JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 6, No. 2, June, pp. 142–153, 1995

On Orthogonal Transforms of Images Using Paraunitary Filter Banks*

RICARDO L. DE QUEIROZ†

Xerox Corporation, 800 Phillips Road, 128-27E, Webster, NY 14580

AND

K. R. RAO†,

Electrical Engineering Department, University of Texas at Arlington, Arlington, Texas 76019-0016

Received January 28, 1994; accepted November 29, 1994

The use of a paraunitary filter bank (PUFB) for image processing requires a special treatment at image boundaries, to ensure perfect reconstruction (PR) of these regions, and periodic or symmetric extensions are commonly assumed. We reduced the analysis to a one-dimensional signal assuming separable processing. Unlike infinite-length signals, PR PUFBs applied to finite-length signals will not necessarily lead to an orthogonal system. For quantization/processing of the subbands, artifacts at the image boundaries can appear due to artificial discontinuities at borders, lack of orthogonality of the effective boundary filter banks, or improper reconstruction procedure. We will explore linear-phase and nonlinear-phase PUFBs and methods to obtain orthogonality from the boundary filter banks. We will show that for symmetric extensions, orthogonality is only possible for special PUFBs based on linear-phase filters. Using time-varying boundary filter banks, we will discuss a procedure that explores all degrees of freedom of the border filters in a method essentially independent of signal extensions, allowing us to design optimal boundary filter banks, while maintaining fast implementation algorithms. © 1995 Academic Press, Inc.

I. INTRODUCTION

The applications of multirate filter banks [1, 2] in image processing are receiving increasing attention and the problems resulting from the processing of finite-length signals, as opposed to infinite-length signals often assumed in the theory of multirate filter banks, have been examined in different ways [3–16]. The only trivial solution is to use periodic extensions and circular convolution, where filtering and up- and down-sampling can be optionally implemented with the aid of the discrete Fourier transform (DFT) [3]. However, the idea of assuming a periodic signal (see Fig. 1) implies that the samples in opposite borders of the image are adjacent for the implementation algorithm. This procedure introduces artificial discontinuities and is undesirable for most applications, such as image coding. The border distortions are generally easily identified by the human eye because they have a well-defined space localization pattern, following the image boundary contours. Other sources of visible distortion patterns can arise from the use of improper reconstruction methods for the image boundaries [6], or from using schemes leading to nonorthogonal boundary filter banks [13, 16]; in which case, perfect reconstruction (PF) is assured, but the addition of near-white noise to the subbands (such as in quantization) leads to colored noise in the reconstructed signal at the boundaries. One simple approach that avoids artificial discontinuities is to assume a symmetric periodic signal [7–12], in the so-called symmetric extension method. Another approach is to use time-varying orthogonal filter banks, in which the filter banks near the borders are changed to force orthogonality [17–25].

Although our primary interest lies in image processing and coding, we reduce the problem to that of processing a finite-length one-dimensional (1D) signal, by assuming a separable transform, i.e., applying the 1D transform along the image rows, followed by 1D transform along the columns of the semitransformed image.

In terms of notation, our conventions are unidimensional concatenation of matrices and vectors indicated by a comma; $[\ ]^T$ means transposition; $\mathbf{I}_n$ is the $n \times n$ identity matrix; $\mathbf{0}_n$ is the $n \times n$ null matrix; and $\mathbf{J}_n$ is the $n \times n$ counter-identity or exchange matrix, as

* This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Technológico (CNPq), Brazil, under Grant 200.804-90-1.

† E-mail: queiroz@wre.xerox.com and rao@ee.uta.edu.

1047-3203/95 $6.00

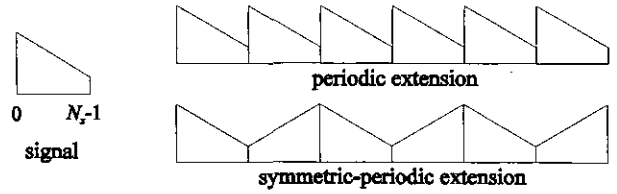


FIG. 1. Illustration of input signal of N_x samples, and its periodic and symmetric–periodic extensions.

$$\mathbf{J}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Section II reviews some filter banks concepts of interest to this paper, linking them to the processing of finite-length signals. Section III is concerned with the discussion of the symmetric extension algorithm, while Section IV presents some results on approaching orthogonal transforms through the use of special time-varying filter banks applied to the signal borders. Finally, Section V contains the conclusions of this paper.

II. FILTER BANKS AND FINITE-LENGTH SIGNALS

II.1. Paraunitary Filter Banks

We will use a PR critically decimated paraunitary uniform filter bank [2] of M FIR filters,[1] as in Fig. 2. The filters are assumed to have a maximum length $L = NM$, where N is also called the overlap factor, and the analysis and synthesis filters have impulse responses $f_k(n)$ and $g_k(n)$ ($k = 0, 1, \ldots, M-1$, $n = 0, 1, \ldots, L-1$), respectively. We will refer to such system shortly as a paraunitary filter bank (PUFB). In a PUFB, $f_k(n) = g_k(L-1-n)$ ($k = 0, 1, \ldots, M-1$, $n = 0, 1, \ldots, L-1$); i.e., synthesis filters are time-reversed versions of the analysis ones. The input signal, $x(n)$ is, thus, transformed by the analysis filter bank into the subband signals $y_k(m)$ ($k = 0, 1, \ldots, M-1$).

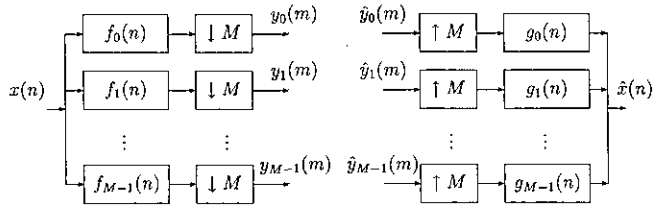


FIG. 2. Critically decimated uniform filter bank. Analysis (left) and synthesis (right) sections are shown.

[1] Filter banks in this class are also called lapped transforms [26], cautioning the reader not to confuse them with the term lapped orthogonal transform [26, 27].

After processing, the samples $\hat{y}_k(m)$ ($k = 0, 1, \ldots, M-1$) are transformed by the synthesis filter bank into the reconstructed signal $\hat{x}(n)$, which is a delayed replica of $x(n)$, if $\hat{y}_k(m) = y_k(m)$. We can also define a transform matrix $\mathbf{P}$ with elements p_{ij} as

$$p_{ij} = f_i(L-1-j) = g_i(j) \tag{1}$$

for $0 \le i \le M-1$ and $0 \le j \le L-1$ [26]. Also, it is convenient to define $\mathbf{Q}$ as the version of $\mathbf{P}$ with reversed column ordering, as

$$q_{ij} = g_i(L-1-j) = f_i(j). \tag{2}$$

We can express the input signal in its polyphase components $x_i(m) = x(mM+i)$ with Z transforms $X_i(z)$ and define a signal $y(n)$ whose polyphase components are the subbands, such that $y_i(m) = y(mM+i)$. Assume the subbands have Z transforms $Y_i(z)$. If

$$\mathbf{y}_p^{\mathrm{T}}(z) = [Y_0(z), Y_1(z), \ldots, Y_{M-1}(z)]$$
$$\mathbf{x}_p^{\mathrm{T}}(z) = [X_0(z), X_1(z), \ldots, X_{M-1}(z)],$$

then we define the polyphase transfer matrix (PTM) $\mathbf{F}(z)$ [2] as the multi-input multi-output transfer matrix such that

$$\mathbf{y}_p(z) = \mathbf{F}(z)\mathbf{x}_p(z), \tag{3}$$

and the inverse of this system $\mathbf{G}(z)$, for the synthesis section, is given by

$$\hat{\mathbf{x}}_p(z) = \mathbf{G}(z)\hat{\mathbf{y}}_p(z). \tag{4}$$

A paraunitary PTM requires that $\mathbf{F}^{-1}(z) = \mathbf{F}^{\mathrm{T}}(z^{-1})$ [2], so that a causal PR PUFB requires

$$\mathbf{G}(z) = z^{-(N-1)}\mathbf{F}^{\mathrm{T}}(z^{-1}). \tag{5}$$

If $\mathbf{P}$ and $\mathbf{Q}$ are segmented into N square matrices as

$$\mathbf{P} = [\mathbf{P}_0\, \mathbf{P}_1 \cdots \mathbf{P}_{N-1}], \tag{6}$$
$$\mathbf{Q} = [\mathbf{Q}_0\, \mathbf{Q}_1 \cdots \mathbf{Q}_{N-1}], \tag{7}$$

we can easily find the PTM as

$$\mathbf{F}(z) = \sum_{i=0}^{N-1} z^{-i}\mathbf{P}_{N-1-i}\mathbf{J}_M = \sum_{i=0}^{N-1} z^{-i}\mathbf{Q}_i \tag{8}$$

and (5) can be rewritten as either of the following set of equations [26]:

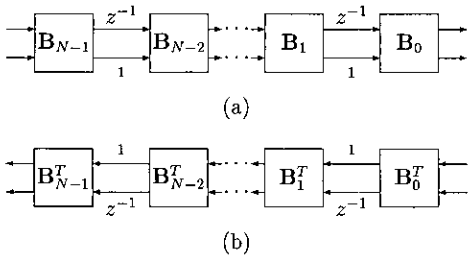


FIG. 3. Flow graph for paraunitary FIR filter banks where $\mathbf{F}(z)$ can be factorized using symmetric delays and N stages. Signals $x(n)$ and $y(n)$ are segmented and processed using blocks of M samples, all branches carry $M/2$ samples, and blocks $\mathbf{B}_i$ are $M \times M$ orthogonal matrices. (a) Analysis section; (b) synthesis section.

$$\sum_{i=0}^{N-1-l} \mathbf{P}_i \mathbf{P}_{i+l}^{\mathrm{T}} = \delta(l)\mathbf{I}_M$$
$$\sum_{i=0}^{N-1-l} \mathbf{P}_i^{\mathrm{T}} \mathbf{P}_{i+l} = \delta(l)\mathbf{I}_M. \tag{9}$$

Note that each representation of the PR conditions ((5) or (9)) implies the other, being, thus, equivalent. The only constraint is the maximum degree $(N-1)$ of the entries of the PTM, such that each filter can have an actual length lying anywhere between M and NM. Also, if one of the filters has a length much smaller than $L = NM$, it can be delayed because if $\mathbf{F}(z)$ is paraunitary, so is $D(z)\mathbf{F}(z)$, where $\mathbf{D}(z) = \mathrm{diag}\{z^{-a_1}, z^{-a_2}, \ldots, z^{-a_M}\}$, for a_i integers. In fact, delaying any filter is irrelevant when considering finite-length signals, as we can manipulate (advance or retard) the input signal and the subband signals to compensate any delay. Filters are assumed, for simplicity, to be roughly centered around $L/2$, and zero samples may be padded whenever necessary, so that we can assume all filters to have actual length L.

We will also consider more carefully the PUFBs which can be parameterized using the symmetric delay factorization (SDF). Let

$$\Lambda(z) = \begin{bmatrix} z^{-1}\mathbf{I}_{M/2} & 0 \\ 0 & \mathbf{I}_{M/2} \end{bmatrix}, \quad \tilde{\Lambda}(z) = \begin{bmatrix} \mathbf{I}_{M/2} & 0 \\ 0 & z^{-1}\mathbf{I}_{M/2} \end{bmatrix}. \tag{10}$$

The SDF of the PTM is given by

$$\mathbf{F}(z) = \mathbf{B}_0 \prod_{i=1}^{N-1} (\Lambda(z)\mathbf{B}_i) \tag{11}$$

$$\mathbf{G}(z) = \mathbf{B}_{N-1}^{\mathrm{T}} \prod_{i=N-2}^{0} (\tilde{\Lambda}(z)\mathbf{B}_i^{\mathrm{T}}), \tag{12}$$

where all stages B_i are allowed to be arbitrary $M \times M$ orthogonal matrices.

The flow graph for implementing a PUFB which can be parameterized using SDF is shown in Fig. 3 for analysis and synthesis sections. The use of SDF is not very restrictive in practice, as, for M even, most PUFBs with any practical advantage can be expressed in this way. Examples of such filter banks are linear-phase PUFBs [28–30] and cosine-modulated filter banks [2, 26, 31, 32]. Its great advantage is that it spares us the task of developing different algorithms for each border of the signal.

If the vectors $\mathbf{x}$ and $\mathbf{y}$ contain the signals $x(n)$ and $y(n)$, of unrestricted length, respectively, then the analysis and synthesis sections can be represented in matrix notation as

$$\mathbf{y} = \mathbf{T}_\infty \mathbf{x} \tag{13}$$

$$\hat{\mathbf{x}} = \mathbf{T}_\infty^{\mathrm{T}} \hat{\mathbf{y}}, \tag{14}$$

where

$$\mathbf{T}_\infty = \begin{bmatrix} \ddots & \ddots & & \ddots & & & \mathbf{0} \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & \\ & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & \\ \mathbf{0} & & & \ddots & \ddots & & \ddots \end{bmatrix}. \tag{15}$$

Note that, from (9), $\mathbf{T}_\infty \mathbf{T}_\infty^{\mathrm{T}} = \mathbf{T}_\infty^{\mathrm{T}} \mathbf{T}_\infty = \mathbf{I}_\infty$, so that the transform mapping $\mathbf{x}$ into $\mathbf{y}$ is orthogonal.

II.2. *Finite-Length Signals*

Suppose the signal $x(n)$ has only N_x samples and assume $N_x = N_B M$, where N_B is an integer representing the number of blocks, with M samples per block. To avoid the expansion of the number of samples, we require $y(n)$ to have N_x samples, so that each subband would have N_B samples. Again, let $x(n)$ and $y(n)$ be represented by the vectors $\mathbf{x}$ and $\mathbf{y}$, respectively, while, $\hat{x}(n)$ and $\hat{y}(n)$ are the corresponding signals in the synthesis section represented by vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, respectively. As a straight application of the PUFB, in the analysis section, requires more than N_x samples in $\mathbf{x}$ in order to find N_x samples of $\mathbf{y}$, we can define $\tilde{\mathbf{x}}$ as an augmented vector obtained from $\mathbf{x}$ by extending the oundary samples in any fashion, in a process we call signal extension. The matrix notation for the analysis is given by

$$\mathbf{y} = \tilde{\mathbf{P}}\tilde{\mathbf{x}}, \tag{16}$$

where $\tilde{\mathbf{P}}$ is as in (15), but with only N_B block rows. The synthesis is accomplished by

$$\tilde{\mathbf{x}} = \tilde{\mathbf{P}}^{\mathrm{T}} \hat{\mathbf{y}}. \tag{17}$$

As an example, for $N = 3$ and $N_B = 5$, we have

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 \end{bmatrix}.$$

From (9) we can see that

$$\tilde{\mathbf{P}}^{\mathrm{T}}\tilde{\mathbf{P}} = \begin{bmatrix} \Gamma_1 & & \\ & \mathbf{I}_{N_x-L+M} & \\ & & \Gamma_2 \end{bmatrix}, \tag{18}$$

where Γ_i are $(L - M) \times (L - M)$ matrices and $\Gamma_i \neq \mathbf{I}_{L-M}$, so that even if $\hat{\mathbf{y}} = \mathbf{y}$ we have $\hat{\mathbf{x}} \neq \mathbf{x}$, where the difference would occur in the last $(L - M)/2$ samples in each border. However, there is a size-limited linear transform $\mathbf{T}$ leading $\mathbf{x}$ into $\mathbf{y}$ so that [16]

$$\mathbf{y} = \mathbf{T}\mathbf{x} \tag{19}$$

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{y}. \tag{20}$$

We are seeking extension methods and filter banks such that $\mathbf{T}$ is orthogonal. For this, the assumptions are that the filters and signals are real valued, M is even, N_{B} is an integer, the filter bank is paraunitary uniform, etc., as described previously. The restrictions imposed on M and N_{B} are certainly mild and are valid for most applications. However, often one will need to use a PUFB with an odd number of channels. We do not cover such a case because SDF will not be valid and one will end up with asymmetrical solutions. Also, even numbers (actually powers of two) are much more popular choices for M. Furthermore, having a signal which is not a multiple of the block size (M) will lead to a very complicated notation and to the generation of a set of particular solutions, which may not be useful for most readers. Whenever possible the signal may be extended to reach a suitable size, as does the JPEG image coder [33]. Such approach will introduce extra subband samples and the best way may be to jointly select N_x and M in order to have N_{B} as an integer.

II.3. Image Processing and Coding

As mentioned earlier, we reduced the image-processing task to a one-dimensional problem by assuming separable transforms, where processing is applied to the image in a row and column fashion. Thus, we are actually interested in what happens to a finite-length segment of N_x samples. As we have seen, the use of PUFBs to infinite-length signals

FIG. 4. Basic subband coding diagram for transmission (or storage), where the PUFB is represented by its corresponding size-limited transform $\mathbf{T}$, and Q represents quantization.

leads to orthogonal transforms, but this is not necessarily true for finite-length signals.

In image coding, the signal $\mathbf{x}$ is transformed by $\mathbf{T}$, quantized, and transmitted, and, at the receiver side, the signal is recovered by inverse operations, as shown in Fig. 4. Orthogonal transforms have several desirable properties in image coding [34–36] regarding statistics of the quantization noise, this being the reason behind the choice to force $\mathbf{T}$ to be orthogonal. Of course, in our case, only the boundary parts of the signal would suffer the effects of the possible nonorthogonality of $\mathbf{T}$ (see (18)). However, as the noise will follow closely the image border, it would have an organized pattern and can be more easily percepted.

II.4. Periodic Signals

If, starting from $\mathbf{x}$, we create an infinite periodic signal (see Fig. 1) $\mathbf{x}_\infty$ as

$$\mathbf{x}_\infty^{\mathrm{T}} = [\cdots \mathbf{x}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \mathbf{x}^{\mathrm{T}}, \cdots], \tag{21}$$

then we use (13) through (15) applied to signals $\mathbf{x}_\infty$ and $\mathbf{y}_\infty$ where $\mathbf{y}_\infty$ is also periodic given by

$$\mathbf{y}_\infty^{\mathrm{T}} = [\cdots, \mathbf{y}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}, \cdots]. \tag{22}$$

Hence the equivalent transform $\mathbf{T}$ such that $\mathbf{y} = \mathbf{T}\mathbf{x}$ is block circulant and orthogonal [37]. The reader can also check that $\mathbf{T}$ is orthogonal for periodic extensions using (9). As an example, for $N = 3$ and $N_{\mathrm{B}} = 5$, we have

$$\mathbf{T} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 \\ \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 & \mathbf{P}_2 \\ \mathbf{P}_2 & \mathbf{0}_M & \mathbf{0}_M & \mathbf{P}_0 & \mathbf{P}_1 \end{bmatrix}.$$

II.5. Signal Extensions and Boundary Filter Banks

There are two ways to face the boundary filter banks, and, if possible, ensure the orthogonality of $\mathbf{T}$. We can produce an unlimited length signal, by assuming a particular extension of the original size-constrained signal. In this case the filter bank is assumed to be time invariant and is applied to the extended signal [7–15]. Here, an equivalent transform $\mathbf{T}$ is found from considerations based on the

particular characteristics of the extension. Alternatively, we can use time-varying filter banks, where the filter bank is changed near the signal boundaries in order to ensure full orthogonality of the size-limited transform **T** [18–25]. This time-varying filter bank approach is essentially independent of any signal extension [16].

In the first case, assume $N < N_B$ and let $\mathbf{x}_r = \mathbf{E}\mathbf{x}$, where **E** is a square matrix used to find the extended part $\mathbf{x}_r$ based on the signal **x**. Thus, we construct an infinite-length periodic signal given by

$$\mathbf{x}_\infty^T = [\cdots, \mathbf{x}^T, \mathbf{x}_r^T, \mathbf{x}^T, \mathbf{x}_r^T, \mathbf{x}^T, \mathbf{x}_r^T, \cdots]. \tag{23}$$

$\mathbf{x}_\infty$ is periodic and is processed by $\mathbf{T}_\infty$, as in (15), so that $\mathbf{y}_\infty = \mathbf{T}_\infty\mathbf{x}_\infty$, and, since both $\mathbf{T}_\infty$ and $\mathbf{x}_\infty$ have a periodic structure, then $\mathbf{y}_\infty$ is also periodic as

$$\mathbf{y}_\infty^T = [\cdots, \mathbf{y}^T, \mathbf{y}_r^T, \mathbf{y}^T, \mathbf{y}_r^T, \mathbf{y}^T, \mathbf{y}_r^T, \cdots], \tag{24}$$

where $\mathbf{y} = \mathbf{T}\mathbf{x}$ and the relation among **y**, $\mathbf{y}_r$, and **E** will be discussed later.

The second method applies time-varying filter banks as

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P}(0) & & & \\ & \mathbf{P}(1) & & \\ & & \ddots & \\ & & & \mathbf{P}(N_B - 1) \end{bmatrix}. \tag{25}$$

Let K be the greatest integer smaller than $N/2$ (the same as integer division as $K = N/2$). Hence, there are K filter banks, at each border, which have their basis functions crossing the signal boundaries. We call this the minimal complete design (MCD) when only K filter banks at each border are changed in order to achieve orthogonality of **T**. We could change all N_B filter banks but only $2K$ of them have any influence to the borders, so that we will often assume an MCBC design. Then, we have

$$\mathbf{P}(m) = \mathbf{P} \quad \text{for } K \le m \le N_B - K - 1 \tag{26}$$

and the remaining filter banks are redesigned, but remaining instantaneously paraunitary [18], and obeying PR rules for time-varying filter banks.

Comparing both methods, signal extension is advantageous for simple extension matrices, as is the popular symmetric extension [7–12], where $\mathbf{E} = \mathbf{J}_{N_x}$. Depending on the filter bank, it may be necessary to apply very complex extension methods to achieve orthogonality; thus it would be better to directly apply time-varying filter banks. We will discuss both methods in detail, and the filter banks to which they are applicable.

III. SYMMETRIC EXTENSION

The symmetric extension is the one where $\mathbf{E} = \mathbf{J}_{N_x}$ so that boundary samples of the signal $x(n)$ are reflected across the borders (see Fig. 1). Its usefulness comes from its simplicity and from the fact that $\mathbf{x}_\infty$ becomes periodic, where the period is a symmetric sequence. Let the period of $\mathbf{x}_\infty$ and of $\mathbf{y}_\infty$ be represented by $\mathbf{x}_p^T = [\mathbf{x}^T, (\mathbf{J}_{N_x}\mathbf{x})^T]$ and $\mathbf{y}_p^T = [\mathbf{y}^T, \mathbf{y}_r^T]$, respectively. Let the transform for the signal in one period be $\mathbf{T}_p$, where

$$\mathbf{y}_p = \mathbf{T}_p\mathbf{x}_p. \tag{27}$$

Then we have

$$\mathbf{x}_\infty^T = [\cdots, \mathbf{x}_p^T, \mathbf{x}_p^T, \mathbf{x}_p^T, \mathbf{x}_p^T, \cdots], \tag{28}$$

$$\mathbf{y}_\infty^T = [\cdots, \mathbf{y}_p^T, \mathbf{y}_p^T, \mathbf{y}_p^T, \mathbf{y}_p^T, \cdots], \tag{29}$$

III.1. Orthogonality and Linear-Phase Filters

Let S_{pre} and S_{pos} be $N_x \times N_x$ block diagonal matrices as diag{**S**, **S**, ... , **S**}, where **S** is an orthogonal matrix whose size divides N_x. If **T** is a size-limited transform based upon **P**, it is clear that $\mathbf{T}' = \mathbf{S}_{pos}\mathbf{T}\mathbf{S}_{pre}$ is also an orthogonal transform generated by a different filter bank, found by pre- or postprocessing the filter bank input or output with trivial block-transform operations. In this case, the extra processing is independent of the filter bank in question, **P**. This consideration is useful for the following proposition:

PROPOSITION 1. *Except for pre- or postprocessing, and with the assumptions from Sec. II.2, symmetric extensions will lead to an orthogonal size-limited transform* **T** *if and only if the filters in the PUFB have linear phase.*

Before presenting a proof, we would like to comment on the consequences of this result. First, we can always ensure PR and orthogonality using symmetric extensions for PUFBs with linear-phase filters. Second, nonlinear-phase filters cannot achieve orthogonality using a symmetric extension (except by filters found by postprocessing the output of a linear-phase PUFB with orthogonal matrices having nonsymmetric basis functions) so that, for these filters, it is better to use directly the time-varying filter bank approach in the design of the size-limited PUFB.

Demonstration. $\mathbf{T}_p$ is a block circulant orthogonal matrix [37], which can be divided into four $N_x \times N_x$ square submatrices as

$$\mathbf{T}_p = \begin{bmatrix} \mathbf{T}_0 & \mathbf{T}_1 \\ \mathbf{T}_1 & \mathbf{T}_0 \end{bmatrix}. \tag{30}$$

From $\mathbf{T}_p^T\mathbf{T}_p = \mathbf{T}_p\mathbf{T}_p^T = \mathbf{I}_{2N_x}$, and from (30), we obtain the relations

$$\begin{aligned}
\mathbf{T}_0\mathbf{T}_0^T + \mathbf{T}_1\mathbf{T}_1^T &= \mathbf{I}_{N_x} \\
\mathbf{T}_0\mathbf{T}_1^T + \mathbf{T}_1\mathbf{T}_0^T &= \mathbf{O}_{N_x} \\
\mathbf{T}_0^T\mathbf{T}_0 + \mathbf{T}_1^T\mathbf{T}_1 &= \mathbf{I}_{N_x} \\
\mathbf{T}_0^T\mathbf{T}_1 + \mathbf{T}_1^T\mathbf{T}_0 &= \mathbf{O}_{N_x}.
\end{aligned} \tag{31}$$

Consider a linear-phase filter bank and define a $M \times M$ diagonal matrix $\tilde{\mathbf{V}}$ with elements $v_{kk} = 1$, if $f_k(m)$ is symmetric and $v_{kk} = -1$, if $f_k(m)$ is antisymmetric. Then

$$P = \tilde{\mathbf{V}}\mathbf{P}\mathbf{J}_L. \tag{32}$$

Let $\mathbf{V}$ be an $N_x \times N_x$ matrix with nonzero block entries only in the counter-diagonal, as

$$\mathbf{V} = \begin{bmatrix} \mathbf{0} & & & \tilde{\mathbf{V}} \\ & & \tilde{\mathbf{V}} & \\ & \tilde{\mathbf{V}} & & \\ \tilde{\mathbf{V}} & & & \mathbf{0} \end{bmatrix} \tag{33}$$

Using (32), it is easy to verify that

$$\begin{aligned}
\mathbf{T}_0 &= \mathbf{V}\mathbf{T}_0\mathbf{J}_{N_x} \\
\mathbf{T}_1 &= \mathbf{V}\mathbf{T}_1\mathbf{J}_{N_x}.
\end{aligned} \tag{34}$$

Applying (34) into (31), and using the fact that $\mathbf{V}$ is orthogonal, we get

$$\begin{aligned}
\mathbf{T}_0\mathbf{T}_0^T + \mathbf{T}_1\mathbf{T}_1^T &= \mathbf{I}_{N_x} \\
\mathbf{T}_0\mathbf{J}_{N_x}\mathbf{T}_1^T + \mathbf{T}_1\mathbf{J}_{N_x}\mathbf{T}_0^T &= \mathbf{0}_{N_x} \\
\mathbf{T}_0^T\mathbf{T}_0 + \mathbf{J}_{N_x}\mathbf{T}_1^T\mathbf{T}_1\mathbf{J}_{N_x} &= \mathbf{I}_{N_x} \\
\mathbf{T}_0^T\mathbf{T}_1\mathbf{J}_{N_x} + \mathbf{J}_{N_x}\mathbf{T}_1^T\mathbf{T}_0 &= \mathbf{0}_{N_x}.
\end{aligned} \tag{35}$$

From (27) and (30), we have $\mathbf{y} = (\mathbf{T}_0 + \mathbf{T}_1\mathbf{J}_{N_x})\mathbf{x}$, so that $\mathbf{T} = \mathbf{T}_0 + \mathbf{T}_1\mathbf{J}_{N_x}$, and $\mathbf{T}$ is orthogonal because

$$\begin{aligned}
\mathbf{T}\mathbf{T}^T &= \mathbf{T}_0\mathbf{T}_0^T + \mathbf{T}_1\mathbf{T}_1^T + \mathbf{T}_0\mathbf{J}_{N_x}\mathbf{T}_1^T + \mathbf{T}_1\mathbf{J}_{N_x}\mathbf{T}_0^T \\
&= \mathbf{I}_{N_x} + \mathbf{0}_{N_x} = \mathbf{I}_{N_x} \\
\mathbf{T}^T\mathbf{T} &= \mathbf{T}_0^T\mathbf{T}_0 + \mathbf{J}_{N_x}\mathbf{T}_1^T\mathbf{T}_1\mathbf{J}_{N_x} + \mathbf{T}_0^T\mathbf{T}_1\mathbf{J}_{N_x} + \mathbf{J}_{N_x}\mathbf{T}_1^T\mathbf{T}_0 \\
&= \mathbf{I}_{N_x} + \mathbf{0}_{N_x} = \mathbf{I}_{N_x}.
\end{aligned}$$

Hence the sufficiency is proved. To prove the necessity, note that given (31) we would reach (35) by algebraic manipulation if and only if we have

$$\begin{aligned}
\mathbf{T}_0 &= \Phi'\mathbf{T}_0\mathbf{J}_{N_x} \\
\mathbf{T}_1 &= \Phi'\mathbf{T}_1\mathbf{J}_{N_x},
\end{aligned} \tag{36}$$

where Φ' is a square orthogonal matrix. As $\mathbf{T}_0$ and $\mathbf{T}_1$ are block circulant, presenting a periodic structure, the reader can check that (36) is only possible if $\mathbf{P}$ presents a structure such that

$$\mathbf{P} = \Phi\mathbf{P}\mathbf{J}_L, \tag{37}$$

where Φ is a square orthogonal matrix. In other words, the filters $f_k(n)$ would have to be found by a linear combination of their time-reversed versions (which are the filters $q_k(n)$). Using the fact that $\mathbf{P}\mathbf{P}^T = \mathbf{I}_M$ and after some manipulation, we can see that (37) is only true if $\Phi = \Phi^{-1}$ (Φ is symmetric and orthogonal) and that $\Phi = \mathbf{P}\mathbf{J}_L\mathbf{P}^T$. For such matrix there is an orthogonal matrix $\mathbf{A}$, such that $\mathbf{D} = \mathbf{A}^T\Phi\mathbf{A}$ [38], where $\mathbf{D}$ is a diagonal matrix. Since $\mathbf{A}$ and Φ are orthogonal, $\mathbf{D}$ is orthogonal, having elements ± 1 along the diagonal. Therefore, for every solution $\mathbf{P}$ to (37), there is a solution $\mathbf{P}_{LP}$ corresponding to linear-phase filters, where $\mathbf{P}_{LP} = \mathbf{A}^T\mathbf{P}$, such that $\mathbf{P}_{LP} = \mathbf{D}\mathbf{P}_{LP}\mathbf{J}_L$ and $\mathbf{P}_{LP}\mathbf{J}_L\mathbf{P}_{LP}^T = \mathbf{D}$. Thus, every solution $\mathbf{P}$ to (37) can be written as $\mathbf{P} = \mathbf{A}\mathbf{P}_{LP}$, which means a postprocessing of a linear-phase PUFB by a block transform $\mathbf{A}$. This concludes the demonstration.

From now on, we will assume that a linear-phase PUFB is associated with symmetric extensions, so that (32) and (34) hold.

III.2. Symmetry of the Extended Subbands

PROPOSITION 2. *$\mathbf{y}_r$ is related to $\mathbf{y}$ by $\mathbf{y}_r = (\mathbf{T}_1 + \mathbf{T}_0\mathbf{E})(\mathbf{T}_0 + \mathbf{T}_1\mathbf{E})^{-1}\mathbf{y}$, and, for symmetric extensions and linear-phase PUFBs, by $\mathbf{y}_r = \mathbf{V}\mathbf{y}$. Such relations always exist regardless of the SDF PUFB.*

Demonstration. It is possible to show that $\mathbf{T}$ is always invertible regardless of the filter bank and extension used [39]. Thus, $\mathbf{x} = \mathbf{T}^{-1}\mathbf{y}$. Since $\mathbf{y}_r = (\mathbf{T}_1 + \mathbf{T}_0\mathbf{E})\mathbf{x}$, we have $\mathbf{y}_e = (\mathbf{T}_1 + \mathbf{T}_0\mathbf{E})\mathbf{T}^{-1}\mathbf{y}$. Hence,

$$\mathbf{y}_r = (\mathbf{T}_1 + \mathbf{T}_0\mathbf{E})(\mathbf{T}_0 + \mathbf{T}_1\mathbf{E})^{-1}\mathbf{y}, \tag{38}$$

which always exists (because $\mathbf{T}$ is invertible) and is only a function of the extension ($\mathbf{E}$) and of the PUFB. For the symmetric extension, it is easy to see that

$$\begin{aligned}
\mathbf{y}_r &= \mathbf{T}_1\mathbf{x} + \mathbf{T}_0\mathbf{J}_{N_x}\mathbf{x} = (\mathbf{V}\mathbf{T}_1\mathbf{J}_{N_x} + \mathbf{V}\mathbf{T}_0)\mathbf{x} \\
&= \mathbf{V}(\mathbf{T}_0 + \mathbf{T}_1\mathbf{J}_{N_x})\mathbf{x} = \mathbf{V}\mathbf{T}\mathbf{x} \\
&= \mathbf{V}\mathbf{y}
\end{aligned} \tag{39}$$

and as a result, using symmetric extensions, $\mathbf{y}_r$ is easily found from $\mathbf{y}$ by sample mirroring (for each subband) and sign inversions.

III.3. Time-Domain Implementation

In Fig. 3, we have a *clocked* system with memory where at each instant (block index) a block of M samples in time domain is the input which is transformed into another block of M subband samples.

Based on the previous results, for the analysis, we extend the signal, through a mirror-image reflection applied to the last $\lambda = (L - M)/2$ samples on each border, resulting in a signal $\tilde{x}(n)$ with $N_x + 2\lambda = N_x + L - M$ samples, as

$$x(\lambda - 1), \cdots, x(0), x(0), \cdots, x(N_x - 1), \\ x(N_x - 1), \cdots, x(N_x - \lambda).$$

The internal states in Fig. 3a can be initialized in any fashion and the signal is processed yielding $N_B + N - 1$ blocks. We discard the first $N - 1$ output blocks, obtaining N_B transform-domain blocks corresponding to N_B samples of each subband.

At the synthesis section, we have the subband signals $\hat{y}_k(m)$ composing the signal $\hat{y}(n)$ as $\hat{y}(mM + i) = \hat{y}_i(m)$ for $0 \le i \le M - 1$. This signal $\hat{y}(n)$ is extended, by extending the subband signals by K samples in each border, as in (39), and processed as in Fig. 3b. The kth subband (initially having N_B samples) is extended as

$$v_{kk}\hat{y}_k(K - 1), \cdots, v_{kk}\hat{y}_k(0), \hat{y}_k(0), \cdots, \hat{y}_k(N_B - 1), \\ v_{kk}\hat{y}_k(N_B - 1), \cdots, v_{kk}\hat{y}_k(N_B - K).$$

Then we proceed with the synthesis over the $N_B + 2K$ blocks of $\hat{y}(n)$, obtaining a reconstructed signal with $N_B + 2K$ blocks $\hat{x}(n)$, initializing the states of Fig. 3b in any fashion. For N odd, $K = (N - 1)/2$, we discard the first $N - 1$ blocks to obtain $\hat{x}(n)$. For N even ($K = N - 2$), we discard the first $N - 1$ blocks, the first $M/2$ samples in the Nth block and the last $M/2$ samples of the signal.

In the absence of quantization/processing of the subbands, $\hat{x}(n) = x(n)$. This approach will assure the perfect reconstruction property and orthogonality of the analysis and synthesis processes, paying the price of running the algorithm over extra N or $N - 1$ blocks, making it suitable for applications where $N_B \gg N$.

III.4. DFT-Aided Implementation

In some applications, it may be more convenient to implement the linear-phase PUFB with the aid of the DFT. For this, the filtering/subsampling, or the upsampling/filtering operations can be performed in the DFT domain, as long as the signal is periodic. For the symmetric extension method, the periodic vectors are $\mathbf{x}_p$ and $\mathbf{y}_p$, and the transform $\mathbf{T}_p$ and its inverse $\mathbf{T}_p^T$ are implemented in the DFT domain. Using (27) and (39), we have

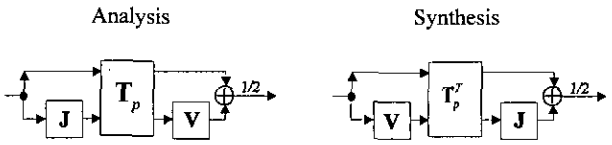


FIG. 5. Illustration of DFT-aided implementation of symmetrical extension method. The blocks $\mathbf{T}_p$ and $\mathbf{T}_p^T$ are implemented by performing the filtering and up- and down-sampling, for the analysis and synthesis sections, in the DFT domain, over an extended (but symmetric) signal.

$$\mathbf{y}_p = \begin{bmatrix} \mathbf{y} \\ \mathbf{V}\mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N_x} \\ \mathbf{V} \end{bmatrix} \mathbf{y} \tag{40}$$

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{x} \\ \mathbf{J}_{N_x}\mathbf{x} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{I}_{N_x} \\ \mathbf{J}_{N_x} \end{bmatrix} \mathbf{x}. \tag{41}$$

Hence, analysis and synthesis can be implemented as

$$\mathbf{y} = \frac{1}{2}[\mathbf{I}_{N_x} \quad \mathbf{V}]\mathbf{T}_p \begin{bmatrix} \mathbf{I}_{N_x} \\ \mathbf{J}_{N_x} \end{bmatrix} \mathbf{x} \tag{42}$$

$$\mathbf{x} = \frac{1}{2}[\mathbf{I}_{N_x} \quad \mathbf{J}_{N_x}]\mathbf{T}_p \begin{bmatrix} \mathbf{I}_{N_x} \\ \mathbf{V} \end{bmatrix} \mathbf{y}. \tag{43}$$

Such operations are illustrated in Fig. 5. We use a DFT of size $2N_x$ of a symmetric real sequence of length $2N_x$, reducing the DFT computation close to the complexity of an N_x-sample DFT. Filtering and subsampling is implemented in the DFT domain followed by an inverse DFT, to whose output we apply N_x additions. For the synthesis, the procedure is similar, where the subbands are extended in a symmetric way, and upsampling followed by filtering is performed in the DFT domain. As in the subsampling case, we apply N_x additions to the output of the inverse DFT.

IV. TIME-VARYING FILTERS

As we have observed thus far, using simple symmetric extensions, a linear-phase PUFB can achieve orthogonality and nonlinear-phase PUFBs cannot do so (except for the special case previously discussed). We will assume SDF PUFBs with generic nonlinear-phase filters and apply time-varying filter banks at the borders in an MCD. Thus, only $\mathbf{P}(0)$ through $\mathbf{P}(K - 1)$ and $\mathbf{P}(N_B - K)$ through $\mathbf{P}(N_B - 1)$ should be changed.

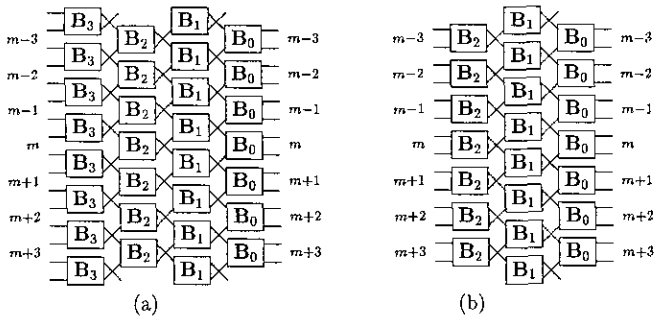


FIG. 6. Flow-graph examples for implementation of a SDF PUFB without internal states. Input and output blocks are numbered. Each branch carries $M/2$ samples. (a) $N = 4$, (b) $N = 3$.

PROPOSITION 3. *If we denote the entries of* $\mathbf{P}(m)$ *as* $p_{ij}(m)$ *for* $0 \le i \le M-1$ *and* $0 \le j \le L-1$, *and denoting* $\lambda = (L-M)/2$, *then, in order to have* $\mathbf{T}$ *as an orthogonal matrix,* $p_{ij}(m) = 0$ *for* $0 \le i \le M-1$ *and*

$$\{m, j \mid j \in [0, \lambda - mM - 1]; m \in [0, K-1]\}$$

$$\{m, j \mid j \in [L - \lambda + (N_B - 1 - m)M, L-1]; \quad m \in [N_B - K, N_B - 1]\}.$$

Demonstration. These two sets imply that $\tilde{\mathbf{P}}$ has zero entries for the first and last λ columns. To see this, consider an unlimited-length signal where a sequence of length N_x is to be transformed by $\mathbf{T}$ and imagine that the adjacent segments are also transformed by any other orthogonal transform, for example, using the identity matrix as a transform matrix. Thus, assuming $\mathbf{T}$ is orthogonal,

$$\left[\begin{array}{c|c|c} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \Psi_1 & \mathbf{T} & \Psi_2 \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array}\right], \tag{44}$$

will be an orthogonal matrix if and only if $\Psi_1 = \Psi_2 = \mathbf{0}$, meaning that we cannot allow any overlap across the signal border, and, thus, $\tilde{\mathbf{P}}$ has its first and last λ columns with zero entries.

IV.1. Minimal and Complete Set of Degrees of Freedom

For an infinite-length signal, we can draw the flow graph relating the input and output of the analysis section, as in the two examples shown in Fig. 6, which accounts for permutations and orthogonal matrices [18, 19, 20] and represents an orthogonal system as

$$\mathbf{T}_\infty = \tilde{\mathbf{B}}_0 \prod_{i=1}^{N-1} \mathbf{W}\tilde{\mathbf{B}}_i, \tag{45}$$

where $\tilde{\mathbf{B}}_i = \text{diag}\{\cdots, \mathbf{B}_i, \mathbf{B}_i, \mathbf{B}_i, \cdots\}$ and $\mathbf{W}$ is a permutation matrix that can be derived from Fig. 6. Hence, the synthesis process, defined by $\mathbf{T}_\infty^T$, would be represented by the same flow graph reversing the direction to follow the paths and substituting the orthogonal matrices by their transposes. As we saw, the transform cannot allow overlap across the signal borders and the two adjacent size-limited transforms have to be completely independent. Thus, the algorithm described by Fig. 3 is applied to a hypothetical unlimited-length signal and the stages $\mathbf{B}_i$ are modified (however, orthogonality is maintained) along the time index so that transitions among SDF PUFBs are achieved. The input and output signals are segmented into blocks of M samples, as shown in Fig. 6, and blocks are labeled 0 through $N_B - 1$ for the actual support region of $x(n)$ and $y(n)$. A simple way to find the complete SDF relevant for the signal is: (i) Construct the flow graph for the hypothetical infinite-length signal as in the examples in Fig. 6. (ii) Eliminate unnecessary paths and boxes, used for the signal outside the bounds. (iii) From the remaining boxes, those which are connected to output blocks numbered K through $N_B - K - 1$ are the same as in the time-invariant SDF and are not changed for an MCD, while the remaining can be any orthogonal matrix (maintaining their sizes) and are responsible for the degrees of freedom in the transitory boundary filter banks.

A straightforward algorithm to perform steps (ii) and (iii) is now presented. Let the ith stage be that with all matrices $\mathbf{B}_i$. Note that each box labeled $\mathbf{B}_i$ has two input or output branches (each carrying $M/2$ samples). To prune unnecessary branches and boxes, start by disconnecting the input samples outside signal bounds from the flow graph. For i varying from $i = N-1$ to $i = 0$, check all boxes in stage $N-1$ and then proceed with stage $N-2$ toward stage 0. For each box in each stage, check its input branches. If both of its input branches are disconnected, erase this box and its output branches. If only one input branch is disconnected, erase one output branch and make the box in question an $M/2 \times M/2$ orthogonal matrix. If both input branches are connected, leave the box as an $M \times M$ orthogonal matrix.

PROPOSITION 4. *The total number of degrees of freedom for all possible choices of boundary PUFBs, assuming MCD and obeying the SDF structure, is* $\nu = [(4K+1)(M-1) - 1]KM/8$ *for each border.*

Demonstration. When the prunning process is complete, and the boxes belonging to the transitory boundary filter banks are selected, we will have some orthogonal matrices (with sizes $M/2 \times M/2$ or $M \times M$) as degrees of freedom. For example, for $N = 4$ and $N_B = 6$, the resulting flow graph is shown in Fig. 7, where the generic orthogonal matrices are indicated. An $n \times n$ orthogonal matrix has $n(n-1)/2$ degrees of freedom corresponding to its plane rotation angles [38]. The reader can check that, for each

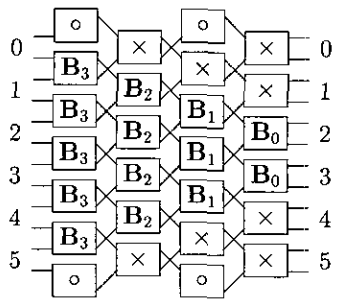


FIG. 7. Prunned flow graph for a size-limited orthogonal implementation of a PUFB for $N = 4$ and $N_B = 6$. Each branch carries $M/2$ samples. The six input and output blocks are numbered and generic $M \times M$ orthogonal matrices are marked with $\times$ while generic $M/2 \times M/2$ orthogonal matrices are marked with O.

border, the number of generic orthogonal boundary matrices is

$$\text{stage } 2i \Rightarrow K - i \text{ matrices of size } M \times M$$

$$\text{stage } 2i + 1 \Rightarrow K - i - 1 \text{ matrices of size } M \times M \text{ and one of size } \frac{M}{2} \times \frac{M}{2}$$

$$i = 0, 1, \ldots, K - 1.$$

Thus, the total number of degrees of freedom for each border is

$$\nu = \left(\sum_{i=1}^{K} i + \sum_{i=1}^{K-1} i \right) \frac{M(M-1)}{2} + K \frac{M/2(M/2-1)}{2}$$

$$= [(4K + 1)(M - 1) - 1] \frac{KM}{8},$$

and 2ν is the total number of degrees of freedom for both borders of the signal.

In the design of the boundary filter banks, for an optimal orthogonal solution we shall span all degrees of freedom in a search for the minimum of a specific cost function. As the relation among the plane rotations and cost functions is generally nonlinear, an optimization algorithm would generally have slow convergence and lead to a local minimum. Thus, a large number of variables to optimize can be burdensome. Note that ν can be a very big number (see Table 1, for some choices of K and M).

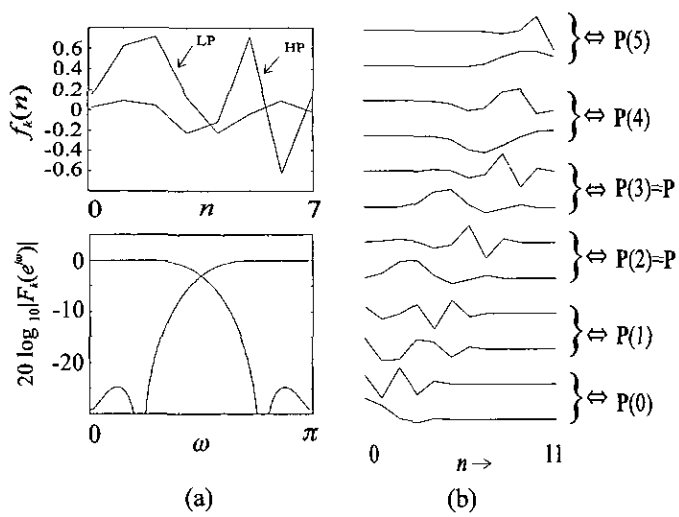


FIG. 8. Design example of orthogonal boundary filter banks based on a two-channel PUFB. (a) An eight tap two-channel PUFB ($L = 8$, $M = 2$), where the low-pass (LP) and high-pass (HP) filters $f_k(n)$ and their frequency responses are shown. (b) Design result of the bases (filters) for a 12-sample signal ($N_B = 6$) where the function maximized was an average of the stopband atenuation of the boundary filters. The relation of the basis functions and $\mathbf{P}(m)$ ($m = 0, \ldots, 5$) is indicated.

IV.2. Optimal Boundary Filter Banks

In a simple example, for $M = 2$, $N = 4$, and $N_B = 6$ (see Fig. 7), we have 4 degrees of freedom at each border. ($M/2 = 1$ and the 1×1 "orthogonal" matrices are set to 1.) We started with a two-channel eight-tap PR PUFB shown in Fig. 8a and used an unconstrained nonlinear optimization routine[2] to optimize the border matrices (one plane rotation angle per matrix), where the function maximized was an average of the stopband atenuation of the boundary filters. The 12 resulting bases for the 12-sample signal are shown in Fig. 8b, where the relation of the basis functions and $\mathbf{P}(m)$ ($m = 0, \ldots, 5$) is indicated. Note that $\mathbf{P}(2) = \mathbf{P}(3) = \mathbf{P}$ for MCD, and the 4 bases in the middle of Fig. 8b are the same as those in Fig. 8a.

As a second example, more tuned to image coding applications, we used the modulated lapped transform (MLT)

[2] All optimizations were carried using function *fmins* provided by MATLAB 4.0.

TABLE 1
The Total Number of Degrees of Freedom for Each Border, ν

	M 2	4	6	8	10	12	14	16	20	24	32
K 1	1	7	18	34	55	81	112	148	235	342	616
2	4	26	66	124	200	294	406	536	850	1236	2224
3	9	57	144	270	435	639	882	1164	1845	2682	4824
4	16	100	252	472	760	1116	1540	2032	3220	4680	8416
5	25	155	390	730	1175	1725	2380	3140	4975	7230	13000

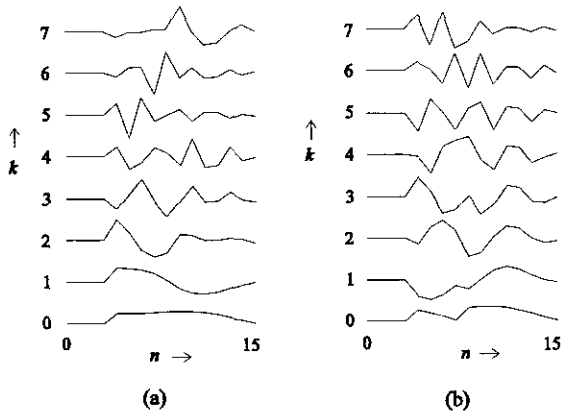


FIG. 9. Design example of orthogonal boundary filter bank for an MLT with $N = 2$ and $M = 8$. (a) Optimized filter bank $G_{TC} = 9.19$ dB. (b) Standard filter bank $G_{TC} = 5.66$ dB.

[26] with $N = 2$ and $M = 8$. We have 34 degrees of freedom at each border and we focused our attention to just one border for comparison purposes. Note that just one filter bank, $\mathbf{P}(0)$, needs to be optimized because $K = 1$. Malvar [26] provided a standard boundary solution for the MLT which is orthogonal and, therefore, it is a special case among all solutions wherein the 34 degrees of freedom would span. Here, we maximized the transform coding gain G_{TC} [34] for the boundary filter bank. Assuming $x(n)$ has autocorrelation $r_x(n) = 0.95^{|n|}$ and an autocorrelation $\mathbf{R}_{xx}$ with entries $R_{ij} = 0.95^{|i-j|}$ $(0 \le (i, j) \le L - 1)$, and denoting the diagonal elements of $\mathbf{P}\mathbf{R}_{xx}\mathbf{P}^T$ as σ_0^2 through σ_{M-1}^2, then

$$G_{TC(dB)} = 10 \log_{10} \left(\frac{1/M \sum_{i=0}^{M-1} \sigma_i^2}{(\prod_{i=0}^{M-1} \sigma_i^2)^{1/M}} \right). \tag{46}$$

In Fig. 9 are shown the bases $p_{kn}(0)$ of the standard boundary filter bank proposed by Malvar [26] and those of the optimized one. The G_{TC} for the optimal boundary filter bank is 9.19 dB, compared to 5.66 dB from that of Malvar. As a reference, the MLT has G_{TC} ranging from 8.25 to 9.22 dB (it depends upon a design parameter [26]) and the popular discrete cosine transform (DCT) has $G_{TC} = 8.83$ dB.

IV.3. Image Coding Comparisons

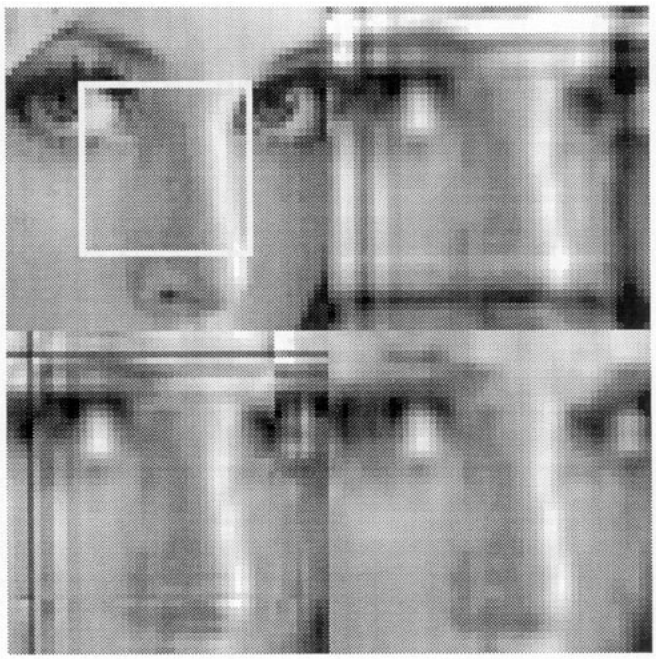
If a PR approach is applied, border distortions are eliminated when $\mathbf{y} = \hat{\mathbf{y}}$. If $\mathbf{T}$ is orthogonal, we increase the chances of avoiding border distortions in image compression applications, when severe quantization distortion corrupts $\mathbf{y}$. However, having $\mathbf{T}$ orthogonal does not assure elimination of border distortions. One key point we have to take into account is the compaction of energy provided by the boundary filter banks, i.e., how the energy of the subband coefficients is concentrated. In a good filter bank for image coding, reasonably smooth regions may just lead to few nonzero subband coefficients. If the boundary filter banks fail to do so, they may be introducing "artificial discontinuities," as is commonly the case using periodic extensions. For high-compression applications, when few subband coefficients are retained, the artificial high-frequency components either are not encoded, producing artificial edge patterns in the reconstructed image, or expend precious information bits.

To highlight the boundary regions, we made image coding tests, using the MLT, a 48×48-pels image, and adopting $M = 8$. For a two-dimensional separable implementation, there are $8^2 = 64$ subbands and 36 coefficients in each subband, where 20 of them result from boundary filter banks. Thus, as each basis function (filter) has 16 elements $(N = 2)$, boundary PUFBs will affect a 12-sample-deep region of the reconstructed image starting at each border. We carried out a comparison, using periodic extension (circular convolution), of the standard boundary orthogonal solution proposed by Malvar [26] and our optimal solution (maximum G_{TC}) described in the previous section. To simulate high-compression, we quantized only 8 out of 64 subbands and discarded the remaining. In Fig. 10a is shown the original image, indicating the region affected by boundary filter banks. In Fig. 10b we see the reconstructed image using periodic extensions, while in Figs. 10c and 10d we have the reconstructed images using Malvar's solution and the optimal boundary PUFB, respectively. As we see from the figures, the optimal boundary solution is free of border distortion at high-compression rates, while the other methods are not.

V. CONCLUSIONS

We have developed techniques to construct orthogonal boundary filter banks for PUFBs. The restrictions imposed are minimal, and the results can be changed to accommodate an odd number of channels M, in which case the basic ideas would not change, but the presentation would be greatly complicated. Simplification, in fact, is the reason behind the choice for restricting the length of the signal N_x to be a multiple of M. The motivation for studying finite-length signals is for the application of PUFBs in image processing/coding. In this case, the image dimensions are often chosen as a multiple of M; otherwise, the image is artificially extended, as does the JPEG [33] baseline coder, where M is the block size.

We have shown that linear-phase PUFBs do lead to orthogonal transforms with a simple symmetric extension (while nonlinear-phase PUFBs in general do not). This extension is exceptionally useful, due to its simplicity and due to the elimination of discontinuities across the signal (image) borders. Also, simple implementation algorithms



picture labelling → $\begin{array}{|c|c|}\hline (a) & (b) \\ \hline (c) & (d) \\ \hline \end{array}$

FIG. 10. Coding tests over a 48 × 48-pels image, simulating high-compression rates. The basic filter bank is the MLT with $M = 8$. (a) Original image, showing the region, in the reconstructed image, which could be affected by the boundary filter banks. (b) Result using periodic extension and circular convolution. (c) Result using the standard MLT boundary filter bank. (d) Result using an optimal boundary filter bank, optimized for maximum G_{TC}.

are presented. Image coding tests using a JPEG baseline coder [33] were carried to test linear-phase filter banks (with variable N) for several compression rates. We used $M = 8$ so that we merely replaced the DCT by the PUFB. In all tests, no artificial patterns caused by border distortion were visible, and the distortions due to quantization in those regions have the same intensity and patterns as the distortions in central regions of the image, which were not affected by the border discontinuity.

For nonlinear-phase PUFBs, methods to construct orthogonal boundary filter banks have been reported earlier [18–25]. We have presented a general solution (as long as the PUFB obeys the SDF) and have explicitly pointed the degrees of freedom of such transitions. This allowed us to easily design optimal boundary filter banks. The absence of border distortion is also clear from our image coding tests using optimized boundary filter bank, providing a great improvement in relation to existing orthogonalization methods for the MLT.

Also, the choice for SDF PUFBs aims to simplify the presentation. For a generic (canonical) factorization [2], the algorithm to find the size-limited transform, as explained in Section IV.1, is similar, except that the symmetries disappear and it would not be so easy to quantify the parameters in each step or to quantify ν, the number of degrees of freedom in each border. However, the basic philosophy and methods to prune the flow graph are the same.

Cases such as nonuniform and IIR filter banks are certainly interesting topics for future research in the field.

REFERENCES

1. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing,* Prentice–Hall, Englewood Cliffs, NJ, 1983.
2. P. P. Vaidyanathan, *Multirate Systems and Filter Banks,* Prentice–Hall, Englewood Cliffs, NJ, 1983.
3. J. W. Woods (Ed.), *Subband Coding of Images,* Kluwer Academic, Hingham, MA, 1991.
5. P. H. Westerink, Sub-band coding of images, Ph.D. thesis, Delft Technische Universiteit, The Netherlands, Oct. 1989.
5. C. Diab, R. Prost, and R. Goutte, Error-free image decomposition/reconstruction for subband coding schemes, *Signal Process. Image Commun.* **2,** May 1990, 53–68.
6. G. Karlsson and M. Vetterli, Extension of finite length signals for subband coding, *Signal Process.* **17,** 1989, 161–168.
7. M. J. Smith and S. L. Eddins, Analysis/synthesis techniques for subband image coding, *IEEE Trans. Acoust. Speech Signal Process.* **38,** Aug. 1990, 1446–1456.
8. S. Martucci, Signal extension and noncausal filtering for subband coding of images, *SPIE VCIP '91: Visual Commun.* **1605,** Nov. 1991, 137–148.
9. L. Chen, T. Q. Nguyen, and K. P. Chuan, Symmetric extension methods for parallel M-channel perfect reconstruction linear-phase FIR analysis/synthesis systems, preprint.
10. J. N. Bradley, C. M. Brislawn, and V. Faber, Reflected boundary conditions for multivariate filter banks, in *Proceedings of International Symposium on Time-Frequency and Time-Scale Analysis, Victoria, Canada Oct. 1992,* pp. 307–310.
11. K. Nishikawa, H. Kiya, and M. Sagawa, Property of circular convolution for subband image coding, in *Proceedings of International Conference on Acoustics, Speech, Signal Processing,* 1992, Vol. IV, pp. 281–284.
12. R. Bamberger, S. L. Eddins, and V. Nury, Generalizing symmetric extensions: Multiple nonuniform channels and multidimensional nonseparable IIR filter banks, in *Proceedings International Symposium on circuits and Systems,* 1992, Vol. II, pp. 991–994.
13. R. L. de Queiroz, Subband processing of finite length signals without border distortions, in *Proceedings of International Conference on Acoustics, Speech, Signal Processing,* 1992, *Vol. IV,* 1992, pp. 613–616.
14. R. L. de Queiroz and K. R. Rao, Reconstruction methods for processing finite-length signals with paraunitary filter banks, *IEEE Trans. Signal Process.,* to appear.
15. J. N. Bradley and V. Faber, Perfect reconstruction with critically sampled filter banks and linear boundary conditions, preprint.
16. V. Nury and R. Bamberger, A theory of size-limited filter banks," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Vol. III, 1993, pp. 161–164.
17. J. Pesquet, Orthonormal wavelets for finite sequences, in *Proceedings*

of International Conference on Acoustics, Speech, Signal Processing, 1992, Vol. IV, pp. 609–612.

18. R. L. de Queiroz and K. R. Rao, Time-varying lapped transforms and wavelet packets, *IEEE Trans. Signal Process.* **41,** Dec. 1993, 3293–3305.
19. R. L. de Queiroz and K. R. Rao, Adaptive extended lapped transforms, in *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing, Minneapolis, MN, Apr. 1993,* Vol. III, pp. 217–220.
20. R. L. de Queiroz and K. R. Rao, On adaptive wavelet packets, in *Proceedings of IEEE International Symposium on Circuits and Systems, Chicago, IL,* May 1993, Vol. I, pp. 511–514.
21. R. A. Gopinath and C. S. Burrus, Factorization approach to unitary time-varying filter banks, Technical Report CML-TR92-93, Rice University, Houston, TX, Dec. 1992.
22. C. Herley, J. Kovacevic, K. Ramchandran, and M. Vetterli, Arbitrary orthogonal tilings of the time-frequency plane, in *Proceedings of International Symposium on Time-Frequency and Time-Scale Analysis, Victoria, Canada, Oct. 1992.*
23. C. Herley, J. Kovacevic, K. Ramchandran, and M. Vetterli, Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms, *IEEE Trans. Signal Process.* **41,** Dec. 1993, 3341–3359.
24. C. Herley and M. Vetterli, Orthogonal time-varying filter banks and wavelet packets, preprint.
25. C. Herley and M. Vetterli, Orthogonal time-varying filter banks and wavelets, in *Proceedings of IEEE International Symposium on Circuits and Systems, Chicago, IL, May 1993,* Vol. I, pp. 391–394.
26. H. S. Malvar, *Signal Processing with Lapped Transforms,* Artech House, Norwood, MA, 1992.
27. H. S. Malvar and D. H. Staelin, The LOT: Transform coding without blocking effects, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-37,** Apr. 1989, 553–559.
28. A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen, Linear-phase orthonormal filter banks, in *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing, Minneapolis, MN,* Apr. 1993, Vol. III, pp 209–212.
29. A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen, Linear-phase paraunitary filter banks: Theory, factorizations and applications, *IEEE Trans. Signal Process.* **41,** Dec. 1993, 3480–3496.
30. R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, The generalized lapped orthogonal transforms, *Electron. Lett.* **30,** Jan. 1994, 107–107.
31. H. S. Malvar, Extended lapped transform: Fast algorithms and applications, in *Proceedings of International Conference on Acoustics, Speech, Signal Processing, Toronto, Canada, 1991,* pp. 1797–1800.
32. R. D. Koilpillai and P. P. Vaidyanathan, Cosine-modulated FIR filter banks satisfying perfect reconstruction, *IEEE Trans. Signal Process.* **40,** Apr. 1992, 770–783.
33. W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard,* Van Nostrand Reinhold, New York, 1993.
34. N. S. Jayant and P. Noll, *Digital Coding of Waveforms,* Prentice-Hall, Englewood Cliffs, NJ., 1984.
35. N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing,* Springer, New York, 1975.
36. K. R. Rao (Ed.), *Discrete Transforms and Their Applications,* Van Nostrand Reinhold, New York, 1985.
37. M. Vetterli and D. Le Gall, Perfect reconstruction filter banks: Some properties and factorizations, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-37,** July 1989, 1057–1071.
38. F. E. Hohn, *Elementary Matrix Algebra,* second ed., MacMillan, New York, 1964.
39. R. L. de Queiroz, On lapped transforms, Ph.D. dissertation, University of Texas at Arlington, Aug. 1994.

RICARDO L. DE QUEIROZ received the B.S. degree from Universidade de Brasilia, Brazil, in 1987, the M.S. degree from Universidade Estadual de Camprinas, Brazil, in 1990, and the Ph.D. degree from University of Texas at Arlingon, in 1994, all in electrical engineering. In 1990–1991, he was with the DSP research group at Universidade de Brasilia, as a research associate. In 1993 he received the Academic Excellence Award from the Electrical Engineering Department of the University of Texas at Arlington and in 1994 he was a teaching assistant at the same university. He joined Xerox Co. in August 1994, where he is currently a member of the research staff at the Advanced Color Imaging group. His research interests are multirate signal processing, filter banks, image and signal compression, and image databases.

K. R. RAO received the Ph.D. degree in electrical engineering from the University of New Mexico, Albuquerque, in 1966. Since 1966, he has been with the University of Texas at Arlington (UTA) where he is currently a professor of electrical engineering. He has published extensively in reviewed technical journals in the areas of discrete transforms and digital image coding. He is the coauthor of the books *Orthogonal Transforms for Digital Signal Processing* (Springer-Verlag 1975), *Fast Transforms: Analyses and Applications* (Academic Press, 1982), and *Discrete Cosine Transform—Algorithms, Advantages, and Applications* (Academic Press, 1990), as well as coauthor or editor on other works. Some of these books have been translated into Chinese, Japanese, and Russian. He has conducted workshops on digital image coding worldwide, for universities, research institutes, and industry.