BOT's Based on Nonuniform Filter Banks

Anamitra Makur

Abstract—Parallels between orthogonal transforms and filter banks have been drawn before. Block orthogonal transform (BOT) is a special case of orthogonal transform where a nonoverlapping window is used. In this paper we relate BOT's to filter banks. Specifically, we show that any BOT can be shown as a perfect reconstruction filter bank, and any tree-structured perfect reconstruction filter bank or any orthonormal filter bank for which no filter length exceeds its decimation factor can be shown as a BOT. We then show that all conventional BOT's map to uniform filter banks. A construction method to design a BOT from any nonuniform filter bank is presented, and finding an optimal tree structure (in the sense of transform coding gain) for a given source has also been discussed. Results show that the optimal nonuniform BOT outperforms uniform BOT's having either the same number of bands or the same size in most cases.

I. INTRODUCTION

BLOCK orthogonal transform (BOT) is a well-established field for decomposition and compression of signals, and a large number of transforms have been proposed and used [1], [2]. For a given size, Karhunen–Loeve transform (KLT) is optimal in the sense that it decomposes a signal into maximally uncorrelated components, consequently producing maximum coding gain. However, KLT is a signal-dependent transform and is seldom used for real-life signals (which are without exception nonstationary) due to operational difficulties. Signalindependent suboptimal transforms, on the other hand, have gained popularity.

BOT is recognized to be a special case of subband coding, another approach used for decomposition and compression of signals [2, p. 246]. A subband coder consists of a set of analysis filter banks (encoder) and a set of synthesis filter banks (decoder). Such filter banks have also been extensively studied [3]. Various construction methodologies for designing filter banks satisfying desirable properties such as perfect reconstruction are known, and a lot of flexibility exists in such designs. Transforms have been designed based on filter banks, for example, the discrete-time wavelet transform [4]. Block transforms have also been recently designed from filter banks [5]. In this paper we extend such methods to designing arbitrary BOT's from filter banks. One freedom one has in choosing a filter bank is that the filter bank may be nonuniform (unequal decimation factors and unequal passband widths). It turns out that all known BOT's are derived from uniform filter banks. If a BOT is to be designed based on any nonuniform filter bank, it is possible to design it in a signal-dependent manner. Such a BOT is suboptimal, but

Manuscript received May 22, 1994; revised January 10, 1996. The associate editor coordinating the review of this paper and approving it for publication was Prof. Roberto H. Bamberger.

The author is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India.

Publisher Item Identifier S 1053-587X(96)05299-3.

being signal-dependent, may perform better than conventional signal-independent suboptimal BOT's of equal size. This fact has been verified later in the paper. Further, the signaldependence of BOT's designed from nonuniform filter banks makes it possible to use them in an adaptive manner for nonstationary signals. It turns out that such adaptation requires much less overhead than adaptation of KLT.

Another expected advantage of nonuniform BOT's is that, unequal frequency resolution implies unequal time resolution (for a nonredundant system such as BOT). Therefore, for a given transform size N, a nonuniform BOT will have time resolution less than N for at least some frequency bands, whereas a uniform BOT has equal time resolution of N samples in every band. When the transform coefficients are quantized for compression, a band with finer time resolution would have localized reconstruction error as opposed to distributed error over the entire block. Distributed error is known to be a problem in transform coding. For example, N = 32 to 512 is used in speech compression. If a silence period is followed by a sharp sound, a transform block spanning across both results in an audible preecho. Taking another example, even though small N is used for image compression, a block containing a sharp edge may result in some blurring of the edge. Since both speech and image are low-pass signals, a nonuniform BOT tuned to such signals has finer time resolution at higher frequencies (see Example 2), which is likely to produce subjectively better results than uniform BOT's.

We start with drawing the parallels between BOT and filter bank. Conventional BOT's will be shown to be uniform. We shall then present a construction procedure to find the optimal nonuniform tree structure for a given signal. Finally, design examples and coding performance comparisons with the known BOT's have been presented.

II. BOT'S AS FILTER BANKS

A BOT of size N is specified by N basis vectors $\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots \boldsymbol{b}_{N-1}$, each being a column vector of size N. For an input vector $\boldsymbol{x}^t = [x_0 x_1 \cdots x_{N-1}]$, transform means decomposing \boldsymbol{x} into N basis vectors,

$$\boldsymbol{x} = \sum_{j=0}^{N-1} \theta_j \boldsymbol{b}_j = \boldsymbol{B} \Theta \tag{1}$$

where θ_j are scalars known as transform coefficients, Θ is a column vector, and $B = [b_0 b_1 \cdots b_{N-1}]$ is the inverse transform matrix of size $N \times N$. The forward transform matrix is, therefore, $A = B^{-1}$. There are two requirements on A. In order to get back x from Θ (perfect reconstruction), the basis vectors have to be linearly independent. For efficient



Fig. 2. Conventional BOT.

decomposition into as much uncorrelated components as possible (which eventually results in better compression), the basis vectors would better be orthogonal. Note that orthogonality here implies linear independence.

A subband coder with N bands has the form as shown in Fig. 1. Here the input sequence x(n) is decomposed through a set of filters $H_0(z), H_1(z) \cdots H_{N-1}(z)$. The *n*th band is subsampled by a factor m_n before leaving the encoder. In the decoder each band is interpolated by the corresponding factor, and passed through the synthesis filter $F_n(z)$ before recombining to output sequence y(n). Again, there are two requirements on a subband system. For efficient decomposition/compression, the number of samples after encoding should not be more than the number of input samples. A nonredundant (also known as critically sampled) system would keep the number unchanged after encoding, that is,

$$\sum_{n=0}^{N-1} \frac{1}{m_n} = 1.$$
 (2)

For perfect reconstruction (y(n) is a delayed version of x(n)), the aliasing terms have to be zero. The alias-component matrix H(z) of size $N \times N$ is defined such that its *mn*th element is $H_m(ze^{-j2\pi n/N})$. Then this matrix should satisfy

$$\boldsymbol{H}(z)\boldsymbol{H}^{T}\left(\frac{1}{z^{*}}\right) = N\boldsymbol{I}$$
(3)

where T denotes conjugate transpose and * denotes conjugate, and I is the identity matrix. Satisfying this condition means that the analysis filter bank $H_n(z)$ is paraunitary. The synthesis filters, in turn, should satisfy

$$F_n(z) = z^{-L+1} H_n(z^{-1}) \quad \text{for } 0 \le n \le N-1$$
 (4)

where L is the length of the analysis filters [3]. Note that the perfect reconstruction requirements are for a nonredundant subband system.

Now consider representing a conventional BOT A by a subband system. The forward transform is denoted by

$$\Theta = Ax = B^t x \tag{5}$$

which can be written as convolution sums

$$\theta_j = \boldsymbol{b}_j^t \boldsymbol{x} = \sum_{i=0}^{N-1} x_i b_{ij} \tag{6}$$

where b_{ij} is the *i*th component of the *j*th basis vector. This, therefore, can be interpreted as a filtering of x(n) advanced by N-1 samples

$$\theta'_j(n) = \sum_{i=0}^{N-1} x(n+N-1-i)h_j(i)$$
(7)

where

$$h_j(i) = b_{N-1-i,j} \tag{8}$$

are the impulse response coefficients of the filter $H_j(z)$. Further, the sequences $\theta'_j(n)$ are subsampled by a factor of N to get the sequences $\theta_j(n)$ which are the transform coefficients. In a similar fashion, the inverse transform can be thought of as interpolation of $\theta_j(n)$ by a factor N to $\theta''_j(n)$ followed by filtering through $F_j(z)$ having impulse response coefficients

$$f_j(i) = b_{ij}.\tag{9}$$

Fig. 2 shows the process of block orthogonal transform realized as a filter bank. Since BOT is a nonredundant perfect MAKUR: BOT'S BASED ON NONUNIFORM FILTER BANKS



reconstruction system, (2)–(4) should be satisfied. Since $f_n(i)$ are of length N and orthonormal, they satisfy

$$\sum_{i=-\infty}^{\infty} f_n(i) f_m^*(i-kN) = \begin{cases} 1, & \text{if } m=n, k=0\\ 0, & \text{else} \end{cases}$$
(10)

which is the orthonormality condition for the basis sequences of an uniform filter bank. Orthonormal filters are shown to be paraunitary in [6, Th. 4]. We present a simpler proof here for BOT's. Since m_n equals N for all n, (2) is satisfied, and any BOT is a critically sampled system. Now, (3) may be expressed as N^2 equations:

$$\sum_{i=0}^{N-1} H_m(ze^{-j2\pi i/N}) H_n(z^{-1}e^{j2\pi i/N}) = \begin{cases} N, & \text{if } m = n \\ 0, & \text{else} \end{cases}$$
(11)

For a BOT, length of each filter is N, and the left-hand side of (11) consists of terms of the following form:

$$h_{n}(p)h_{m}(q)z^{q-p}\sum_{i=0}^{N-1}e^{-j2\pi i(p-q)/N} = \begin{cases} Nh_{n}(p)h_{m}(p), & \text{if } p = q\\ 0, & \text{else.} \end{cases}$$
(12)

Therefore, the left-hand side of (11) simplifies to

$$N\sum_{p=0}^{N-1}h_n(p)h_m(p) = \begin{cases} N, & \text{if } m = n\\ 0, & \text{else} \end{cases}$$
(13)

from the orthonormality condition, and hence, the filters $H_j(z)$ are paraunitary. From (8) and (9) it is clear that the synthesis filters also satisfy (4).

Since the decimation factor m_j of a conventional BOT is always N, it is expected that the equivalent filters would constitute a uniform filter bank. This can be verified for the popular transform. The discrete Fourier transform (DFT) is given by

$$h_n(i) = b_{N-1-i,n} = \frac{1}{\sqrt{N}} e^{-j2\pi n(N-1-i)/N}.$$
 (14)

Note that $h_0(i)$ is a (real) constant for all *i*, hence, $H_0(z)$ is a (symmetric) low-pass filter. From (14), $H_n(z)$ for $n \neq 0$ is

$$H_n(z) = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} e^{-j2\pi(N-1-i)n/N} z^{-i}$$
$$= \frac{1}{\sqrt{N}} e^{j2\pi n/N} \sum_{i=0}^{N-1} e^{j2\pi in/N} z^{-i}$$
(15)

which turns out to be

$$H_n(z) = e^{j2\pi n/N} H_0(z e^{-j2\pi n/N})$$
(16)

or a modulated version of $H_0(z)$, shifted on the frequency scale by $(2\pi n/N)$. For $n \neq 0$ the response is not symmetric, hence the coefficients are complex.

Taking another case, the discrete cosine transform (DCT) is given by

$$b_{in} = \sqrt{\frac{2}{N}} \alpha(n) \cos \frac{(2i+1)n\pi}{2N} \tag{17}$$

where $\alpha(0) = 1/\sqrt{2}$ and $\alpha(n) = 1$ for $n \neq 0$. Thus $H_0(z)$ in this case is identical to the discrete Fourier case. Further, from (8) and (17), $H_n(z)$ for $n \neq 0$ is

$$H_{n}(z) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \cos \frac{(2N-2i-1)n\pi}{2N} z^{-i}$$

$$= \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} (e^{j\pi(2N-2i-1)n/2N} + e^{-j\pi(2N-2i-1)n/2N}) z^{-i}$$

$$= \frac{1}{\sqrt{2N}} e^{j\pi n} \left[e^{j\pi n/2N} \sum_{i=0}^{N-1} e^{j\pi i n/N} z^{-i} + e^{-j\pi n/2N} \sum_{i=0}^{N-1} e^{-j\pi i n/N} z^{-i} \right]$$

$$= \frac{1}{\sqrt{2}} e^{j\pi(2N+1)n/N} H_{0}(ze^{-j\pi n/N})$$

$$+ \frac{1}{\sqrt{2}} e^{j\pi(2N-1)n/N} H_{0}(ze^{j\pi n/N}). \quad (18)$$

Neglecting the phase terms, the filter is the sum of two symmetrically modulated versions of the lowpass filter, shifted on either side of the frequency scale by $(\pi n/N)$. Since the response is symmetric, the coefficients are real. Note that well-known modulated filter bank or pseudo-QMF filter bank has a similar form [7],

$$h_n(i) = 2h(i)\cos\left[\frac{\left(n - \frac{L-1}{2}\right)(2i+1)\pi}{2N} + \theta_n\right]$$
(19)

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 44, NO. 8, AUGUST 1996





Fig. 4. (a) A tree-structured filter bank. (b) Equivalent nonuniform filterbank. (c) Equivalent BOT system.

where h(i) is a low-pass prototype, and L = 2mN is the length of the filters. It is known that choosing h(i) to be a constant and θ_n to be $\pi/2$ not only satisfies the perfect reconstruction condition but also is a good choice for small m[7, Eqs. (30), (34), (35)]. Thus the discrete cosine transform turns out to be an extension of cosine-modulated perfect reconstruction filter bank for L = N.

Discrete Hadamard transform (DHT) is another known BOT existing only for integer powers of 2, using the following recursion formula:

$$A_{2^{n}} = \frac{1}{\sqrt{2^{n}}} \begin{bmatrix} A_{2^{n-1}} & A_{2^{n-1}} \\ A_{2^{n-1}} & -A_{2^{n-1}} \end{bmatrix}$$
(20)

for nonnegative integer n, where $A_1 = 1$. This turns out to be a binary tree structured filter bank with A_2 decomposing each node, where the basic filters are $G_0(z) = 0.707 + 0.707z^{-1}$ (low pass) and $G_1(z) = z^{-1}G_0(-z^{-1})$ (highpass) [5]. The recursion of (20) produces the *k*th filter

$$H_k(z) = G_{i_0}(z)G_{i_1}(z^2)G_{i_2}(z^{2^2})\cdots G_{i_{n-1}}(z^{2^{n-1}})$$
(21)

where $\{i_0i_1\cdots i_{n-1}\}$ is the binary representation of k. Since this is repeated splitting of each band into two halves, the resulting bands are uniform.

The KLT is signal-dependent, hence cannot be considered in general. However, for an AR(1) process with autoregressive coefficient $b_1 \rightarrow 1$ the eigenfrequencies are asymptotically uniformly distributed from 0 to π , and the eigenfrequencies somewhat denote the center frequencies of the filters. Fig. 3 shows the magnitude responses of the filter banks obtained for various BOT's for N = 8. The KLT is for an AR(1) process with coefficient $b_1 = -0.95$ (example 3).

(b)

III. NONUNIFORM BOT

On the other hand, it is not always possible to represent any critically sampled perfect reconstruction subband system by an equivalent BOT. In [5], any perfect reconstruction uniform filter bank is folded to generate a BOT, which turns out to be same or very similar to DHT. In Appendix I, we show that the folding works only for equal decimation ratios and not for unequal cases.

However, with one restriction, we can achieve the equivalence. The length of each filter $H_j(z)$ may at most be the decimation factor m_j . If the length exceeds m_j , then subsequent samples of the encoded sequence $\theta_j(n)$ would not be independent even if input sequence x(n) were memoryless, and we cannot express them as results of memoryless



Fig. 5. Equivalent filter bank of Fig. 1.

forward transform of (5). However, we may keep our attention only to the case where the length is equal to m_i since a length smaller than m_i is possible only in cases separable to smaller transform sizes. For example, consider the treestructured filter bank shown in Fig. 4(a). This would be a perfect reconstruction system if each component system is a perfect reconstruction system. Assume the length of the filters $H_{0i}(z)$ to be p and that of $H_{1i}(z)$ to be q. This system has p + q - 1 nonuniform bands, which may be shown as an equivalent nonuniform filter bank of Fig. 4(b). Here, $H_i(z) =$ $H_{00}(z)H_{1i}(z^p)$ for i=0 to q-1, and $H_i(z)=H_{0,i-q+1}(z)$ for i = q to p+q-2. Note that length of $H_{1i}(z^p)$ is (q-1)p+1; therefore, the length of $H_i(z)$ would be pq for i = 0 to q - 1and p for the rest. Fig. 4(c) shows another equivalent form of the same filter bank where all decimation factors have been made equal. Here, $G_i(z) = H_i(z)$ for i = 0 to q - 1, and for i > q - 1, the filters are given by

$$G_i(z) = z^{-(i \mod q)p} H_{q-1+|i/q|}(z).$$
(22)

Note that each filter is of length pq or smaller, and there are pq bands with a decimation factor of pq. This being a perfect reconstruction system, the filters $G_i(z)$ must satisfy equation (13). Hence, their impulse response vectors are orthonormal, and can be expressed as a BOT of size pq. In a similar fashion, any general tree-structured perfect reconstruction filter bank may be expressed as a BOT, so long as it satisfies the constrain on filter length.

It turns out that a wider class of filter banks may be expressed as BOT. If the basis functions of a filter bank as shown in Fig. 1 are orthonormal and of length equal to respective decimation factor, then it is equivalent to a BOT. Orthonormality of basis functions (of arbitrary length) in the context of wavelet packets is given by a modified form of (10)

$$\sum_{i=-\infty}^{\infty} f_n(i-pm_n)f_k^*(i-qm_k) = \begin{cases} 1, & \text{if } n=k, p=q\\ 0, & \text{else.} \end{cases}$$
(23)

Taking $\hat{N} = \operatorname{lcm}\{m_0, m_1, \cdots, m_{N-1}\}$, define a set of synthesis filters

$$\hat{F}_n(z) = z^{-nm_0} F_0(z), \quad \text{for } 0 \le n < (\hat{N}/m_0)$$
$$\hat{F}_{(\hat{N}/m_0)+n}(z) = z^{-nm_1} F_1(z), \quad \text{for } 0 \le n < (\hat{N}/m_1) \quad (24)$$

and so on. Each filter is associated with an interpolation factor of \hat{N} . Note that this system, shown in Fig. 5, is equivalent to the system of Fig. 1. Clearly, number of such filters would

be \hat{N} since m_n 's satisfy (2). Also, length of each filter is \hat{N} or less. All filters derived from a single $F_n(z)$ are clearly orthonormal. Now consider two filters $\hat{F}_n(z)$ and $\hat{F}_k(z)$ such that $n < (\hat{N}/m_0)$ and $k = (\hat{N}/m_0) + l$ where $l < (\hat{N}/m_1)$. It follows that

$$\sum_{i=-\infty}^{\infty} \hat{f}_n(i)\hat{f}_k^*(i) = \sum_{i=-\infty}^{\infty} f_0(i-nm_0)f_1^*(i-lm_1) = 0$$
(25)

from (23), hence they are orthonormal. Therefore, the system can be expressed as a BOT of size \hat{N} . Since a tree-structured perfect reconstruction filter bank forms an orthonormal basis [6, Th. 3], it follows that any such filter bank with proper lengths is equivalent to a BOT. The class of wavelet packets with orthonormal basis, however, is more general than treestructured filter banks, as shown in the following example.

Example 1: Consider a depth-2 binary tree-structured system with

$$H_0(z) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}z^{-1}$$
 and $H_1(z) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}z^{-1}$.

This is a filter bank with four uniform bands, as shown in Fig. 6(a). Suppose we wish to have three bands as shown in Fig. 6(b). Noting that $H_1(z^2)$ has the desirable shape of the middle band, we may use it. Since $H_1(z^2)$ has length 3, $z^{-2}H_1(z^2)$ cannot be used; instead, $z^{-1}H_1(z^2)$ has to be used. Therefore, the equivalent filters are

$$G_{0}(z) = H_{0}(z)H_{0}(z^{2})$$

$$G_{1}(z) = H_{1}(z^{2})$$

$$G_{2}(z) = z^{-1}H_{1}(z^{2})$$

$$G_{3}(z) = H_{1}(z)H_{0}(z^{2}).$$
(26)

This is a perfect reconstruction system with three bands, but cannot be expressed as a tree structure. The reason it is perfect reconstruction is as follows. The second and third bands in Fig. 6(a) are $H_0(z)H_1(z^2)$ and $H_1(z)H_1(z^2)$. For our choice of initial filters, note that

$$G_{1}(z) = \frac{H_{0}(z)H_{1}(z^{2}) + H_{1}(z)H_{1}(z^{2})}{\sqrt{2}}$$
$$G_{2}(z) = \frac{H_{0}(z)H_{1}(z^{2}) - H_{1}(z)H_{1}(z^{2})}{\sqrt{2}}.$$
 (27)

Since the filters of Fig. 6(a) are perfect reconstruction, from (27) it follows that the filters of (26) are also perfect reconstruction.

The above example was motivated from [6, fig. 13], where another example of a perfect reconstruction filter bank that cannot be expressed as a tree structure has been provided. Appendix II shows the transform matrix obtained for that example.

IV. OPTIMUM TREE STRUCTURE FOR A SOURCE

Since a nonuniform BOT matrix can be constructed from any tree-structured filter bank with restricted length, a large number of choices exist for designing a BOT of a given size. In this section we discuss how one may go about choosing IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 44, NO. 8, AUGUST 1996



the BOT which is expected to produce maximum transform coding gain.

In the context of signal compression, maximum coding gain (in dB) for an orthogonal transform A of size $N \times N$ is given by [1]

$$G_{\mathbf{A}} = \log\left(\frac{1}{N}\sum_{k=0}^{N-1}\sigma_{k}^{2}\right) - \frac{1}{N}\sum_{k=0}^{N-1}\log\sigma_{k}^{2}$$
$$= \log\sigma_{x}^{2} - \frac{1}{N}\sum_{k=0}^{N-1}\log\sigma_{k}^{2}$$
(28)

where σ_k^2 is the variance of $\theta_k(n)$ of Fig. 2, and σ_x^2 is the input signal variance, which is independent of A. Therefore, in order to maximize G_A , one needs to minimize $\sum_{k=0}^{N-1} \log \sigma_k^2$. The design procedure, therefore, involves estimating σ_k^2 , and finding the tree structure that minimizes the above cost function.

Note that each level of a tree-structured filter bank consists of an uniform filter bank. Assuming brickwall filters, the passband for some resulting filter $H_n(z)$ (say, of Fig. 4(b), N = pq) spans from $s\pi/N$ to $t\pi/N$ for some integers $0 \le s < t \le N$. Consider an uniform filter bank of N bands. We can estimate σ_k^2 for this case as

$$\sigma_k^2 = \int_{k\pi/N}^{(k+1)\pi/N} S_{xx}(e^{j\omega}) \frac{d\omega}{\pi}$$
(29)

where $S_{xx}(e^{j\omega})$ is the power spectral density of x(n). Then, $\sigma_n^{\prime 2}$ for $H_n(z)$ is given by

$$\sigma_n^{\prime 2} = \sum_{i=s}^{t-1} \sigma_i^2. \tag{30}$$

From equation (22), $\sigma_m''^2$ for the filter $G_m(z)$ of Fig. 4(c), which is derived from $H_n(z)$, would then be

$$\sigma_m''^2 = \frac{1}{t-s} \sigma_n'^2 = \frac{1}{t-s} \sum_{i=s}^{t-1} \sigma_i^2.$$
(31)

This is the key equation to be used for estimating the transformed coefficient variances.

Any tree-structured filter bank may be considered as a pruned subtree of a full tree (uniform filter bank). For example, the full tree corresponding to the filter bank in Fig. 4(a) would be a depth-2 tree, with first level splitting to p nodes and

second level splitting to q leaves each. Note that each leaf of the full tree should have equal decimation factor, but it is not necessary that they be at equal depth. The number of leaves in a pruned subtree refers to the number of bands (channels) in the subband system. Thus the problem becomes of finding the pruned subtree with a given number of leaves that minimizes the cost function mentioned above. The pruning process involves replacing all the children of a node by the node itself, when all children were leaves. This is equivalent to eliminating a "level" of filter banks from a branch. The minimization can be achieved using the generalized version of Breiman, Friedman, Olshen, and Stone (BFOS) algorithm, which is a greedy (hence, simple) but optimal algorithm to optimize any monotonic linear/affine tree functional. Using the notation of [2, p. 640 ff.], denote by T the full tree, and by \tilde{T} the set of leaves of T. Let S be a pruned subtree with \tilde{S} its set of leaves. Define two tree functionals (cost functions) associated with a leaf t as $u_1(t) = 1$ and $u_2(t) = \log \sigma_t^2$. where σ_t^2 is the estimated coefficient variance for the band corresponding to t. The cost of a tree or subtree is the sum of the costs of its leaves, for example,

$$u_i(T) = \sum_{t \in \tilde{T}} u_i(t) \tag{32}$$

for i = 1, 2. Note that for this choice, $u_1(T)$ is the number of bands and $u_2(T)$ is the coding gain or the desired cost function. Pruning involves removing all children t_1, \ldots, t_n of some internal node t' when each child t_i was a leaf, and changing t' to a leaf. Using (30) and (31), the tree functionals for such an internal node t' may be defined as

$$u_1(t') = 1, \quad u_2(t') = n \log\left[\frac{1}{n} \sum_{i=1}^n e^{u_2(t_i)}\right].$$
 (33)

Thus, $u_1()$ and $u_2()$ maintains their respective interpretation even for a pruned subtree. To see that $u_2(t')$ is the contribution of t' to coding gain, observe that $\Sigma \sigma_{t_i}^2$ denotes that t' has a passband width equal to the sum of passband widths of t_i s, the scaling 1/n comes because there will be n times more coefficients from t' than from t_i , and the outer factor of n denotes that length of the filter for t' is n times smaller than that for t_i s (hence, the number of basis vectors t' will contribute is n times more than each t_i contributes). Note that



Fig. 7. A pruned subtree.

if S_2 is a subtree of S_1 , then $|\tilde{S}_2| < |\tilde{S}_1|$, hence $u_1()$ is a monotonically increasing functional. In addition, since

$$u_{2}(t') = n \log \left[\frac{1}{n} \sum_{i=1}^{n} \sigma_{t_{i}}^{2}\right]$$
$$\geq n \log \left[\left(\prod_{i=1}^{n} \sigma_{t_{i}}^{2}\right)^{1/n}\right] = \sum_{i=1}^{n} u_{2}(t_{i}) \qquad (34)$$

 $u_2()$ is a monotonically decreasing functional (coding gain falls with less number of bands). Now, for a prunable node t', the slope

$$\lambda(t') = -\Delta u_2(t') / \Delta u_1(t') \tag{35}$$

(where Δ denotes change due to pruning at t') denotes the increase in the coding gain per reduced band. The generalized BFOS algorithm may now be applied which, at every step, prunes at the node t that has the smallest $\lambda(t)$. There is a slight addition required if we desire to find the best pruned subtree with an arbitrary N leaves. At each step, some pruning may not be allowed if that results in a number of leaves from which N cannot be achieved. For example, consider the subtree in the nth iteration to be the tree in Fig. 7, having number of leaves $N_n = 15$ while the desired number of leaves is, say, N = 11. Therefore, we cannot prune at either t_1 of t_2 , since that would result in $N_{n+1} = 12$ from which N = 11 is unachievable by any pruning. While resolving whether to allow pruning or not is nontrivial in general, for most cases of interest it is likely to become trivial, or can be found out by a systematic search.

For the sake of completeness, we are reproducing the algorithm that finds the best pruned subtree with a desirable number of leaves for a given full tree with estimated coefficient variances.

Input Full tree T

Variance σ_t^2 for each leaf $t \in \tilde{T}$ Desired number of leaves N

Initialization Iteration index n = 0

- Tree $S_0 = T$, number of leaves $N_0 = |T|$
- Set of all prunable nodes $P = \{ \text{parent of } t \in \tilde{T} \}$ Array of number of leaves reduced by pruning at $p, D = \{\delta(p): p \in P\}$ where $\delta(p) = n - 1$ if p has n children
- Array of variances $V = \{\sigma_p^2 : p \in P\}$ using the relation:

$$\sigma_p^2 = \left[\frac{1}{n}\sum_{i=1}^n \sigma_{t_i}^2\right]^n \tag{36}$$

where t_1 to t_n are the children of p

Iteration-n If pruning at some $p \in P$ makes it

impossible to subsequently reach N leaves, remove p from $P, \delta(p)$ from D, and σ_p^2 from V

If P is empty, terminate

Find the node with minimum cost $p^* \in P$ such that, from (35),

$$\frac{1}{\delta(p)} \left[\log \sigma_p^2 - \sum_{i=1}^n \log \sigma_{t_i}^2 \right]$$
(37)

is minimized by p^*

Prune at p^* to get S_{n+1} from $S_n, N_{n+1} = N_n - \delta(p^*)$ Remove $p^*, \delta(p^*)$ and $\sigma_{p^*}^2$ from P, D and V

Say t is the parent node of p^* . Check if all children of t are leaves in S_{n+1} or not. If yes, compute σ_t^2 using (36) and add $t, \delta(t)$ and σ_t^2 to P, D and V

V. SIMULATION RESULTS

In this section we present design examples for a few sources for some given full tree and the desired number of bands, and compare the performance of the nonuniform BOT's with the conventional uniform BOT's. We shall use the following notations. A nonuniform BOT denoted by $[j_1 j_2 \cdots j_N]$ has N passbands ideally extending $\{0-j_1\pi/N_0\}, \{j_1\pi/N_0-(j_1+1)\}, \{j$ $(j_2)\pi/N_0$, etc., with the full tree having N_0 leaves. For a uniform BOT, its subscript denotes the size. Thus, while DCT_6 is the DCT of size 6, $DCT_{2\times 3}$ is a full tree of depth-2 with first level splitting to 2 nodes using DCT_2 and second level splitting to 3 leaves each using DCT_3 . For each BOT, we show the block size $(N_0 \times N_0)$ in the second columns, and the computation required in terms of the average number of taps (multiplications) per sample in the third columns (which is more than or equal to $1/N_0$ the number of nonzero elements in the transform matrix). The special structure of transforms like DCT or DFT is not considered for either case (nonuniform and uniform).

The theoretical coding gain for each BOT, obtained using (28) but in absolute unit (without the logarithm), is shown in the fourth and fifth columns. For the fourth column, the variances are estimated using (29) which assumes brickwall filtering. For the fifth column, the actual variances are computed by simulating the source and using the BOT. Actual coding gain is shown in the last column. The simulated source is transformed and quantized. Optimal bit allocation, approximated to nonnegative integer, is done using the estimated variances for quantizers of various bands. All quantizers are pdf-optimized to Gaussian pdf (since all our sources are Gauss-Markov sources), and scaled by the square-root of the estimated variance of the band. The coding gain denotes the ratio of the reconstruction error variance in transform coding, to that in scalar quantizing, the same sequence using the same rate (in this case 3 b/sample). For KLT, uniform bands have been assumed.

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 44, NO. 8, AUGUST 1996





Fig. 9. Normalized power spectral density for Example 2.

Example 2: Consider an AR(1) process with coefficient $b_1 = 0.95$ and a Gaussian pdf for the input noise. Consider a full tree of depth-2 having six leaves, and the desired number of bands is N = 4. Note that there can be two alternate structures for T, shown in Fig. 8. The first alternative has two possible pruned subtrees with four leaves, $\begin{bmatrix} 3 & 1 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. The second alternative has three possible subtrees, $\begin{bmatrix} 2 & 2 & 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$ $1 \ 2$ and $\begin{bmatrix} 1 & 1 & 2 & 2 \end{bmatrix}$. The power spectral density for this source is shown in Fig. 9. Integrating it according to (29) gives the normalized variances for the full tree as {1,0.0276,0.0098,0.0056,0.0040,0.0035}. Running the generalized BFOS algorithm selects [1 1 1 3] in the first case, and $\begin{bmatrix} 1 & 1 & 2 & 2 \end{bmatrix}$ in the second case. Since the filters used for constructing the nonuniform BOT's are BOT_2 (for size 2, all known BOT's are same) and DCT_3 , the full trees of Fig. 8 are $DCT_{2\times 3}$ and $DCT_{3\times 2}$. We have also included the performance of KLT of size 4 (since N = 4). Table I shows the performance comparison. Since approximating the optimal bit allocation to integer value affects different BOT's differently, we don't see an exact correspondence between theoretical and simulated gains. However, the optimized subtree performs better than other subtrees for both alternatives. Some nonuniform BOT's perform better than DFT₆, a uniform BOT of equal size. Nonuniform transforms have comparatively lower number of taps per sample. The BOT matrix for the best nonuniform case, [1 1 1 3], is given below, and the magnitude response of the four filters is shown in Fig. 10.

	[0.408]	0.408	0.408	0.408	0.408	ך 0.408	
	0.5	0.5	0	0	-0.5	-0.5	
A =	0.289	0.289	-0.577	-0.577	0.289	0.289	
	0.707	-0.707	0	0	0	0	
	0	0	0.707	-0.707	0	0	
	L 0	0	0	.0	0.707	-0.707	
						(3	38)



Fig. 10. Magnitude response of the nonuniform BOT of (38).

 TABLE I

 Performance Comparison for Example 2

computed) 3.24	2.40
3.24	2 4 2
3.24	2 4 2
	5.42
6.28	5.66
4.79	4.71
4.80	4.60
6.34	5.25
7.10	6.55
7.06	6.51
6.28	5.57
6.34	5.21
5.68	4.90
5.83	5.81
_	6.28 4.79 4.80 6.34 7.10 7.06 6.28 6.34 5.68 5.83

Example 3: Consider another AR(1) process with coefficient $b_1 = -0.95$ with Gaussian input noise. T is the full binary tree of depth-3 $(N_0 = 8)$ and N is four. There are 5 possible pruned subtrees— $\begin{bmatrix} 1 & 1 & 2 & 4 \end{bmatrix}$, [2] $[4 \ 2 \ 1 \ 1]$ $1 \ 1 \ 4$], $[4 \ 1 \ 1 \ 2],$ and |2|2 2 2]. The generalized BFOS algorithm chooses 2 1 1] as the best subtree. Here the full tree is [4]DHT₈, and we have also included KLT₄. Table II shows the performance comparison for each subtree as well as for uniform BOT's. Note that here the 4-band nonuniform transform optimized to the source ([4 2 1 1]) performs better than the 4-band optimal transform (KLT_4) for that source. This is not a contradiction since for N-band nonuniform transform, the filter lengths are generally longer than N. Further, the best nonuniform BOT outperforms all but the optimal of the known uniform BOT's of equal size DCT_8 , DFT_8 and DHT_8 —using only half the number of taps on an average. This is because, DCT is known to perform poorly when b_1 is negative. DFT_n actually has only n/2 + 1 bands. While DHT should have performed as good as the nonuniform BOT, approximations in bit allocation has marginally lowered its performance. Nonuniform transforms have same or lower number of taps per sample than uniform transforms. The BOT matrix for the best nonuniform case, $\begin{bmatrix} 4 & 2 & 1 & 1 \end{bmatrix}$, is given in (39) at the bottom of the next page, whose magnitude response is shown in Fig. 11. Note that this is an example of block wavelet transform with dyadic tree.

Example 4: Now we take an AR(2) process with coefficients $b_1 = 0.1, b_2 = 0.56$ (poles at -0.7, 0.8). Take



Fig. 11. Magnitude response of the nonuniform BOT of (39).

 TABLE II

 Performance Comparison for Example 3

Transform	Size	Taps per Sample	r Theoretical Gair		Simulated Gain		
			(estimated)	(computed)			
Nonuniform							
[1 1 2 4]	8×8	4	3.48	3.32	3.68		
[2 1 1 4]	8×8	4	3.49	3.31	3.67		
[4 1 1 2]	8×8	4	5.98	5.54	6.57		
[4 2 1 1]	8×8	4	7.21	6.67	7.96		
[2 2 2 2]	8×8	4	5.96	5.54	6.56		
Uniform							
KLT ₈	8×8	8	7.35	8.31	9.20		
DCT ₈	8×8	8	7.35	3.37	1.06		
DFT ₈	8×8	8	7.09	6.32	6.49		
DHT ₈	8×8	8	7.35	6.67	7.80		
KLT_4	4×4	4	4.75	6.10	6.71		



Fig. 12. Magnitude response of the nonuniform BOT of (40).

 TABLE III

 PERFORMANCE COMPARISON FOR EXAMPLE 4

Transform	Size	Taps per Sample	Theoretic	Simulated Gain	
		_	(estimated)	(computed)	
Nonuniform					
[2 1 1]	4×4	3	1.16	1.10	1.00
[1 2 1]	4×4	3	1.42	1.27	1.35
[1 1 2]	4×4	3	1.29	1.19	1.26
Uniform					
KLT ₄	4×4	4	1.42	1.27	1.29
DCT ₄	4×4	4	1.42	1.23	1.27
DFT_4	4×4	4	1.42	1.27	1.35
DHT ₄	4×4	4	1.42	1.27	1.29
KLT ₃	3 × 3	3	1.31	1.19	1.22

VI. CONCLUSION

 $N_0 = 4$ and desired bands N = 3. Possible systems are $\begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 2 \end{bmatrix}$, of which the second one may not be generated using a tree structure, but the transform matrix of Example 1 may be used. Table III compares the performances of each subtree as well as of uniform BOT's. Here the best nonuniform BOT, $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$, performs equal to the best uniform BOT, DFT₄. In fact, they happen to be the same since (40) is DF'T₄ of $\begin{bmatrix} 1, Eq. (12.150) \end{bmatrix}$. KLT₄ fails to perform as well due possibly to integer bit allocation. Thus in this case the nonuniform BOT equals or betters any other BOT performance. The resulting transform matrix for $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ is given in (40), whose magnitude response is shown in Fig. 12.

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.707 & 0 & -0.707 & 0 \\ 0 & 0.707 & 0 & -0.707 \\ 0.5 & -0.5 & 0.5 & -0.5 \end{bmatrix}$$
(40)

It has been shown that any BOT can be expressed as a perfect reconstruction filter bank. The reverse, however, is not true. When the length of each filter does not exceed its decimation factor, then it has been shown that any orthonormal (or tree-structured perfect reconstruction) filter bank can be expressed as a BOT. A method to arrive at an equivalent BOT from any such filter bank has been presented. Popular BOT's such as DFT, DCT, and DHT have been shown to consist of uniform filter banks. The extra freedom in choosing a nonuniform filter bank can be exploited to design a BOT appropriate for a given source. For this purpose, a simple way to estimate the coding gain has been suggested, and a known algorithm has been used to find the optimal tree structure.

The results from a few case studies allow us to draw the following conclusions. The estimates for relative coding gain of nonuniform BOT's match with simulation results. Nonuniform transform performs same or better than the popular uniform

	г0.707	0.707	0	0	0	0	0	ך 0
	0	0	0.707	0.707	0	0	0	0
	0	0	0	0	0.707	0.707	0	0
	0	0	0	0	0	0	0.707	0.707
A =	0.5	-0.5	-0.5	0.5	0	0	0	0
	0	0	0	0	0.5	-0.5	-0.5	0.5
	0.354	-0.354	0.354	-0.354	-0.354	0.354	-0.354	0.354
	0.354	-0.354	0.354	-0.354	0.354	-0.354	0.354	-0.354

Γ^a	a	a	a	a	a	0	0	0	0	0	0	0	0	0	0	0	ך 0
0	0	0	0	0	0	a	a	a	a	a	a	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	a	a	a	a	a	a
e	-h	e	-f	-g	-f	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	e	$^{-h}$	e	-f	-g	-f	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	e	$^{-h}$	e	-f	-g	-f
a	0	-a	a	0	-a	a	0	-a	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	a	0	-a	a	0	-a	a	0	-a
b	0	-b	0	0	0	-b	0	b	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	b	0	-b	0	0	0	-b	0	b
c	0	-c	-d	0	d	c	0	-c	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	c	0	-c	-d	0	d	c	0	-c
f	g	f	-e	h	-e	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	f	g	f	-e	h	-e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	f	g	f	-e	h	-e
c	-d	c	-c	d	-c	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	c	-d	c	-c	d	-c	0	0	0	0	0	0
LO	0	0	0	0	0	0	0	0	0	0	0	c	-d	c	-c	d	$-c \rfloor$

(46)

transforms having same size (more bands) in most cases, while requiring same or lower number of taps per sample than the latter. Nonuniform BOT performs better than the optimum uniform BOT having equal number of bands (smaller size) in most cases, too. Therefore, nonuniform BOT's are seen to be very promising for compression of signals. It would be of interest to explore such nonuniform transforms for compression of real signals in fixed or adaptive manner.

APPENDIX I

In [5], a set of filter banks as in Fig. 1 with equal decimation ratios $m_i = N$ is considered. The filters $H_n(z)$ are of length K. Since K may be larger than N, the filters are folded to length N in order to generate a BOT. If $\hat{H}_n(z)$ denotes the folded filter, then

$$\hat{h}_n(i) = \sum_j h_n(i+jN) \tag{41}$$

for i = 0 to N - 1. Without loss of generality, assume K = MN. Define

$$\boldsymbol{h}_{n}^{(m)} = [h_{n}(mN) \quad h_{n}(mN+1) \quad \cdots \quad h_{n}(mN+N-1)]$$
(42)

for m = 0 to M - 1. Since $H_n(z)$ are perfect reconstruction bank, they satisfy relation (10). To show that $\hat{H}_n(z)$ are also perfect reconstruction bank, note that

$$\left(\sum_{m=0}^{M-1} oldsymbol{h}_n^{(m)}
ight) \cdot \left(\sum_{m=0}^{M-1} oldsymbol{h}_n^{(m)}
ight)^t$$

$$= \sum_{m=0}^{M-1} \sum_{p=0}^{M-1} \boldsymbol{h}_{n}^{(m)} \boldsymbol{h}_{n}^{(p)t}$$

$$= \sum_{m=0}^{M-1} \sum_{p=0}^{M-1} \boldsymbol{h}_{n}^{(m)} \boldsymbol{h}_{n}^{((m-p) \mod M)t}$$

$$= \sum_{p=0}^{M-1} \left[\sum_{i} h_{n}(i)h_{n}(i+pN) \right]$$

$$+ \sum_{i} h_{n}(i)h_{n}(i-(M-p)N) \right]$$

$$= \sum_{i} h_{n}(i)h_{n}(i)$$

$$= 1.$$
(43)

Similarly, when $n \neq k$, we get

$$\begin{pmatrix} \sum_{m=0}^{M-1} \boldsymbol{h}_{n}^{(m)} \end{pmatrix} \cdot \left(\sum_{m=0}^{M-1} \boldsymbol{h}_{k}^{(m)} \right)^{t} = \sum_{p=0}^{M-1} \left[\sum_{i} h_{n}(i) h_{k}(i+pN) + \sum_{i} h_{n}(i) h_{k}(i-(M-p)N) \right]$$

= 0. (44)

For a nonuniform filter bank, the decimation ratios not equal. Hence, folding does not preserve are reconstruction example, the perfect property. For filters $H_1(z) =$ 0.1294 0.2241 -0.83650.4830decimation ratio 2, and $H_0(z)H_1(z^2)$ with = $[0.0625 \ 0.1082 \ 0.1372 \ 0.1707 \ -0.3538 \ -0.7287 \ 0.0458$ 0.5123 0.1082 0.0625] with decimation ratio 4 (obtained from Daubechies basis N = 2[8]), are orthonormal. But $\hat{H}_1(z) = [-0.7071]$ [0.7071] and $H_0(z)H_1(z^2) = [-0.1831 - 0.6830 0.1831 0.6830]$ are clearly not orthonormal. The reason is, folding does not preserve the shape (passband) of a filter. Since folding of an uniform filter bank preserves orthonormality, the result is another uniform filter bank (even though some filters switch bands). For a nonuniform filter bank, switching of bands results in nonorthogonality after folding.

APPENDIX II

For the following choice of initial filters

$$A_{0}(z) = B_{0}(z) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}z^{-1}$$

$$A_{1}(z) = B_{1}(z) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}z^{-1}$$

$$C_{0}(z) = D_{0}(z) = \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{3}}z^{-1} + \frac{1}{\sqrt{3}}z^{-2}$$

$$C_{1}(z) = D_{1}(z) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}z^{-2}$$

$$C_{2}(z) = D_{2}(z) = \frac{1}{\sqrt{6}} - \frac{2}{\sqrt{6}}z^{-1} + \frac{1}{\sqrt{6}}z^{-2}$$
(45)

the filter bank of [6, Fig. 13] is equivalent to the following BOT of size 18×18 following relation (24), as shown in (46) at the top of the previous page. Here

$$\begin{split} &a = \frac{1}{\sqrt{6}}, \quad b = \frac{1}{2}, \quad c = \frac{1}{\sqrt{12}}, \quad d = \frac{1}{\sqrt{3}}, \\ &e = \frac{a+c}{\sqrt{2}}, \quad f = \frac{a-c}{\sqrt{2}}, \quad g = \frac{d+a}{\sqrt{2}}, \quad \text{and} \quad h = \frac{d-a}{\sqrt{2}}. \end{split}$$

REFERENCES

- N. S. Jayant and P. Noll, Digital Coding of Waveforms-Principles and Applications to Speech and Video. Englewood Cliff, NJ: Prentice Hall, 1984.
- [2] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Boston, MA: Kluwar Academic, 1992.
- [3] P. P. Vaidyanathan, Multirate Systems and Filter Banks. Englewood Cliff, NJ: Prentice Hall, 1993.
- [4] M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207–2232, Sept. 1992.
- [5] A. E. Cetin, O. N. Gerek, and S. Ulukus, "Block wavelet transforms for image coding," *IEEE Trans. Circuits Syst. Video Technol.* vol. 3, pp. 433–435, Dec. 1993.
- [6] A. K. Soman and P. P. Vaidyanathan, "On orthonormal wavelets and paraunitary filter banks," *IEEE Trans. Signal Processing*, vol. 41, pp. 1170-1183, Mar. 1993.
- [7] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, Apr. 1992.
- [8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Comm. Pure Appl. Math., vol. 41, pp. 909–996, Nov. 1988.



Anamitra Makur received the B.Tech. degree in electronics and electrical communication engineering from Indian Institute of Technology, Kharagpur, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1986 and 1990, respectively.

Since 1990, he has been with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, as an Assistant Professor. His research in source coding includes vector quantization and image vector quantization, trans-

forms, logic coding, and video motion field coding. His interests in channel coding include nonlinear code design and decoding algorithms/architectures for block codes. His interests in image/video processing include image halftoning, multidimensional filter design, and television video processing.