\right.\\ &\quad\cdot\cos\left(\frac{2j+1}{4N}(2n-N+1)\pi\right)\\ &\quad+(2-w_A^2(n))\cdot\cos\left(\frac{2i+1}{4N}(5N-2n-1)\pi\right)\\ &\quad\left.\cdot\cos\left(\frac{2j+1}{4N}(5N-2n-1)\pi\right)\right]\\ &= \frac{2}{N}\sum_{n=N}^{(3/2)N-1}\cos\left(\frac{2i+1}{4N}(2n-N+1)\pi\right)\\ &\quad\cdot\cos\left(\frac{2j+1}{4N}(2n-N+1)\pi\right)\end{aligned}$$

where we have used (26), and the fact that

$$\begin{aligned}&\cos\left(\frac{2i+1}{4N}(5N-2n-1)\pi\right)\\ &\quad=\cos\left(\frac{2i+1}{4N}(2n-N+1)\pi-(2i+1)\pi\right)\\ &\quad=\cos\left(\frac{2i+1}{4N}(2n-N+1)\pi\right).\end{aligned}$$

It is thus obvious that $\boldsymbol{a}_i\boldsymbol{a}_j^T$ does not depend on the window and is the same for any set of modulated lapped transform filters.

B. Construction of Overlapping Modulated Lapped Transforms

To check that the construction is valid, one has first to prove that the orthogonality of tails for the N_1-channel bank holds. To prove that, remember that the N_2-channel bank is perfect reconstruction by assumption, and thus, orthogonality of its tails holds, together with the fact that all of its basis functions are unitary. The tails in the N_1-channel case are obtained from the tails of the N_2-channel bank, and thus, by construction, they will be orthogonal to each other (since the tails in the N_2-channel bank are). Also by construction, the overlapping tails of the N_1- and N_2-channel banks will be orthogonal. The two facts left to show are that the resulting vectors from the N_1-channel modulated lapped transform are unitary, as well as that they are mutually orthogonal. Call the vectors

from the N_1-channel modulated lapped transform, $\boldsymbol{g}_k$, according to (37) and (38). Then

$$\boldsymbol{g}_k^T \cdot \boldsymbol{g}_k = \frac{N_2}{N_1}\left(\boldsymbol{h}_{k'}^T \cdot \boldsymbol{h}_{k'} - \sum_{n=(N_1+N_2)/2}^{(3N_2-N_1)/2} h_{k'}^2(n)\right) \quad (42)$$

where $\boldsymbol{h}_{k'}$ is the vector from the N_1-channel bank corresponding to the one from the N_1-channel one, and, since N_2-channel bank is perfect reconstruction $\boldsymbol{h}_{k'}^T \cdot \boldsymbol{h}_{k'} = 1$. The sum term in parentheses is

$$\sum_{n=(N_1+N_2)/2}^{(3N_2-N_1)/2} h_{k'}^2(n)$$

$$= \frac{2}{N_2}\sum_{n=(N_1+N_2)/2}^{(3N_2-N_1)/2} \cos^2\underbrace{\left(\frac{2k+1}{4N_2}\pi(2n - N_2 + 1)\right)}_{x}, \quad (43)$$

$$= \frac{2}{N_2}\sum_{n=(N_1+N_2)/2}^{N_2-1}\left(\cos^2(x) + \cos^2 \cdot \left(\frac{2k+1}{2}\pi - x\right)\right), \quad (44)$$

$$= \frac{2}{N_2}\sum_{n=(N_1+N_2)/2}^{N_2-1}(\cos^2(x) + \sin^2(x)), \quad (45)$$

$$= \frac{N_2 - N_1}{N_2}. \quad (46)$$

Substituting this into (42), we obtain

$$\boldsymbol{g}_k^T \cdot \boldsymbol{g}_k = \frac{N_2}{N_1}\left(1 - \frac{N_2 - N_1}{N_2}\right) = 1 \quad (47)$$

and thus, the vectors are unitary. To prove that they are mutually orthogonal, one has to form all the products $\boldsymbol{g}_i^T \cdot \boldsymbol{g}_j$, $i \neq j$. After some manipulations, and with indices in the range as given in (36), it can be shown that all of these products are zero. □

REFERENCES

[1] I. Daubechies, "Ten lectures on wavelets," *SIAM*, 1992.

[2] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 11, pp. 674–693, July 1989.

[3] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Mag.*, vol. 8, pp. 14–38, Oct. 1991.

[4] A. Papoulis, *The Fourier Integral and its Applications*. New York: McGraw-Hill, 1962.

[5] R. Coifman, Y. Meyer, and V. Wickerhauser, "Wavelet analysis and signal processing," M. B. Ruskai *et al.*, Ed. in *Wavelets and their Applications*, Boston, MA: Jones and Bartlett, 1992, pp. 153–178.

[6] M. V. Wickerhauser, "INRIA lectures on wavelet packet algorithms," Dep. Math., Yale Univ., Tech. Rep., Mar. 1991.

[7] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–173, Apr. 1993.

[8] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Arbitrary orthogonal tilings of the time-frequency plane," in *Int. Symp. on Time-Frequency and Time-Scale Analysis*, Victoria, BC, Oct. 1992, pp. 11–14.

[9] M. J. T. Smith and T. P. Barnwell III, "Exact reconstruction for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 434–441, June 1986.

[10] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[11] M. Vetterli and D. Le Gall, "Perfect reconstruction FIR filter banks: Some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1057–1071, July 1989.

[12] K. Nayebi, T. P. Barnwell III, and M. J. T. Smith, "Analysis-synthesis systems with time-varying filter bank structures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, Mar. 1992, pp. 617–620.

[13] A. Cohen, I. Daubechies, and P. Vial, "Wavelets on the interval and fast algorithms," *J. Appl. Comput. Harmonic Anal.*, submitted 1992.

[14] K. Asai, K. Ramchandran, and M. Vetterli, "Image representation using time-varying wavelet packets, spatial segmentation and quantization," in *CISS*, Baltimore, MD, Mar. 1993.

[15] C. Herley and M. Vetterli, "Orthogonal time-varying filter banks and wavelets," in *Proc. ISCS*, Chicago, IL, vol. 1, May 1993, pp. 391–394.

[16] —, "Orthogonal time-varying filter banks and wavelet packets," *IEEE Trans. Signal Processing*, submitted 1993.

[17] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207–2232, Sept. 1992.

[18] K. Nayebi, T. P. Barnwell III, and M. J. T. Smith, "Time domain filter bank analysis: A new design theory," *IEEE Trans. Signal Processing*, vol. 40, pp. 1414–1429, June 1992.

[19] R. de Queiroz, "Subband processing of finite length signals without border distortions," in *Proc. ICASSP*, San Francisco, CA, May 1992, pp. 613–616.

[20] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. XLI, pp. 909–996, 1988.

[21] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech, 1992.

[22] J. Princen, A. Johnson, and A. Bradley, "Subband transform coding using filter bank designs based on time domain aliasing cancellation," in *Proc. ICASSP*, Dallas TX, Apr. 1987, pp. 2161–2164.

[23] Y. Shoham and A. Gersho, "Efficient allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1445–1453, 1988.

[24] R. L. de Queiroz and K. R. Rao, "Time-varying lapped transforms and wavelet packets," *IEEE Trans. Signal Processing*, this issue, pp. 3293–3305.

Cormac Herley, for a photograph and biography, see p. 2556 of the August 1993 issue of this TRANSACTIONS.

Jelena Kovačević (S'88–M'91), for a photograph and biography, see p. 2066 of the June 1993 issue of this TRANSACTIONS.

Kannan Ramchandran, for a photograph and biography, see p. 174 of the April 1993 issue of IEEE TRANSACTIONS ON IMAGE PROCESSING.

Martin Vetterli (S'86–M'86–SM'90), for a photograph and biography, see p. 2556 of the August 1993 issue of this TRANSACTIONS.