Orthogonal Time-Varying Filter Banks and Wavelet Packets

Cormac Herley, *Member IEEE,* and Martin Vetterli, *Senior Member, IEEE*

Abstract— **We consider the construction of orthogonal time-varying filter banks. By examining the time domain description of the two-channel orthogonal filter bank we find it possible to construct a set of orthogonal boundary filters, which allows us to apply the filter bank to one-sided or finite-length signals, without redundancy or distortion. The method is constructive and complete. There is a whole space of orthogonal boundary solutions, and there is considerable freedom for optimization. This may be used to generate subband tree structures where the tree varies over time, and to change between different filter sets. We also show that the iteration of discrete-time time-varying filter banks gives continuous-time bases, just as in the stationary case. This gives rise to wavelet, or wavelet packet, bases for half-line and interval regions.**

I. Introduction

THE subject of two-channel perfect reconstruction filter banks in the case where the filters have finite impulse responses has been extensively studied, and such structures are widely used in subband coding applications. Their design is by now well understood in the case where we treat infinite signals and do not wish to vary the analysis over time. The case where we process finite length signals, or wish to vary the analysis, requires special treatment. Particular solutions of the finite length problem have appeared, but analysis of the time-varying case has only begun recently. The purpose of this paper is to address these problems in the case where orthogonal filter banks are involved. In particular, we show that, while stationary filter bank designs are based on polynomial algebra methods, these do not easily extend to the time-varying case. However, using simple matrix algebra operations on the time domain operator description of the filter bank, solutions are very easily found, and, further, we show that they are constructive and complete.

Consider then the system depicted in Fig. 1, the basic two-channel filter bank, which is called perfectly reconstructing if $\hat{x}$ is identically equal to the input. We will assume that the reader has some familiarity with two-channel filter banks. Suitable references are [1]–[4]. There are many ways of looking at the design problem and expressing the

Manuscript received April 27, 1993; revised January 10, 1994. This work was supported by the National Science Foundation under Grants ECD-88-11111 and MIP-90-14189. The associate editor coordinating the review of this paper and approving it for publication was Prof. Henrik V. Sorensen.

C. Herley was with the Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York NY; he is now with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA.

M. Vetterli was with the Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York NY; he is now with the Electrical Engineering and Computer Science Department, University of California, Berkeley, CA 94720 USA.

IEEE Log Number 9403756.

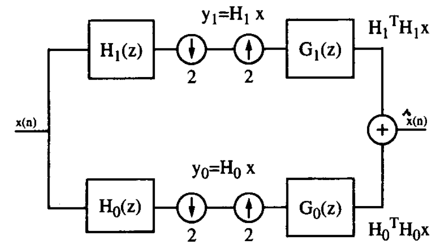


Fig. 1. Maximally decimated two-channel multirate filter bank.

solution. A very popular solution, first given in [1] and [5], is that where $H_1(z) = z^{-(N-1)}H_0(-z^{-1})$, and $G_0(z) = H_0(z^{-1}), G_1(z) = H_1(z^{-1})$, where N is the filter length. This is known as the orthogonal solution, since it can be easily verified that

$$\langle h_1(n), h_0(n-2k)\rangle = 0 \tag{1}$$

$$\langle h_0(n), h_0(n-2k)\rangle = \delta_k = \langle h_1(n), h_1(n-2k)\rangle \tag{2}$$

i.e., the filter impulse responses are orthogonal with respect to even shifts.

The action of the filter bank on an infinite signal column vector x can be represented using the operators in (3) (at the top of the next page) and $\mathbf{H}_1$, which is defined in a similar fashion to $\mathbf{H}_0$, using the filter $h_1(n)$ in place of $h_0(n)$. In this notation, instead of writing filters in terms of their transfer functions or z-transforms, we write their shifted impulse responses as rows of a matrix. These row vectors are actually the basis functions for a linear expansion of the signal. While z-transforms are the natural tool for handling linear time-invariant systems, we will find the matrix representation more useful for analysis of time-varying systems.

Clearly, premultiplication by $\mathbf{H}_0$ has the effect of filtering the signal by $H_0(z)$ and subsampling by 2 (represented by the shift by 2 in the matrix (3)). The analysis filter outputs are hence represented by $\mathbf{y}_0 = \mathbf{H}_0 \cdot \mathbf{x}$ and $\mathbf{y}_1 = \mathbf{H}_1 \cdot \mathbf{x}$, as indicated in Fig. 1. Because of the time reversal, the synthesis filters are represented by $\mathbf{H}_0^*$ and $\mathbf{H}_1^*$, where $*$ denotes Hermitian transpose. In all cases we shall be interested only in filters with real coefficients so that $\mathbf{H}_0^* = \mathbf{H}_0^T$. That the system gives perfect reconstruction is shown by the fact that the sum of the two paths is an identity

$$\mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1 = \mathbf{I}. \tag{4}$$

The orthogonality properties above in the time domain imply orthogonality properties of the matrices. From (1) it follows

$$\mathbf{H}_0 = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \ddots & h_0(0) & \cdots & \cdots & h_0(N-2) & h_0(N-1) & 0 & 0 & \cdots & \ddots \\ \ddots & 0 & 0 & h_0(0) & \cdots & h_0(2) & h_0(N-2) & h_0(N-1) & \cdots & \ddots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3)$$

that

$$\mathbf{H}_0 \mathbf{H}_1^T = \mathbf{0} \quad (5)$$

and from (2) that

$$\mathbf{H}_0 \mathbf{H}_0^T = \mathbf{I} = \mathbf{H}_1 \mathbf{H}_1^T. \quad (6)$$

An equivalent representation can be found by interleaving the rows of $\mathbf{H}_0$ and $\mathbf{H}_1$ (see the bottom of this page). Note that we have used the relation $H_1(z) = z^{-(N-1)} H_0(-z^{-1})$ in writing the coefficients of the filter $H_1(z)$ in the matrix. Equations (4)–(6) can then be written in the single requirement that $\mathbf{T}$ be unitary

$$\mathbf{T} \cdot \mathbf{T}^T = \mathbf{T}^T \mathbf{T} = \mathbf{I}.$$

Observe that the ordering of the rows of $\mathbf{T}$ is essentially arbitrary, since we can always multiply by a unitary permutation matrix, and $\mathbf{P} \cdot \mathbf{T}$ is still unitary. For the ordering chosen here, $\mathbf{T}$ is a doubly infinite block of Toeplitz matrix [6], [7] with blocks of size 2×2; since the filters in the two-channel case are always of even length [4], there are $K = N/2$ such 2×2 blocks. Hence $\mathbf{T}$, with this ordering of the rows, can be written in partitioned form [6], shown in (7) at the top of the next page where

$$\mathbf{A}_i = \begin{bmatrix} h_0(2i) & h_0(2i+1) \\ -h_0(2K-2i-1) & h_0(2K-2i-2) \end{bmatrix}. \quad (8)$$

Thus the subband transform, as we may call it, takes N points of the input data to produce two output points. Its operation is illustrated by

$$\mathbf{y} = \mathbf{T} \cdot \mathbf{x} = \mathbf{P}^{-1} \cdot \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \cdot \mathbf{x} = \mathbf{P}^{-1} \cdot \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} \quad (9)$$

where $\mathbf{x} = (\ldots, x(-1), x(0), x(1), x(2), \ldots)$ and $\mathbf{y} = (\ldots, y_0(-1), \ y_1(-1), y_0(0), y_1(0), y_0(1), y_1(1), \ldots)$. We define $\mathbf{P}$ as the orthogonal permutation matrix that reorders the elements of $\mathbf{y}$

$$\mathbf{P} \cdot \mathbf{y} = [\ldots y_0(0), y_0(1), y_0(2), \ldots y_1(0), y_1(1), y_1(2), \ldots].$$

The fact that we have an orthogonal filter bank necessarily implies that $\mathbf{T}$ is unitary, and this is equivalent to the fact that the rows of $\mathbf{T}$ form an orthonormal basis for $l^2(Z)$, the space of square summable sequences. However, the filter bank necessarily operates over infinite signals (the matrix is infinite along both dimensions) and the filters do not change with time ($\mathbf{T}$ is block Toeplitz, so the blocks are the same along diagonals). If we wish to apply the filter bank to a finite length signal we must find a *finite* unitary matrix $\mathbf{T}_f$ which has the same block structure as $\mathbf{T}$, and if we wish to change the filters we must find a unitary matrix $\mathbf{T}_c$, where at some point we change from the blocks of one filter set to another. We will show that both of these problems can be solved by an appropriate treatment of the boundary conditions. For the sake of simplicity, in this paper we consider only the two-channel case, but extension to the M-channel case is also possible [8].

The outline of the paper is as follows. In Section II, we show how to construct boundary filters to allow implementation of time-varying filter banks. The important case of how to apply a filter bank to finite length signals is treated in Section III. In Section IV, we show how boundary filters can be used to change the topology of a subband analysis tree, or to form smooth transitions between different filter sets. In Section V, we explore the relation between discrete-time time-varying filter banks and continuous-time bases of time-varying wavelets.

A preliminary version of the results of this paper was reported in [9], and some treatment, with applications, was given in [10] and [11]. Work on wavelet bases for interval regions, starting from the continuous-time domain, can also be found in [12] and [13]. This paper starts from the discrete-time point of view (filter banks) and then derives continuous-time bases. For a purely discrete-time discussion, time-varying filter banks are also constructed in [14], but the emphasis there is on numerical nonorthogonal designs, and the method is entirely different. Solutions which give inversion at the boundary are given in [15]. The M-channel solution, without continuous-time results, is given in [8].

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

$$\mathbf{T} = \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{7}$$

II. Construction of Orthogonal Boundary Filters

In this section, our concern will be to construct a half-infinite matrix which has the same block structure as $\mathbf{T}$ in (7), but has some special form at the boundary. This represents the operation of the two-channel filter bank over an infinite length right-sided signal, starting at some time n_0. This may seem like a contrived example, but treatment of this right-sided sequence allows us to examine the construction of one boundary in isolation from any other boundary effects. This will form the basis for the more complicated solutions that follow in later sections.

To apply the filter bank to this right-sided sequence, we might first try using the operator with time-domain description

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{10}$$

Clearly, $\mathbf{Q}$ is derived from $\mathbf{T}$ in that it is again block Toeplitz, with $\mathbf{A}_i$ lying along the ith block superdiagonal. However, $\mathbf{Q}$ is half infinite: it extends infinitely down and to the right, but not up or to the left. Because of the fact that we are using an orthogonal filter bank, all of the rows of $\mathbf{Q}$ are mutually orthogonal, so $\mathbf{Q}\cdot\mathbf{Q}^T = \mathbf{I}_r$, where $\mathbf{I}_r$ is the half-infinite right-sided identity matrix. However, $\mathbf{Q}^T\mathbf{Q} \neq \mathbf{I}$, so the matrix is not unitary. Hence, while the rows of $\mathbf{Q}$ form an orthogonal set, they do not form a basis for the space of right-sided sequences. In other words, more vectors are necessary to complete the basis. It is an easy matter to complete the basis, by using the Gram–Schmidt orthogonalization procedure. However, since we are using FIR filter banks, we would prefer that the additional vectors would have a small number of nonzero coefficients also. We now show that this is always the case. Hence, we can apply the Gram–Schmidt procedure and obtain the additional boundary vectors needed to complete the basis, and these will have nonzero values only in the immediate vicinity of the boundary. An elementary result from linear algebra [16], [17] will play an important role in our development.

Proposition 2.1: If $\mathbf{Q}\cdot\mathbf{Q}^T = \mathbf{I}$, then the matrix $\mathbf{Q}^T\cdot\mathbf{Q}$ is the orthogonal projection onto the span of the rows of $\mathbf{Q}$, and $(\mathbf{I} - \mathbf{Q}^T\cdot\mathbf{Q})$ is the orthogonal projection onto the space orthogonal to the span of the rows of $\mathbf{Q}$.

Proof: An operator Θ in a linear space is an orthogonal projection if and only if $\Theta^2 = \Theta$ and Θ is self-adjoint [16]–[18]. Note that

$$(\mathbf{Q}^T\cdot\mathbf{Q})^2 = \mathbf{Q}^T\cdot(\mathbf{Q}\cdot\mathbf{Q}^T)\cdot\mathbf{Q} = \mathbf{Q}^T\cdot\mathbf{I}\cdot\mathbf{Q} = \mathbf{Q}^T\cdot\mathbf{Q}.$$

Also

$$(\mathbf{Q}^T\cdot\mathbf{Q})^T = \mathbf{Q}^T\cdot\mathbf{Q}$$

so that $\mathbf{Q}^T\cdot\mathbf{Q}$ is self-adjoint, and, hence, an orthogonal projection. The range is obviously the column space of $\mathbf{Q}^T$, that is the transpose of the row space of $\mathbf{Q}$. Clearly, then, $(\mathbf{I} - \mathbf{Q}^T\cdot\mathbf{Q})$ is the projection onto the space orthogonal to the row space of $\mathbf{Q}$. □

We can use this to derive the basic result, which is that when $\mathbf{Q}$ is as given in (10), the boundary vectors also have finite support.

Proposition 2.2: If $\mathbf{Q}$ is the right-sided half-infinite matrix given in (10), corresponding to the application of an orthogonal two-channel filter bank, with filters of length $N = 2K$, then any vector orthogonal to all the rows of $\mathbf{Q}$ has nonzero values in at most the first $2K - 2$ positions.

The proof is somewhat technical, and so is given in Appendix B. However, the main idea is easily communicated. If we compare the truncated matrix in (10) with the infinite orthogonal matrix in (7), we see that $\mathbf{Q}$ resembles $\mathbf{T}$ except that it is missing the rows that extend up and to the left. Since the rows of $\mathbf{T}$ form a basis, the nullspace of $\mathbf{Q}$ lies in the span of the missing rows. Essentially, the projection matrix $\mathbf{Q}^T\mathbf{Q}$ has the form

$$\mathbf{Q}^T\mathbf{Q} = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where $\mathbf{D}$ is of dimension $(N-2)\times(N-2)$. Hence, $(\mathbf{I} - \mathbf{Q}^T\mathbf{Q})$ contains only zeros, except in the top left corner, where there is a block of size $(N-2)\times(N-2)$. Now any row vector suitable to be added to the basis, and hence orthogonal to all the rows of $\mathbf{Q}$, has the form $[(\mathbf{I} - \mathbf{Q}^T\mathbf{Q})\cdot\mathbf{x}]^T = \mathbf{x}^T(\mathbf{I} - Q^T\mathbf{Q})$ for some $\mathbf{x}$. Thus, it will have nonzero values in at most the first $N-2$ positions. It is shown in Appendix B that, in the two-channel case, the dimension of the nullspace is $(N-2)/2$.

This essentially determines the structure of the boundary, i.e., we see that we need $(N-2)/2$ boundary filters, and that each of them will have at most $N-2$ nonzero values. However, other boundary structures are possible, as we next show. These are easily explored using the material already developed.

What we have seen so far shows that the vectors needed to complete the basis necessarily have a certain form; we then just use the Gram–Schmidt procedure to compute them.

Algorithm 2.1:

1. Set $\mathbf{S}_0 = \mathbf{Q}$, choose $p = (N-2)/2$ vectors $\mathbf{l}_0, \mathbf{l}_1, \ldots \mathbf{l}_{p-1}$ linearly independent of the rows of $\mathbf{S}_0$. Set $i = 0$.
2. Set $\mathbf{l}'_i = \mathbf{l}_i^T(\mathbf{I} - \mathbf{S}_i^T \mathbf{S}_i)$ Normalize so that $\mathbf{l}''_i = \mathbf{l}'_i / \|\mathbf{l}'_i\|$.
3. Form a new matrix $\mathbf{S}_{i+1}$ by adding $\mathbf{l}''_i$ as a row to $\mathbf{S}_i$.
4. Set $i = i+1$. If $i = p$ then stop, else GOTO Step 2.
5. Set $\mathbf{T} = \mathbf{S}_p$.

Clearly, then, $\mathbf{T}$ is a half-infinite matrix, whose rows form an orthonormal basis for the space of right-sided sequences. An example of a unitary boundary in the case where $N = 4$ is

$$\mathbf{T} = \begin{bmatrix} l_0(2) & l_0(3) & 0 & 0 & 0 & 0 & \cdots \\ h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & \cdots \\ -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & \cdots \\ 0 & 0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & \cdots \\ 0 & 0 & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{11}$$

We refer to the added rows $\mathbf{l}''_i$ as the *left boundary vectors*, and the nonzero elements of these vectors as the *left boundary filters*. Of course, it should be understood that these are not linear shift-invariant filters. Clearly, both the number of boundary filters and their support are determined by the filter length N, i.e., this fixes the structure of the boundary. However, even then the set of boundary filters is not unique. A different set of linearly independent vectors input to the procedure would produce a different set of output boundary filters. Also note that the number of boundary filters for a two-channel filter bank is not necessarily a multiple of two. We can easily take care of this. To see this write the matrix containing the $p = (N-2)/2$ boundary filters in partitioned form $[\mathbf{L}_1\ \mathbf{L}_2, \ldots \mathbf{L}_{K-1}]$, where $\mathbf{L}_i$ is $p \times 2$ for $i = 1, 2, \ldots K-1$. That is

$$\mathbf{L} = [\mathbf{L}_1\ \mathbf{L}_2 \ldots \mathbf{L}_{K-1}] = \begin{bmatrix} l_0(2) & \cdots & l_0(N-2) & l_0(N-1) \\ l_1(2) & \cdots & l_1(N-2) & l_2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ l_{p-1}(2) & \cdots & l_{p-1}(N-2) & l_{p-1}(N-1) \end{bmatrix}.$$

Thus, $\mathbf{T}$ (the final output of the Gram–Schmidt procedure) can be written

$$\mathbf{T} = \begin{bmatrix} [\mathbf{L} & \mathbf{0}] \\ & \mathbf{Q} \end{bmatrix}. \tag{12}$$

If this matrix is unitary, then so is

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_d & & \mathbf{0} \\ \mathbf{0} & [\mathbf{L} & \mathbf{0}] \\ \mathbf{0} & & \mathbf{Q} \end{bmatrix} \tag{13}$$

for an arbitrary-sized identity matrix $\mathbf{I}_d$. Thus, if $\mathbf{L}$ is a boundary solution, so also is

$$\begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}. \tag{14}$$

TABLE I

COEFFICIENT OF THE BOUNDARY FILTERS FOR THE LENGTH-4 DAUBECHIES FILTERS, WITH $d = 3$. THERE ARE $(4-2)/2 + 3 = 4$ BOUNDARY FILTERS EACH OF LENGTH $(4-2) + 3 = 5$.

n	$l_0(n)$	$l_1(n)$	$l_2(n)$	$l_3(n)$
0	0.00000380	0.00002095	0.94638827	-0.32303134
1	0.48294873	-0.12939716	0.27975733	0.81960524
2	0.83651995	-0.22416796	-0.16150877	-0.47317863
3	0.22415658	0.83651290	-0.00002252	-0.00000909
4	-0.12941686	-0.48296095	0.00001300	0.00000525

Thus, in the example shown in (11), the original boundary solution is $\mathbf{L} = [l_0(2)\quad l_0(3)]$, but then so also is

$$\mathbf{L}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & l_0(2) & l_0(3) \end{bmatrix}. \tag{15}$$

In this case, by choosing $d = 1$ we have made the number of boundary filters even. Of course, the boundary filters, i.e., the rows of this matrix, won't have good frequency selectivity or other desired properties. Note that the first d filters in (13) are just "polyphase filters" (they have only a single nonzero coefficient each). The way around this is to observe that

$$\mathbf{U}.\begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix} \tag{16}$$

is also a boundary solution for arbitrary unitary matrix $\mathbf{U}$. The rows of (16) have, in general, a full $(N-2)/2 + d$ nonzero coefficients each. The next proposition shows that all possible orthogonal boundary solutions are of this form.

Proposition 2.3: If $\mathbf{T}$ and Σ are matrices corresponding to the time domain description of the same two-channel orthogonal filter bank applied to a right-sided half-infinite signal, with the same number of boundary filters, then they are related by

$$\Sigma = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}.\mathbf{T}$$

where $\mathbf{U}$ is a square unitary matrix of size $(N-2)/2 + d$.

Proof: If we chose $d = 0$, we would have $(N-2)/2$ boundary filters in each case, and both $\mathbf{T}$ and Σ would be unitary and of the form given in (12). If $d > 0$, then we write $\Lambda = [\Lambda_0 \Lambda_1 \cdots \Lambda_{K-1}]$, where Λ_0 is of size $p \times d$ and Λ_i is of size $p \times 2$ for $p = (N-2)/2 + d$, and $\mathbf{T} = [\mathbf{T}_0 \mathbf{T}_1 \cdots \mathbf{T}_{K-1}]$ where $\mathbf{T}_0$ is of size $p \times d$ and $\mathbf{T}_i$ is of size $p \times 2$ for $p = (N-2)/2 + d$

$$\Sigma \cdot \mathbf{T}^{-1} = \begin{bmatrix} \Lambda_0 & [\Lambda_1 \cdots \Lambda_{K-1} & \mathbf{0}] \\ \mathbf{0} & \mathbf{Q} & \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} \mathbf{T}_0^T \\ \mathbf{T}_1^T \\ \cdots \\ \mathbf{T}_{K-1}^T \\ \mathbf{0}^T \end{bmatrix} & \begin{matrix} \mathbf{0} \\ \mathbf{Q}^T \end{matrix} \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}$$

where

$$\mathbf{U} = [\Lambda_0\ \Lambda_1 \cdots \Lambda_{K-1}] \cdot \begin{bmatrix} \mathbf{T}_0^T \\ \mathbf{T}_1^T \\ \cdots \\ \mathbf{T}_{K-1}^T \end{bmatrix}$$

TABLE II
COEFFICIENTS OF THE BOUNDARY FILTERS FOR THE LENGTH-6 DAUBECHIES FILTERS, WITH $d = 4$. THIS GIVES SIX FILTERS OF LENGTH 8.

n	$l_0(n)$	$l_1(n)$	$l_2(n)$	$l_3(n)$	$l_4(n)$	$l_5(n)$
0	0.94069836	0.05706230	0.33257739	-0.03496808	-0.00000411	-0.00010806
1	-0.31868533	0.48658383	0.80900408	-0.08478349	-0.00001091	0.00005722
2	-0.10756660	-0.80639908	0.45660965	0.13327872	-0.33265203	-0.03517089
3	0.04422263	0.33120387	-0.13350597	0.46073113	-0.80682677	-0.08533406
4	0.00019492	0.00197942	-0.08568592	-0.80683744	-0.45999046	0.13477945
5	-0.00001788	-0.00082222	0.03530424	0.33251450	0.13505012	0.45997216
6	-0.00009779	0.00000960	0.00003587	0.00020936	0.08545297	-0.80689024
7	0.00004032	-0.00000396	-0.00001479	-0.00008632	-0.03523111	0.33267003

TABLE III
COEFFICIENTS OF THE BOUNDARY FILTERS FOR LENGTH-8 DAUBECHIES "LEAST ASYMMETRIC" FILTERS, WITH $d = 3$. THERE ARE SIX FILTERS OF LENGTH 9.

n	$l_0(n)$	$l_1(n)$	$l_2(n)$	$l_3(n)$	$l_4(n)$	$l_5(n)$
0	0.01017834	0.04074208	-0.06967680	-0.46614930	-0.88094611	-0.00451760
1	-0.02196937	-0.07282237	0.39583907	0.79348003	-0.45495344	0.03053683
2	-0.06216030	-0.35587650	0.84053651	-0.38525430	0.12013946	0.01149018
3	0.46793191	0.80092361	0.34333419	-0.05519467	0.04500677	-0.09915590
4	0.81957346	-0.46769638	-0.11360772	0.03630090	-0.02081259	-0.30637343
5	0.30425950	-0.02564484	-0.00794118	-0.01647026	0.00756330	0.80132973
6	-0.10526283	0.07424628	0.03326185	0.00585934	-0.00096755	-0.49650099
7	-0.01280043	-0.00016525	-0.00024664	0.00068985	-0.00034958	-0.02953959
8	0.03272538	0.00042248	0.00063055	-0.00176366	0.00089372	0.07552044

The dimensions of $\mathbf{U}$ are as advertised, and the fact that it is unitary follows directly from the orthogonality of the boundary filters. □

Thus, together Propositions 2.2 and 2.3 tell us how all orthogonal completions of the boundary may be constructed. The Gram–Schmidt procedure produces one solution, and then we can use Proposition 2.3 to explore the space. The set of real unitary $M \times M$ matrices can be parameterized by a lattice of $M(M-1)/2$ rotation elements [3]. Thus, if we search the set of such matrices we examine all possible orthogonal boundary terminations for a given value of d. This allows optimization to preserve desirable features at the boundary.

Clearly, all of the analysis applied here to right-sided sequences can be repeated for left-sided sequences, with suitable alteration of the details. This gives rise to *right boundary filters* associated with a particular filter set, such that (17) (at the bottom of this page) is unitary. Recall that because we consider orthogonal two-channel filter banks, there are no symmetric solutions, and this gives that the right boundary filters are necessarily different from the left boundary filters.

However, observe from Appendix A that the right boundary filters are easily generated from the left boundary filters, so separate calculation is not necessary. Note that Matlab routines to calculate the boundary filters are available electronically by anonymous ftp to ftp.ctr.columbia.edu in directory CTR-Research/advent/public/software/matlab.

A. Optimization of the Boundary Filters

Some optimization of the boundary filters seems well worthwhile in practice. While the boundary filters are obviously more constrained than the stationary filters, we can nonetheless improve the solution produced by the Gram–Schmidt procedure quite easily.

If we choose $d = 0$, the boundary filters are as constrained as possible. For example, in the case of length-4 filters there is only a single boundary filter, and the solution is unique; for length-6 filters there are two boundary filters but only $2(2-1)/2 = 1$ degree of freedom in the optimization of (16). Choosing $d = 2$ or 4 can improve things a lot by giving extra degrees of freedom. In Table I are tabulated the coefficients of the boundary filters for the Daubechies length-4 filter with $d = 3$, and in Table II those for the length-6 with $d = 4$. Choosing d larger than zero has allowed improved frequency selectivity in both cases.

We also note that if the stationary filters are minimum or maximum phase the optimization is generally more difficult than if mixed-phase filters are used. For this reason, in the examples where the "least-asymmetric" Daubechies filters [19] are different from the original filters [20] (this occurs for filter length $N \geq 8$), we have computed the boundary filters for the "least-asymmetric designs." In Table III are those for the length-8 case with $d = 3$, and in Table IV those for the length-10 with $d = 0$.

$$\mathbf{T} = \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \mathbf{0} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} \\ \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \mathbf{R}_{K-2} \end{bmatrix} \quad (17)$$

TABLE IV
COEFFICIENTS OF THE BOUNDARY FILTERS FOR LENGTH-10 DAUBECHIES "LEAST ASYMMETRIC" FILTERS, WITH $d = 0$. THERE ARE FOUR FILTERS OF LENGTH 8.

n	$l_0(n)$	$l_1(n)$	$l_2(n)$	$l_3(n)$
0	-0.07618269	-0.19420544	0.75241044	-0.62388309
1	0.18902576	-0.03570002	0.63085303	0.75080127
2	0.71119528	0.64299764	0.07437575	-0.20450737
3	0.64855872	-0.71030563	-0.17327283	-0.05884156
4	0.01442059	0.20030225	-0.01358764	-0.02895042
5	-0.17599791	0.03631470	0.01277259	0.03071044
6	-0.02145668	0.02925804	0.00002524	0.00051507
7	0.01986744	-0.02709098	-0.00002337	-0.00047692

III. APPLICATION TO FINITE LENGTH SIGNALS

A very important consideration in the construction of any subband coding scheme is that of how the borders should be treated when a finite-length signal is processed. The standard filter bank design assumes processing of infinite-length signals, and application to finite lengths will generally involve either distortion at the boundary or the introduction of some redundancy. A typical way around this difficulty has been to treat the finite signal as a segment of an infinite one formed by replicating boundary values or periodizing, or padding with zeros. A study of a number of such methods was given in [21], although some distortion remained at the boundary. A very popular remedy, given in [22], is to take a symmetric periodic extension of the signal and process with linear phase filters. There is no distortion or redundancy in this case, but the filters that can be used are restricted to be symmetric. A more recent approach allows the use of general filters, but orthogonality is not easily preserved [15].

In fact, it is immediately apparent that if, for a given filter set, we embed using left and right boundary filters, we can get a square unitary matrix for any size $L = N+2(k-1)+d_l+d_r$, provided $k > (N-2)/2$. For example, see (18) (at the bottom of this page). So we can use the already computed boundary filters. We should note that generally $L \gg N-2$, that is, the boundary filters affect only a small portion of the signal, since generally the stationary filters (of length N) will be chosen to have support much less than that of the signal.

We could equivalently write this solution as the output of a Gram–Schmidt procedure. As before, we can note that this does not produce a unique result, but all of the solutions which have the form given in (18) can be found by premultiplying any solution by

$$\begin{bmatrix} \mathbf{U}_l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_r \end{bmatrix} \qquad (19)$$

where $\mathbf{U}_l$ and $\mathbf{U}_r$ are square unitary matrices of sizes $(N-2)/2 + d_l$ and $(N-2)/2 + d_r$, respectively.

There are, however, solutions which do not have the form in (18), for example, the circulant solution. This actually corresponds to an output of the Gram–Schmidt procedure where the input vectors have nonzero portions at both ends. Other variations of this kind are found by premultiplying the circulant solution by (19) as before. The circulant solution to the finite length filter bank problem has been known for some time, however it has not proved popular, since $N-2$ of the filters operate over noncontiguous data segments, and this can result in spurious high frequency components due to the inherent periodization of the signal.

Treatment of the boundaries for other filter banks can be found in [27] and [28].

IV. CHANGING FILTER BANK TREES

A. Changing Topology

We have made extensive use of the operator, or matrix, notation to illustrate the action of the two-channel filter bank. The vector $\mathbf{H}_0 \cdot \mathbf{x}$ represents the subsampled lowpass output and $\mathbf{H}_1 \cdot \mathbf{x}$ the subsampled highpass output. Thus the single two-channel division is written

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \cdot \mathbf{x} = \mathbf{P} \cdot \mathbf{T} \cdot \mathbf{x}$$

and $\mathbf{P}$ is the orthogonal permutation matrix defined in Section I.

We can also write the action of trees based on two-channel filter banks in this fashion. For example, the outputs of a "discrete wavelet transform" are obtained by iterating along the output of the subsampled lowpass filter, and leaving the other branches alone. This can be written (in the case where there are three stages)

$$\mathbf{T}_{\text{comp}} \cdot \mathbf{x} = \text{Diag}\{\mathbf{P} \cdot \mathbf{T}, \mathbf{I}, \mathbf{I}, \mathbf{I}\} \cdot \text{Diag}\{\mathbf{P} \cdot \mathbf{T}, \mathbf{I}\} \cdot \mathbf{P} \cdot \mathbf{T} \cdot \mathbf{x}.$$

Clearly $\mathbf{T}_{\text{comp}}$ is also unitary, and is again doubly infinite. Any tree based on two-channel filter banks can be similarly written as a cascade of such matrices. So this includes arbitrary binary trees as well as discrete wavelet-packet operators [23].

The operators used need not be the doubly infinite ones associated with stationary filter banks, however. We can use any of the half-infinite or finite unitary matrices designed in Sections II and III, provided we ensure that in the cascade we apply matrices only to compatible vectors. This gives an important gain in flexibility: we can add and prune stages

$$\mathbf{T} = \begin{bmatrix} \mathbf{L}_0 & \mathbf{L}_1 & \cdots & \cdots & \mathbf{L}_{K-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{K-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \cdots & \mathbf{R}_{K-1} \end{bmatrix} \qquad (18)$$

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ddots & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \ddots & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & r_0(0) & r_0(1) & r_0(2) & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & r_1(0) & r_1(1) & r_1(2) & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & l_0(2) & l_0(3) & l_0(4) & l_0(5) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & l_1(2) & l_1(3) & l_1(3) & l_1(5) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & g_0(0) & g_0(1) & g_0(2) & g_0(3) & g_0(4) & g_0(5) & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & -g_0(5) & g_0(4) & -g_0(3) & g_0(2) & -g_0(1) & g_0(0) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (20)$$

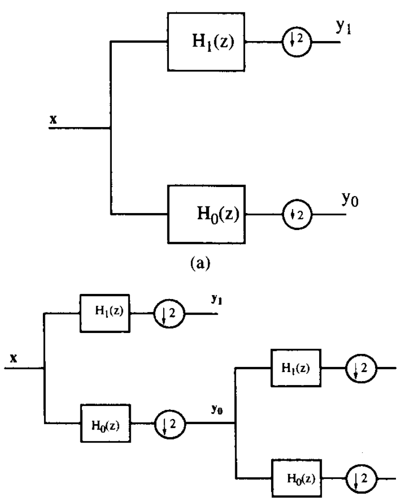


Fig. 2. Splitting of the input signal **x** using orthogonal two-channel filterbanks: (a) Single division; (b) iterated division.

from a filter bank tree if we use the left boundary filters when we wish to add a stage, and the right boundary filters when we wish to prune. In other words, using time-varying filter banks, we can change the topology of the structure over time. Consider the example of changing from the structure of Fig. 2(a) to that of Fig. 2(b) or the reverse. This would make it possible to improve the frequency resolution (going from (a) to (b)) or the time resolution (from (b) to (a)) in a time-varying fashion. An application is in the construction of arbitrary tilings of the time-frequency plane [11], where we trade time and frequency resolutions in an adaptive fashion in different parts of the plane.

To achieve the transition from Fig. 2(a) to Fig. 2(b) 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 $\mathbf{y}_0$ could be written

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

where $\mathbf{I}_l$ is a half-infinite identity matrix extending infinitely to the left and $\mathbf{T}_r$ 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

$$\mathbf{T}_2 \cdot \mathbf{x} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \mathbf{P} \cdot \mathbf{y}$$
$$= \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \mathbf{P} \cdot \mathbf{T} \cdot \mathbf{x}.$$

Thus, using cascades that include half infinite or finite unitary filter bank matrices, we can implement subband trees where the tree structure can be changed at will over time [11].

B. Overlapping Transitions

We have already used the fact that we can change the filters used in a subband analysis by using the appropriate boundary filters for each of the sets of filters involved. In the case of switching between two filter banks with filters of length N_0 and N_1, the duration of the transition is $N_0 - 2 + d_0 + N_1 - 2 + d_1$. Note that the boundary filters of the first filter set do not overlap with those of the second. The situation is illustrated in the following example matrix for the case of transition between a length $N_0 = 4$ and length $N_1 = 6$ filter bank. We chose $d_0 = 1$ and $d_1 = 0$, so that there are two boundary filters of length 3, and two of length 4. See (20) (at the top of the page).

If, however, we premultiply by a matrix of the form

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (21)$$

while the result is still unitary, the structure of the boundary is changed, so we end up with (22) (at the bottom of the next page).

Since the rows in the center overlap with both filter sets, we refer to the $t_i(n)$ as *transition filters*. These hold out the possibility of a smoother transition between the two filter sets than if the boundary filter approach is used.

A possible use is where we wish to change between filter sets with different characteristics, for example between one set that has good frequency selectivity and another that has good step response. This idea was explored in [24], where it is suggested that, for subband coding of images, filters with good frequency selectivity should be used much of the time, but that, whenever sharp edges appear, it is advantageous to switch to a set with good step response to avoid ringing distortion. Time-varying IIR filters were used in [24], and the design approach is completely different from that used here.

V. Relation Between Discrete-Time and Continuous-Time Bases

The discussion thus far has been concerned with the construction of novel discrete-time bases which were based on time-varying filter banks. In the non-time-varying case, however, it is well-known that there is a strong connection between discrete-time and continuous-time bases, at least in the two-channel case [20], [19], [25], [26], [4]. In fact, under certain constraints, we can use the former to generate the latter. In this section, we explore this relation for the time-varying bases presented above. We will construct continuous-time bases by taking the infinite iteration of discrete-time ones, much as was done for the stationary case in [20].

We shall again rely heavily on the time-domain representation of the discrete-time bases. However, while in Sections II and III we worked mostly with the matrix $\mathbf{T}$, we will now find it more convenient to explicitly refer to the lowpass and highpass operators $\mathbf{H}_0$ and $\mathbf{H}_1$ independently.

A. Boundary Functions

Just as in Section II, we will first consider the case where only a single boundary is involved, since this simplifies the analysis, and the other cases follow from it. Consider the unitary half infinite matrix $\mathbf{T}$ which was constructed in (12), so that its rows would form an orthonormal basis for the space of right-sided sequences. In Section I we had no difficulty defining $\mathbf{H}_0$ and $\mathbf{H}_1$ from $\mathbf{T}$, since $\mathbf{H}_0$ contained the rows with the lowpass filter impulse response, and $\mathbf{H}_1$ contained the rows with the highpass filter impulse response. Using the operator $\mathbf{T}$ defined for right-sided sequences in (12), we again assign to $\mathbf{H}_0$ the lowpass rows, and to $\mathbf{H}_1$ the highpass ones, but what of the boundary vector rows? We could assign all of them to $\mathbf{H}_0$, or all to $\mathbf{H}_1$, or distribute them in our definition of the two matrices. This is obviously just a question of definition; any assignment of the boundary filters between the matrices will still give $\mathbf{H}_0$ and $\mathbf{H}_1$ that satisfy (4)–(6). From now on, $\mathbf{I}$ and $\mathbf{0}$ are to be understood to represent half-infinite identity and zero matrices.

In Section II we showed that by choosing d appropriately we could always have an even number of boundary filters, and could choose $\mathbf{U}$ in (16) such that we optimized, for example, the frequency selectivity. For this section, however, we will assign all boundary vectors to $\mathbf{H}_0$, even if they have been optimized to give good lowpass/highpass responses. We will take $d = 0$.

Nonetheless, $\mathbf{H}_0$ represents the operation of the lowpass filter, followed by a subsampler, and $\mathbf{H}_1$ that of the highpass, but now $\mathbf{H}_0$ contains boundary filters and operates over right-sided sequences. Thus, for example, in the $N = 4$ case with $d = 0$, we have

$$\mathbf{H}_0 = \begin{bmatrix} l_0(1) & l_0(2) & 0 & 0 & 0 & 0 & \cdots \\ h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & \cdots \\ 0 & 0 & h_0(0) & h_0(1) & h_0(2) & h_0(3) & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

and

$$\mathbf{H}_1 = \begin{bmatrix} -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & \cdots \\ 0 & 0 & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

We will investigate the relation between discrete-time and continuous-time bases by considering iterations of the lowpass filter, just as in the non-time-varying case [20]. In operator notation this amounts to studying the products $\mathbf{H}_0^k$ for increasing k.

Since the first few rows of $\mathbf{H}_0$ contain the boundary filters, some of the first rows of $\mathbf{H}_0^k$ will be affected by the boundary filters also. However, as $k \to \infty$, only a small number of the rows of $\mathbf{H}_0^k$ are affected by the boundary filters. We show this inductively. Note that if the ith row of $\mathbf{H}_0$ has its first nonzero element in column j where $j \geq i$, then the ith row of $\mathbf{H}_0^k$ depends only on the rows $i, i+1, i+2 \cdots$ of $\mathbf{H}_0^{k-1}$ and not on the rows $< i$. Hence, if the ith row of $\mathbf{H}_0^{k-1}$ is independent of the boundary filters, so is the ith row of $\mathbf{H}_0^k$. The maximum number of rows of $\mathbf{H}_0^k$ which are affected by the boundary filters is thus easily calculated to be the least $i = i_0$, such that its first nonzero element is in the $j_0 = i_0$ column position (and $i_0 \geq N - 3$). For the example of $\mathbf{H}_0$ in the $N = 4$ case above, clearly $i_0 = 2$, since the second

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ddots & h_0(0) & h_0(1) & h_0(2) & h_0(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \ddots & -h_0(3) & h_0(2) & -h_0(1) & h_0(0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & t_0(0) & t_0(1) & t_0(2) & t_0(3) & t_0(4) & t_0(5) & t_0(6) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & t_1(0) & t_1(1) & t_1(2) & t_1(3) & t_1(4) & t_1(5) & t_1(6) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & t_2(0) & t_2(1) & t_2(2) & t_2(3) & t_2(4) & t_2(5) & t_2(6) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & t_3(0) & t_3(1) & t_3(2) & t_3(3) & t_3(4) & t_3(5) & t_3(6) & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & g_0(0) & g_0(1) & g_0(2) & g_0(3) & g_0(4) & g_0(5) & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & -g_0(5) & g_0(4) & -g_0(3) & g_0(2) & -g_0(1) & g_0(0) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (22)$$

row (we index the rows and columns from zero) has its first nonzero coefficient in the second column. This maximum is always achieved.

Since there are $(N-2)/2$ boundary filters, the $(i+N-3)$th row of $\mathbf{H}_0$ has its first nonzero element in the $2i$th column, for $i \geq N-3$. So the value of i_0 is determined by

$$2i_0 - (N-3) \geq i_0$$
$$\Rightarrow i_0 \geq N-3.$$

Hence, when we consider the products $\mathbf{H}_0^k$, there will be only $N-2$ rows of $\mathbf{H}_0^k$ affected by the boundary filters for arbitrarily large k. Similarly, the ith row of $\mathbf{H}_1$ has its first nonzero element in column $2i$. So the number of rows i_1 of $\mathbf{H}_1 \cdot \mathbf{H}_0^k$ affected by the boundary filters is given by

$$2i_1 \geq N-2$$
$$\Rightarrow i_1 \geq (N-2)/2.$$

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

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

Since the product $\mathbf{H}_0^k$ corresponds to k-stages of filtering and subsampling, it can now be shown that

$$L_{ik}(z) = z^{-1-2^k(i-i_0+1)} 2^{k/2} \prod_{p=0}^{k-1} H_0(z^{2^p}) \quad i \geq i_0. \quad (23)$$

The function $L_{i_0,k}(z)$ can easily be recognized as the z-transform of the "graphical iteration" [20] to find the scaling function $\phi_h(x)$, corresponding to the filters $H_0(z)$ (see [4]). That is, if we define from $L_{ik}(z)$ a continuous-time function

$$f_i^{(k)}(x) = \mathbf{H}_0^k(i,j) \quad j/2^k \leq x < (j+1)/2^k \quad (24)$$

it can be shown that for $i \geq i_0$, $f_i^{(k)}(x)$ converges to the scaling function $\phi_h(x + i_0 - i)$ as $k \to \infty$ (under some constraints on $h_0(n)$ [20]).

Thus the rows $i \geq i_0$ give a scheme that converge to the functions $\phi_h(x+i_0-i)$. But what of the earlier rows? This question is answered by the next proposition.

Proposition 5.1: Given $\epsilon > 0$ when $0 \leq i < i_0$, we find that $\lim f_i^{(k)}(x)$ is a finite linear combination of $\phi_h(2^p x), \phi_h(2^{p-1}x), \ldots \phi_h(2x)$ for $x > \epsilon$. As a consequence, the sequences $f_i^{(k)}(x)$ for $0 \leq i < i_0$ have the same convergence and continuity properties as those for $f_i^{(k)}(x)\, i \geq i_0$.

Proof: Examine the rows $0 \leq i < i_0$ of $\mathbf{H}_0^k$

$$\begin{aligned}\mathbf{H}_0^k(i,j) &= \sum_{m=0}^{\infty} \mathbf{H}_0(i,m)\mathbf{H}_0^{(k-1)}(m,j) \\ &= \sum_{m=0}^{i_0-1} \mathbf{H}_0(i,m)\mathbf{H}_0^{(k-1)}(m,j) \\ &\quad + \sum_{m=i_0}^{\infty} \mathbf{H}_0(i,m)\mathbf{H}_0^{(k-1)}(m,j).\end{aligned}$$

Now, using (24), and taking the limit as $k \to \infty$, we find

$$\begin{aligned}\lim_{k\to\infty} f_i^{(k)}(x) &= \sum_{m=0}^{i_0-1} \mathbf{H}_0(i,m) \lim_{k\to\infty} f_m^{(k-1)}(2x) \\ &\quad + \sum_{m=i_0}^{\infty} \mathbf{H}_0(i,m) \lim_{k\to\infty} f_m^{(k-1)}(2x) \\ &\quad\quad j/2^k \leq x < (j+1)/2^k \\ &= \sum_{m=0}^{i_0-1} \mathbf{H}_0(i,m) \lim_{k\to\infty} f_m^{(k-1)}(2x) \\ &\quad + \sum_{m=i_0}^{\infty} \mathbf{H}_0(i,m)\phi_h(2x+i_0-m) \\ &\quad\quad j/2^k \leq x < (j+1)/2^k, \ i \leq i_0. \quad (25)\end{aligned}$$

Now suppose when $m \leq i_0$ that the $f_m^{(k)}(x)$ with the largest support is zero for $x > x_0$. We know that $x_0 < N-1$ since the $f_i^{(k)}(x)$ are zero outside of $[0, N-1]$ for $i \geq i_0$. Equation (25) writes $\lim_{k\to\infty} f_i^{(k)}(x)$ in terms of $\lim_{k\to\infty} f_m^{(k-1)}(2x)$ and $\phi_h(2x+i_0-m)$. Note that for $x > x_0$ we have it in terms of $\phi_h(2x+i_0-m)$ only since all of the $f_m^{(k-1)}(2x)$ are zero for $x > x_0$. If we now write $\lim_{k\to\infty} f_m^{(k-1)}(2x)$ in terms of $\lim_{k\to\infty} f_m^{(k-2)}(4x)$ and $\phi_h(4x+i_0-m)$, we get that the left hand side of (25) can be written in terms of $\phi_h(2x+i_0-m)$ and $\phi_h(4x+i_0-m)$ for $x > x_0/2$. Continuing in this manner, we can write $\lim_{k\to\infty} f_m^{(k)}(x)$ as a finite linear combination of $\phi_h(2^p x+i_0-m), \ldots \phi_h(2x+i_0-m), \forall x > x_0/2^p$. For p large enough we have $x_0/2^p < \epsilon$, thus, for $x > \epsilon$, the $\lim_{k\to\infty} f_i^{(k)}(x), 0 \leq i < i_0$, is a finite linear combination of $\phi_h(2^p x+i_0-m), \ldots \phi_h(2x+i_0-m)$. Thus, the $\lim_{k\to\infty} f_i^{(k)}(x), 0 \leq i < i_0$, have the same convergence and continuity properties as the $\lim_{k\to\infty} f_i^{(k)}(x), i \geq i_0$. Thus, they are continuous for any $x > \epsilon > 0$, since we assume that the stationary function $\phi_h(x)$ is continuous. □

This is important, since it gives that away from the boundary the boundary functions are just as smooth as the usual scaling function $\phi_h(x)$. We define

$$\phi_i(x) = \lim_{k\to\infty} f_i^{(k)}(x).$$

For $0 \leq i \leq i_0$ we call these functions the left boundary scaling functions. Of course, for $i \geq i_0\ \phi_i(x) = \phi_h(x+i_0-i)$.

Next, let us examine the properties of these functions. Observe that by construction all rows of $\mathbf{H}_0$ are orthogonal. Since $\mathbf{H}_0\mathbf{H}_0^T = \mathbf{I}$ gives that $\mathbf{H}_0^k(\mathbf{H}_0^k)^T = \mathbf{I}$, we also have orthogonality of the rows of $\mathbf{H}_0^k, \forall k$. Hence

$$\langle f_n^{(k)}(x), f_p^{(k)}(x)\rangle = \delta_{np} \quad n,p \in \{0,1,2,\ldots\}$$

and in the limit

$$\langle \phi_n(x), \phi_p(x)\rangle = \delta_{np} \quad n,p \in \{0,1,2,\ldots\}.$$

Again, following the analogy with the non-time-varying case, to complement the scaling functions we define

$$s_i^{(k)}(x) = (\mathbf{H}_1 \cdot \mathbf{H}_0^k)(i,j) \quad j/2^k \leq x < (j+1)/2^k. \quad (26)$$

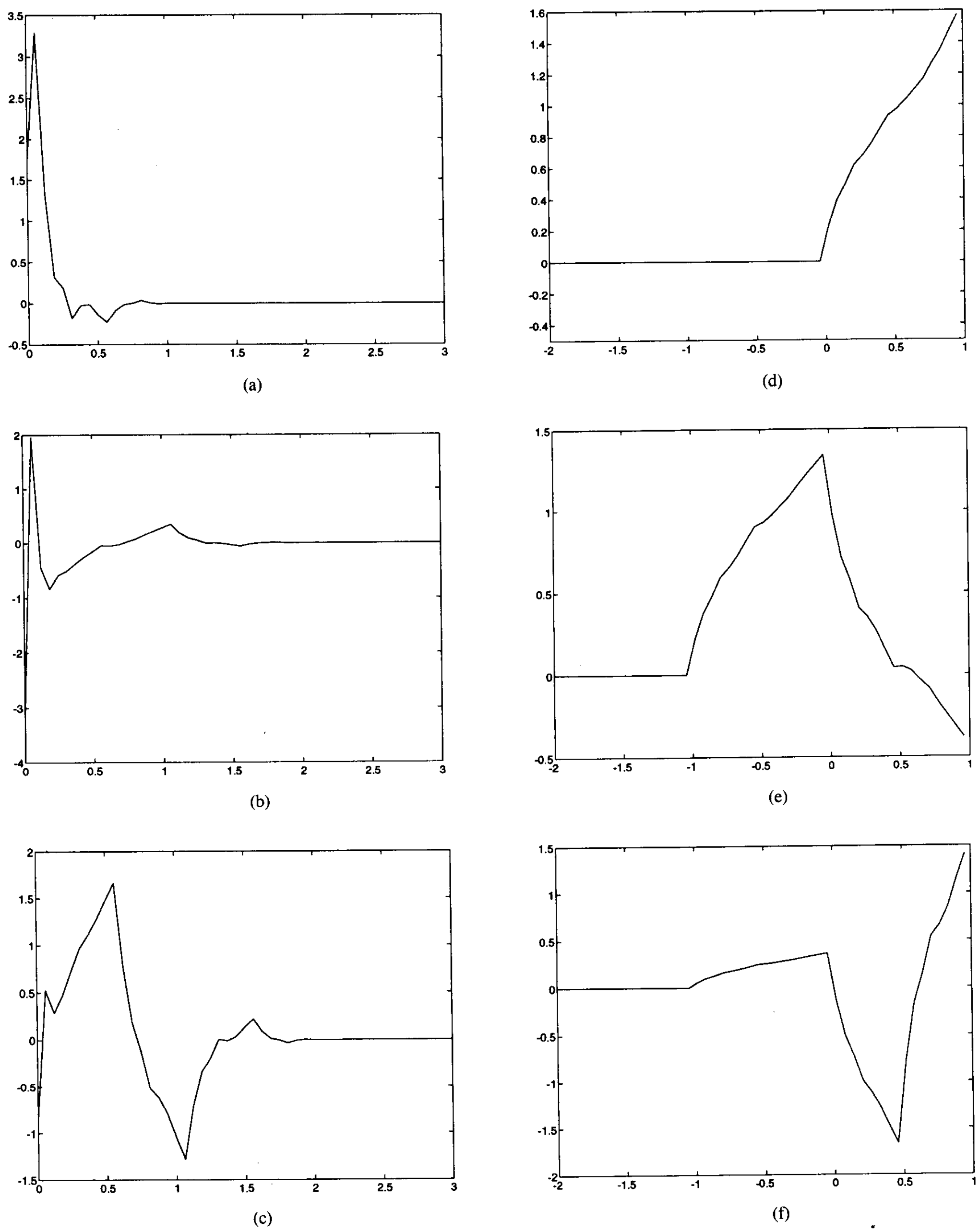


Fig. 3. The boundary scaling functions and wavelets for the Daubechies length-4 filter set. Recall that $i_0 = 1, i_1 = 0$, so there are two boundary scaling functions and one boundary wavelet at each edge. (a) First left boundary scaling function $\phi_0(x)$; (b) second left boundary scaling function $\phi_1(x)$; (c) first left boundary wavelet $\psi_0(x)$; (d) first right boundary scaling function $\phi_0^{\text{right}}(x)$; (e) second right boundary scaling function $\phi_1^{\text{right}}(x)$; (f) first right boundary wavelet $\psi_0^{\text{right}}(x)$.

Using exactly the same analysis as above, we find that the $s_i^{(k)}(x)$ converge to $\psi_h(x+i_1-i)$ for $i \geq i_0$. For $0 \leq i < i_0$, the $s_i^{(k)}(x)$ converge to functions $\psi_0(x), \psi_1(x), \ldots \psi_{i_1-1}(x)$, which we call the left boundary wavelets. It is a direct consequence of Proposition 5.1 that these sequences converge, and have the same smoothness as $\psi_h(x)$ away from the boundary. The relation

$$\langle \psi_n(x), \psi_p(x) \rangle = \delta_{np} \quad n, p \in \{0, 1, 2, \ldots\}$$

follows from $\mathbf{H}_0\mathbf{H}_0^T = \mathbf{I}$.

Because of the fact that Proposition 2.2 guaranteed orthogonality of all rows of $\mathbf{T}$ we have

$$\mathbf{H}_0 \cdot \mathbf{H}_1^T = \mathbf{0}.$$

This guarantees

$$\langle f_n^{(k)}(x), s_p^{(k)}(x)\rangle = \delta_{np} \quad n,p \in \{0,1,2,\ldots\}$$

and in the limit

$$\langle \phi_n(x), \psi_p(x)\rangle = \delta_{np} \quad n,p \in \{0,1,2,\ldots\}.$$

The two left boundary scaling functions and left boundary wavelet are shown in Fig. 3(a)–(c). The two right boundary scaling functions and left boundary wavelet are shown in Fig. 3(d)–(f). So the left boundary scaling functions and wavelets are orthogonal, as expected. To show orthogonality of the wavelets across scale, note that the wavelet at a scale k is found from the rows of $\mathbf{H}_1 \cdot \mathbf{H}_0^k$. So to show orthogonality of the wavelets at scales k and $k-1$ we observe

$$\mathbf{H}_1 \cdot \mathbf{H}_0^k \cdot (\mathbf{H}_0^T)^{k-1} \cdot \mathbf{H}_1^T = \mathbf{H}_1 \cdot \mathbf{H}_0 \cdot \mathbf{H}_1^T \tag{27}$$

$$= \mathbf{0}. \tag{28}$$

Thus

$$\langle s_n^{(k)}(x), s_p^{(k-1)}(x)\rangle = 0 \quad n,p \in \{0,1,2,\ldots\}$$

and in the limit

$$\langle \psi_n(x), \psi_p(2x)\rangle = 0 \quad n,p \in \{0,1,2,\ldots\}.$$

We have derived a set of boundary scaling functions and wavelets orthogonal to the stationary, or interior, wavelets on the half-line $[0,\infty)$. To show that we actually have a basis requires that the functions also form a complete set. To show this we compare our constructon with that of Meyer in [12]. In that construction Meyer essentially forms a basis for $[0,\infty)$ by starting with a compactly supported basis for $(-\infty,\infty)$, retaining those functions that lie completely inside $[0,\infty)$, and applying Gram–Schmidt to those that lie on the boundary. Our interior functions can thus be placed in one-to-one correspondence with those of the Meyer construction, since for $i \geq i_0$ our $\phi_i(x)$ are unaffected by the boundary. Interestingly, Meyer found that, if the basis functions are of length $N-1$, that while there are $N-2$ wavelets and $N-2$ scaling functions that straddle the boundary at any scale, only $(N-2)/2$ of the wavelets survive the Gram–Schmidt procedure (the others become zero when projected on the nullspace). Thus, we need $N-2$ boundary scaling functions, and $(N-2)/2$ boundary wavelets at any scale. These are exactly the numbers i_0 and i_1 that we found above. Hence, our functions can be put into one-to-one correspondence with those of the Meyer basis at any scale, and are complete. Note that our functions are quite distinct from those of Meyer, in that he essentially applied Gram–Schmidt to the continuous-time basis, whereas we first derived a discrete-time basis and then used an iterative scheme to converge to the continuous-time basis. For example, our scheme does not suffer from the numerical ill-conditioning observed in that of Meyer.

In the stationary case we get a wavelet basis for $(-\infty,\infty)$ by using the multiresolution analysis scheme where $\{\psi(2^{-k}x-m), m \in Z\}$ is a basis for the space W_k and the whole space $(-\infty,\infty)$ is made up as

$$\bigcup_{k=-\infty}^{\infty} W_k.$$

The basis is made up of the prototype wavelet $\psi(x)$ taken at all scales k, and at all shifts m. Because of the nesting property of the multiresolution scheme we get that $V_k = V_{k+1} \oplus W_{k+1}$. From this, we see that we do not have to use all scales of the wavelet, but can halt the analysis at some scale j, and use the scaling functions at scale j, and the wavelets at scales greater than j to form the basis

$$V_j \cup \bigcup_{k=j+1}^{\infty} W_k.$$

To get the basis for the half-line then note that we can write

$$V_j^{\text{left}} \cup \bigcup_{k=j+1}^{\infty} W_k^{\text{left}}.$$

Unlike the conventional mutiresolution analysis scheme, here the functions in V_j^{left} and W_k^{left} have support only on the half line $[0,\infty)$. Our basis for V_j^{left} is

$$\{2^{-j/2}\phi_i(2^{-j}x), 0 \leq i < i_0\} \cup \{2^{-j/2}\phi_h(2^{-j}x-m), m \in \{0,1,2\cdots\}\} \tag{29}$$

and for W_k^{left}

$$\{2^{-k/2}\psi_i(2^{-k}x), 0 \leq i < i_1\} \cup \{2^{-k/2}\psi_h(2^{-k}x-m), m \in \{0,1,2\cdots\}\}. \tag{30}$$

Together, (29) and (30), using $k = j+1, j+2\cdots$, form the basis for $[0,\infty)$.

Obviously, we construct right boundary scaling functions and wavelets in the same manner, by iterating infinitely the half-infinite matrix containing the right boundary filters. This gives a basis for the half line V_j^{right}

$$\{2^{-j/2}\phi_i^{\text{right}}(2^{-j}x), 0 \leq i < i_0\} \cup \{2^{-j/2}\phi_h(2^{-j}x-m), m \in \{0,1,2\cdots\}\} \tag{31}$$

and for W_k^{right}

$$\{2^{-k/2}\psi_i^{\text{right}}(2^{-k}x), 0 \leq i < i_1\} \cup \{2^{-k/2}\psi_h(2^{-k}x-m), m \in \{0,1,2\cdots\}\}. \tag{32}$$

Again

$$V_j^{\text{right}} \cup \bigcup_{k=j+1}^{\infty} W_k^{\text{right}}$$

gives the basis for the half-line $[-\infty, 1]$.

We would like to know if this can be used to generate a basis for [0, 1]. It can readily be seen that, since we have derived basis functions for the boundaries, if we use the left boundary scaling functions and wavelets at the left boundary, and the right boundary functions at the boundary, overall orthogonality and completeness is maintained, provided that there is no overlap between the boundary functions from the two separate boundaries. Once overlap occurs orthogonality will be lost, since the boundary functions at both edges were designed to be orthogonal to the interior functions, but not to the boundary functions from the other edge. Thus, until we reach some very coarse scale J, there is no interaction between the boundaries. Thus, we can use the wavelets to take care of all scales $j < J$ and then use the scaling functions to take care of the rest. Formally, define V_j^{int} to contain the interior functions common to V_j^{left} and V_j^{right} along with the boundary scaling functions

$$\begin{aligned} V_j^{\text{int}} =& V_j^{\text{left}} \cap V_j^{\text{right}} \cup \\ & \{2^{-j/2}\phi_i^{\text{left}}(2^{-j}x), 0 \le i < i_0\} \cup \\ & \{2^{-j/2}\phi_i^{\text{right}}(2^{-j}x), 0 \le i < i_0\}. \end{aligned}$$

And similarly define

$$\begin{aligned} W_j^{\text{int}} =& W_j^{\text{left}} \cap W_j^{\text{right}} \cup \\ & \{2^{-j/2}\psi_i^{\text{left}}(2^{-j}x), 0 \le i < i-1\} \cup \\ & \{2^{-j/2}\psi_i^{\text{right}}(2^{-j}x), 0 \le i < i_1\}. \end{aligned}$$

In this case our basis for the interval is, for sufficiently coarse J

$$V_J^{\text{right}} \cup \bigcup_{k=J+1}^{\infty} W_k^{\text{right}}.$$

Since the $\phi(x)$ and $\psi(x)$ have support over an interval of at most $N-1$, it emerges that choosing J such that $2^J \ge N$ suffices to ensure that the left and right boundary functions never overlap.

B. Properties of the Boundary Functions

Note that while we attempted to optimize the boundary filters in the discrete-time case we did not make any corresponding effort in the continuous-time case. Our goal was merely to demonstrate that discrete-time bases could be used to generate continuous time ones, much as in the stationary case.

Recall that we chose $d = 0$, and assigned all boundary filters to $\mathbf{H}_0$. This led to a system which has much in common with the continuous time bases derived for the half-line and interval by Meyer [12]. A peculiarity of this scheme is clearly that we have more boundary scaling functions than boundary wavelets at each edge. This is, of course, a consequence of the assignment of all boundary filters to $\mathbf{H}_0$. Different choices of d and different assignments of the boundary filters lead to different schemes, and equality of the numbers of boundary scaling functions and wavelets can be achieved. See, for example, the cases generated in [13]. That work also explored how the boundary filters could be designed to preserve the polynomial-approximating properties of the stationary basis functions. While the method used in this paper gives excellent control over the discrete-time boundary filters, it is less clear how to preserve desirable properties of the continuous-time boundary functions.

C. Transition Functions

In Section IV-B we saw that transition filters, which overlapped with the filter impulse responses on both sides of the transition, could be constructed from the boundary filters. When changing between filter sets, one finds transition functions as the iterates of transition filters. These can be derived by defining $\mathbf{H}_0$ and $\mathbf{H}_1$ to contain the even and odd-indexed

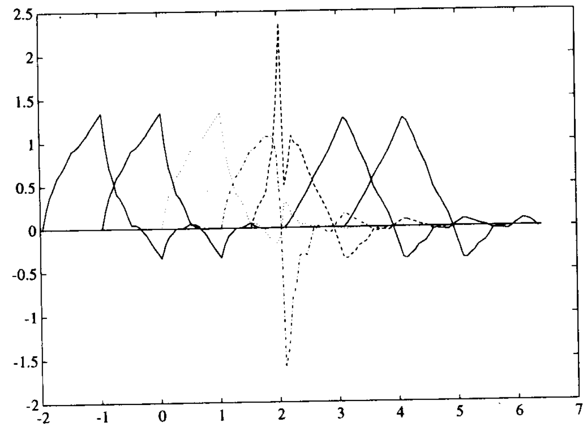


Fig. 4. Transition functions for the transition between $\phi_2(t)$ and $\phi_3(t)$. There are three orthogonal transition functions shown, which span the null space, at one scale, between $\{\phi_2(t+1), \phi_2(t+2), \ldots\}$ and $\{\phi_3(t-2), \phi_3(t-3), \ldots\}$.

rows of the doubly infinite matrix in (22) and considering the iterates of $\mathbf{H}_0^k$ as before.

In a similar way, we can consider iterates of the matrix containing the transition filters. For example, if $\mathbf{H}_0$ is the doubly infinite matrix containing the even-indexed rows of $\mathbf{T}$ from (22), then the rows of $\mathbf{H}_0^k$ give the iterates of the graphical recursion that converges to $\phi_2(x-n)$ for rows $n < 0$, to $\phi_3(x-n)$ for rows $n \ge 3$, and to transition functions for $n = 0, 1, 2$. The transition functions are as smooth as the wavelets away from the transition; however, at the single transition point there is a possible discontinuity. This is illustrated in Fig. 4, which shows the appropriate functions for the transition between the $\phi_2(x)$ and $\phi_3(x)$ scaling functions.

VI. Conclusion

We have demonstrated how to change between orthogonal filter banks in such a manner as to preserve orthogonality of the system. The method presented is complete and constructive. It allows us to construct time-varying orthogonal filter bank decompositions, to use filter banks over finite length signals, and to grow and prune an orthogonal tree on the fly. When iterated these filter banks lead to time-varying wavelet bases, and wavelet bases for interval regions.

Appendix A
Proof of Proposition A.1

Proposition A.1: The nullspace of the matrix $\mathbf{Q}$ in (10) is of dimension $(N-2)/2$.

Proof: Consider the *finite* matrix obtained by taking a truncation at both boundaries (see (33) at the top of the next page). If the matrix has r block rows then the dimension is clearly $2r \times 2(r+K-1)$, so the nullspace is of dimension $2K-2 = N-2$. Note that the top left corner of $\mathbf{H}_t$ is identical to that of $\mathbf{Q}$. We will show that precisely half of the vectors that span the nullspace of $\mathbf{H}_t$ have support only on the right, while the other half have support on the left.

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{k-1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{k-1} & 0 & \cdots & 0 & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{k-1} \end{bmatrix}. \tag{33}$$

Define $\mathbf{J}_n$ to be the antidiagonal exchange matrix, which has alternating $+1$ and -1 on the principal antidiagonal. For example

$$\mathbf{J}_4 = \begin{bmatrix} & & & 1 \\ & & -1 & \\ & 1 & & \\ -1 & & & \end{bmatrix}.$$

Because of the structure of the coefficients of $\mathbf{H}_t$ (i.e., $H_1(z) = z^{-(N-1)}H_0(-z^{-1})$, it is readily shown that

$$\mathbf{J}_{2r} \cdot \mathbf{H}_t \cdot \mathbf{J}_{2(r+k-1)} = \mathbf{H}_t. \tag{34}$$

Suppose now that we have calculated one of the left boundary vectors of $\mathbf{Q}$. It is clear that if $\mathbf{l}_0 = (l_0(2), l_0(3), \ldots l_0(N-1, 0, \cdot 0)$, then

$$\mathbf{H}_t \cdot \mathbf{l}_0 = \mathbf{0}.$$

That is, since the left boundary of $\mathbf{H}_t$ and $\mathbf{Q}$ are the same, we can use the same boundary filters. Observe now, however, that because of (34)

$$\mathbf{H}_t \cdot \mathbf{J}_{2(r+k-1)} \cdot \mathbf{l}_0 = \mathbf{0}$$

so $\mathbf{J}_{2(r+k-1)} \cdot \mathbf{l}_0$ is also a boundary filter, but it has support only on the right. Similarly, any boundary vector of $\mathbf{Q}$ can be used to generate two boundary vectors for $\mathbf{H}_t$, one with support on the left and one with support on the right. There must be precisely $(N-2)/2$ boundary vectors for $\mathbf{Q}$, since this is half the nullspace of $\mathbf{H}_t$. □

Appendix B
Proof of Proposition 2.2

Label the block rows and block columns of $\mathbf{T}$ and $\mathbf{Q}$ such that $\mathbf{T}(0,0) = \mathbf{A}_0$ and $\mathbf{Q}(0,0) = \mathbf{A}_0$. Then it is clear that the blocks are given by

$$\mathbf{T}(i,j) = \mathbf{A}_{j-i} \quad 0 \le j-i \le K-1 \tag{35}$$

$$\mathbf{Q}(i,j) = \mathbf{A}_{j-i} \quad 0 \le j-i \le K-1, i,j \ge 0 \tag{36}$$

but are zero elsewhere.

Since we know that $\mathbf{T}$ is unitary

$$\mathbf{T}^T \cdot \mathbf{T}(i,j) = \sum_{m=-\infty}^{\infty} \mathbf{T}^T(i,m)\mathbf{T}(m,j) = \sum_m \mathbf{A}_{i-m}^T \mathbf{A}_{j-m} = \delta_{i-j}\mathbf{I}. \tag{37}$$

Note that we have the restrictions $0 \le i-m \le K-1$ and $0 \le j-m \le K-1$. Thus, the range of summation is

$$\max(-i,-j) \le -m \le \min(K-i, K-j) - 1. \tag{38}$$

Now, since $\mathbf{Q}$ is defined by taking a truncation of $\mathbf{T}$ we can use (37) to calculate the blocks of $\mathbf{Q}^T\mathbf{Q}$ with the additional restriction that $m \ge 0$ in the summation. This gives the following: if $\min(K-i-1, K-j-1) \ge 0$ then $m \ge 0$, thus the additional inequality is satisfied provided

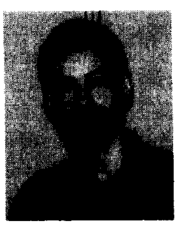
$$i \ge K-1 \quad \text{or } j \ge K-1. \tag{39}$$

Thus, the blocks of $\mathbf{Q}^T\mathbf{Q}$ are equal to the identity, except for $i < K-1$ and $j < K-1$. $\mathbf{Q}^T\mathbf{Q}$ is thus equal to a half-infinite identity matrix, except in the top corner. Clearly, then, $(\mathbf{I} - \mathbf{Q}^T\mathbf{Q})$, which is the orthogonal projection onto the span of the rows of $\mathbf{Q}$, is zero except in the top left corner. □

References

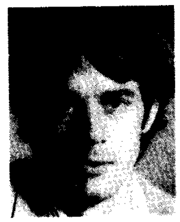
[1] M. J. T. Smith and T. P. Barnwell III, "Exact reconstruction for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 434–441, June 1986.
[2] P. P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction technique," *IEEE ASSP Mag.*, vol. 4, pp. 4–20, July 1987.
[3] ——, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice Hall, 1992.
[4] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, Sept. 1992, pp. 2207–2232.
[5] F. Mintzer, "Filters for distortion-free two-band multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 626–630, June 1985.
[6] 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.
[7] 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. 4, June 1992, pp. 1414–1429.
[8] C. Herley, "Boundary filters for finite-length signals and time-varying filter banks," *IEEE Trans. Circuits Syst. II*, scheduled for publication 1994.
[9] C. Herley and M. Vetterli, "Orthogonal time-varying filter banks and wavelets," in *Proc. IEEE Int. Symp. Circ. Syst.* (Chicago), May 1993, pp. 391–394, vol. I.
[10] C. Herley *et al.*, "Time-varying orthonormal tilings of the time-frequency plane," in *Proc. IEEE Int. Conf. ASSP* (Minneapolis, MN), Apr. 1993, pp. 205–208, vol. III.
[11] ——, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. Signal Processing*, vol. 41, Dec. 1993, pp. 3341–3359.
[12] Y. Meyer, "Ondelettes sur l'Interval," *Rev. Mat. Iberoamericana*, vol. 7, pp. 115–133, 1992.
[13] A. Cohen, I. Daubechies, and P. Vial, "Wavelet bases on the interval and fast algorithms," *J. Appl. Comput. Harmonic Anal.*, vol. 1, pp. 54–81, Dec. 1993.
[14] 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. ASSP* (San Francisco, CA), Mar. 1992, pp. 617–620.
[15] R. L. de Queiroz, "Subband processing of finite length signals without border distortions," in *Proc. IEEE Int. Conf. ASSP* (San Francisco CA), May 1992, pp. 613–616.
[16] G. Strang, *Introduction to Applied Mathematics*. Wellesley, MA: Wellesley Cambridge Press, 1986.
[17] G. E. Shilov, *Linear Algebra*. New York: Dover, 1971.

[18] I. Gohberg and S. Goldberg, *Basic Operator Theory*. Boston: Birkhäuser, 1981.
[19] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992.
[20] ——, "Orthonormal bases of compactly supported wavelets," *Commun. Pure, Appl. Math.*, vol. XLI, pp. 909–996, 1988.
[21] G. Karlsson and M. Vetterli, "Sub-band coding of finite length signals," *Signal Processing*, vol. 17, no. 2, pp. 161–168, 1989.
[22] M. J. T. Smith and S. L. Eddins, "Analysis/synthesis techniques for subband image coding," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 38, pp. 1446–1456, Aug. 1990.
[23] R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, pp. 713–718, Mar. 1992.
[24] J. L. Arrowood and M. J. T. Smith, "Exact reconstruction analysis/synthesis filter banks with time-varying filters," in *Proc. IEEE Int. Conf. ASSP* (Minneapolis, MN), Apr. 1993, pp. III 233–236.
[25] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, 1989.
[26] Y. Meyer, *Ondelettes*, vol. 1 of *Ondelettes et Opérateurs*. Paris: Hermann, 1990.
[27] R. L. de Queiroz and K. R. Rao, "Time-varying lapped transforms and wavelet packets," *IEEE Trans. Signal Processing*, vol. 41, pp. 3293–3305, Dec. 1993.
[28] H. S. Malver, *Signal Processing with Lapped Transforms*. Dedham, MA: Artech House, 1992.



Cormac Herley (S'89–M'93) was born in Cork, Ireland, in 1964. He received the B.E. (Elect.) degree from the National University of Ireland in 1985, the M.S.E.E. degree from the Georgia Institute of Technology in 1987, and the Ph.D. degree from Columbia University in 1993.

From 1987 to 1989, he worked for Kay Elemetrics Corp., Pine Brook, NJ. In 1993, he was with AT&T Bell Laboratories, Murray Hill, NJ. In 1994, he joined Hewlett-Packard Laboratories, Palo Alto, CA.



Martin Vetterli (S'86–M'86–SM'90) was born in Switzerland in 1957. He received the Dipl. El.-Ing. degree from the Eidgenössische Technische Hochschule, Zürich, Switzerland, in 1981, the Master of Science degree from Stanford University, Stanford, CA, 1982, and the Doctorat en Science degree from the Ecole Polytechnique Fédéral de Lausanne, Switzerland, in 1986.

In 1982, he was a Research Assistant at Stanford University, and from 1983 to 1986, he was a Researcher at the Ecole Polytechnique. He has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University in New York where he was an Associate Professor of Electrical Engineering, a member of the Center for Telecommunications Research, and Codirector of the Image and Advanced Television Laboratory. He is now with the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley. His rseearch interests include wavelets, multirate signal processing, computational complexity, signal processing for telecommunications, and digital video processing.

Dr. Vetterli is a member of SIAM and ACM, a member of the MDSP Committee of the IEEE Signal Processing Society and of the editorial boards of *Signal Processing, Image Communication,* and *Annals of Telecommunications.*, He received the Best Paper Award of EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986 for his thesis, and the IEEE Signal Processing Society's 1991 Senior Award (DSP Technical Area) for a 1989 TRANSACTIONS paper with D. LeGall on filter banks for subband coding.