EXTENDED LAPPED TRANSFORMS: PROPERTIES, APPLICATIONS, AND FAST ALGORITHMS

Henrique S. Malvar, *Senior Member, IEEE*

Dept. of Electrical Engineering, Universidade de Brasília
CP 153041, 70910 Brasília, DF, Brazil

Abstract – The family of lapped orthogonal transforms is extended to include basis functions of arbitrary length. Within this new family, the extended lapped transform (ELT) is introduced, as a generalization of the previously reported modulated lapped transform (MLT). Design techniques and fast algorithms for the ELT are presented, as well as examples that demonstrate the good performance of the ELT in signal coding applications. Therefore, the ELT is a promising substitute for traditional block transforms in transform coding systems, and also a good substitute for less efficient filter banks in subband coding systems.

Appeared in IEEE Trans. on Signal Processing, vol. 40, no. 11, pp. 2703–2714, Nov. 1992

1 Introduction

Transform coding (TC) and subband coding (SBC) are among the most effective techniques for the representation of signals such as speech and images [1] at medium bit rates. The two techniques are not fundamentally different, however. We can view a transform operator as a critically-decimated filter bank [2] where the basis functions of the transform are the impulse responses of the synthesis filters [3]. Similarly, we can view a filter bank as a transform with memory, in which the transform of a block may depend not only on that block but also on its neighboring blocks [4].

Both TC and SBC can be used for any given signal coding problem, but each one has its major drawbacks. In TC, the reduced length of the basis functions correspond to filters whose impulse response end abruptly, and this causes discontinuities in the reconstructed signal, the so-called blocking effects [5], [6]. In SBC, the filter impulse responses generally decay gradually to zero, and thus blocking effects are virtually not present, in spite of the multirate nature of SBC. However, SBC systems are generally restricted to work with a reduced number of bands, because of the computational complexity of the quadrature mirror filter (QMF) banks that are usually employed [1].

With the introduction of the lapped orthogonal transform (LOT), [7], [8], in which the basis functions have lengths given by $L = 2M$, where M is the transform size, a new alternative for the transform choice in TC became available. For a given transform length, the LOT has a higher coding gain than the usual discrete cosine transform (DCT), and the LOT is also free from blocking effects [3], [6]. The LOT basis functions also correspond to filters with much better bandpass responses than those of the DCT [3], [9]. Another

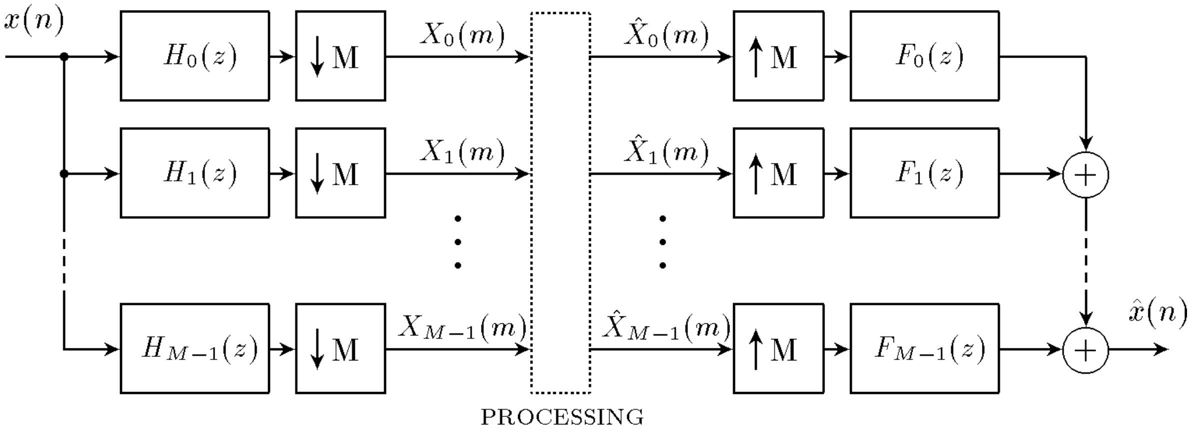

Figure 1: Subband signal processing with M-band critically decimated filter banks.

lapped transform with $L = 2M$ is the modulated lapped transform (MLT) [3], first developed in [10] with the denomination "oddly-stacked time domain aliasing cancellation (TDAC) filter bank" (the evenly-stacked TDAC [11] is not a lapped transform). The MLT is actually a TDAC in which the window is asymptotically optimal for coding highly correlated sources [3]. Another good choice for the window was presented in [12]. The MLT is similar to the LOT, the main differences being that the MLT basis functions are not symmetrical, and that the MLT is actually a cosine-modulated filter bank, which has slightly better bandpass responses than the LOT.

Lapped transforms like the LOT or the original MLT may not be adequate substitutes for QMF filter banks in applications where a strong subband separation is necessary, as in transmultiplexing and analog voice scrambling. This is because the basis functions have length $L = 2M$, which is much shorter than the usual lengths of QMF filters (which may go from $4M$ to $16M$). Thus, it would be of practical value if the lengths of the basis functions of lapped transforms could be made longer, without much penalty in the computational complexity.

The purpose of this paper is to present fast algorithms for the extended lapped transform (ELT), which is a generalization of the MLT with arbitrarily long basis functions. In Section 2 we review the ELT definition, first introduced in [13], and its perfect reconstruction property. In Section 3 we introduce the fast ELT implementation. Finally, in Section 4 we present some basic results of the application of the ELT to signal coding.

2 Extended Lapped Transforms

A lapped transform (LT) is a multirate filter bank in which the analysis and synthesis filters have a particular structure. Referring to Fig. 1, the analysis bank is composed of filters $H_0(z) \ldots H_{M-1}(z)$, and the synthesis bank is composed of filters $F_0(z) \ldots F_{M-1}(z)$. The system is critically decimated, since the down-sampling and up-sampling factors are equal to the number of subbands, M.

For a lapped transform, the analysis and synthesis filters are FIR, and they are defined from the columns of a matrix $\mathbf{P}$, in the form

$$f_k(n) = p_{nk}, \quad k = 0, 1, \ldots, M-1, \quad n = 0, 1, \ldots, NM-1 \tag{1}$$

and

$$h_k(n) = f_k(NM - 1 - n) \tag{2}$$

where p_{nk} is the element in the n-th row and k-th column of $\mathbf{P}$. Note that the impulse responses of the analysis filters are just the time-reversed versions of the synthesis responses.

The matrix $\mathbf{P}$ defines a lapped transform if and only if it satisfies the orthogonality conditions

$$\mathbf{P}'\mathbf{W}^m\mathbf{P} = \delta(m)\,\mathbf{I} \tag{3}$$

where $\mathbf{P}'$ denotes the transpose of $\mathbf{P}$, $\mathbf{I}$ is the identity matrix, $\delta(m)$ is the unitary impulse ($\delta(0) = 1, \delta(m) = 0, m \neq 0$), and $\mathbf{W}$ is the one-block shift matrix, defined by

$$\mathbf{W} \equiv \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4}$$

where the identity matrix above is of order $(N-1)M$.

The definition of $\mathbf{P}$ above is just an extension of the lapped orthogonal transform (LOT) in [3], [6]. The LOT is a particular LT, in which $N = 2$. We note that N specifies how many input blocks are used to compute an output block, in the direct transform. When $N = 1$, the LT reduces to a standard block transform, like the discrete cosine transform (DCT). We see then the reason for the name lapped transform: it is a generalization of a block transform, in which the input block is longer than the output block. Since the decimation factor (the block size) is always equal to M, we see that when $N > 1$ there is an overlap among the data blocks that are used to compute consecutive transform blocks. An alternative way of representing a signal processing system using lapped transforms is shown in Fig. 2. Note that, as usual in block signal processing, increasing indices at the input of the direct transform operator correspond to more recent samples.

A. Perfect Reconstruction

With the enforcement of the orthogonality conditions in (3), and also with the analysis and synthesis filter banks being defined by $\mathbf{P}'$ and $\mathbf{P}$, respectively, the transform/subband processing system of Fig. 2 has the perfect reconstruction (PR) property [14]. This means that the reconstructed signal is a delayed replica of the input signal, i.e. $\hat{x}(n) = x(n - NM + 1)$, if the subband signals are not modified, that is, if $\hat{X}_i(m) = X_i(m)$ in Fig. 2. Note that the time-domain orthogonality condition in (3) is a simplified form of the more generic conditions first presented in [4] and generalized in [15] and [16], since we have constrained the lapped transform to use the same matrix $\mathbf{P}$ both for the analysis and for the synthesis filter banks.

In order to verify the PR property for an LT, it is enough to show that its polyphase component matrix $\mathbf{E}(z)$ is lossless [14]. We can write the impulse response of each analy-

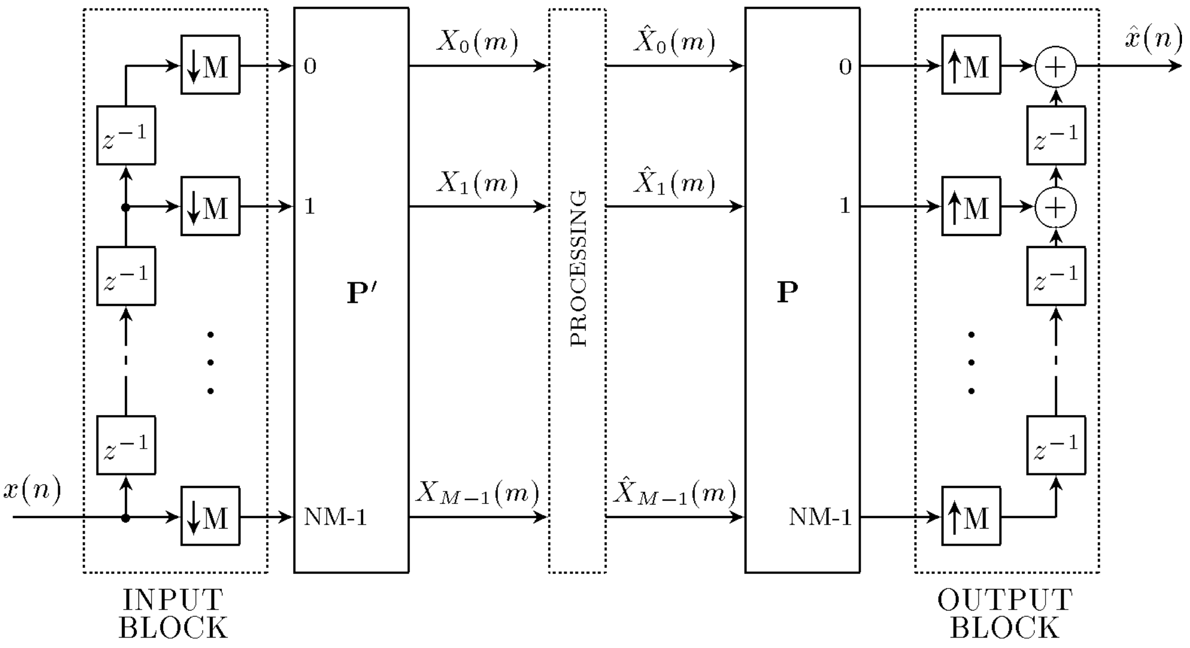

Figure 2: Structure equivalent of Fig. 1 for a lapped transform $\mathbf{P}$.

sis filter in terms of its polyphase components, in the form

$$H_k(z) = \sum_{r=0}^{NM-1} h_k(r) z^{-r} = \sum_{r=0}^{M-1} z^{-r} E_{kr}(z^M) \tag{5}$$

where the r-th polyphase component of $H_k(z)$ is given by

$$E_{kr}(z) = \sum_{m=0}^{N-1} p_{(N-m)M-1-r,k} \; z^{-m} \; . \tag{6}$$

Thus, the polyphase component matrix $\mathbf{E}(z)$ is given by

$$\mathbf{E}(z) = \sum_{l=0}^{N-1} z^{-(N-1-l)} \, \mathbf{P}'_l \, \mathbf{J} \tag{7}$$

where $\mathbf{P}_l$ is the l-th square block of $\mathbf{P}$, defined by $\mathbf{P}' \equiv [\mathbf{P}'_0 \; \mathbf{P}'_1 \; \cdots \; \mathbf{P}'_{N-1}]$, and $\mathbf{J}$ is the counter-identity matrix [3], i.e., $[\mathbf{J}]_{nk} = 1$ if $n + k = M - 1$, and zero otherwise, for $0 \leq n, k \leq M - 1$.

One of the advantages of the polyphase representation is that the decimators in Fig. 1 or Fig. 2 can be moved to the left of the analysis filter bank, and the interpolators to the right of the synthesis filter bank, as shown in Fig. 3. Note that, as usual in multirate DSP, when we increase r in $E_{kr}(z)$ we increase the delay associated with that polyphase component, i.e., we move back in time. This is why in the input block in Fig. 3 we have $x(n)$ coming in from the top. Due to the time-reversal relationship between the analysis

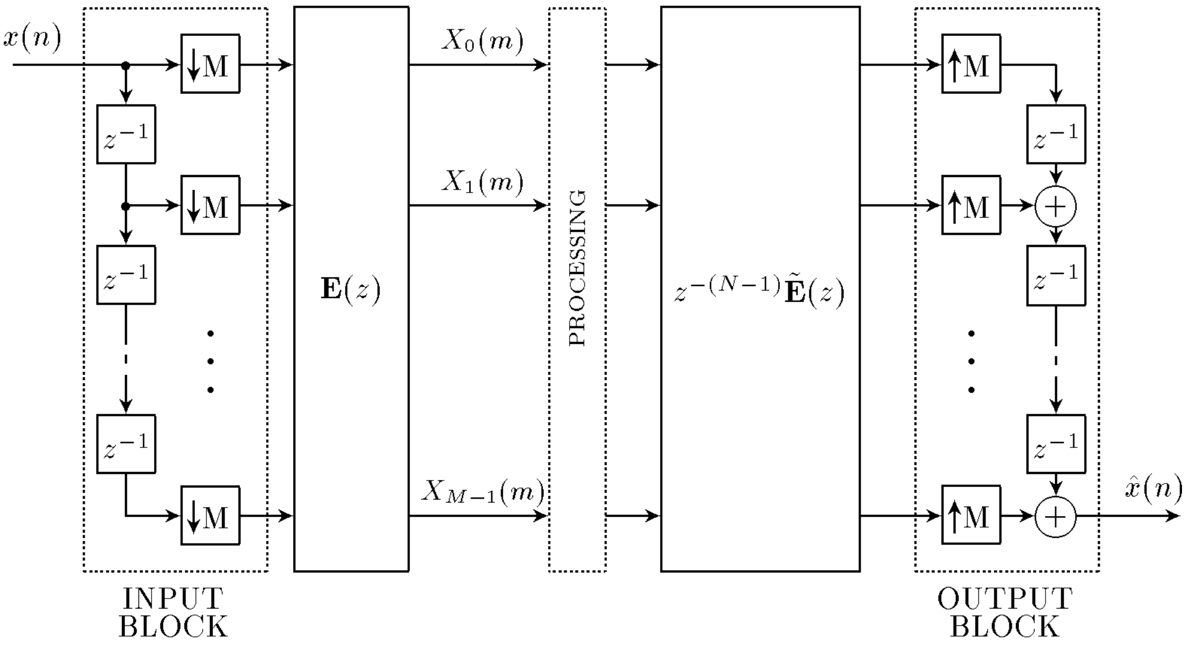

Figure 3: Structure equivalent of Figs. 1 and 2, with the analysis and synthesis filters being implemented in terms of their polyphase component matrices.

and synthesis filters in (2), the polyphase component matrix of the synthesis filters are given by $z^{(N-1)}\tilde{\mathbf{E}}(z)$, as shown in Fig. 3, where $\tilde{\mathbf{E}}(z) \equiv \mathbf{E}'(z^{-1})$ [14].

With $\mathbf{E}(z)$ given by (7), we have

$$\mathbf{E}(z)\tilde{\mathbf{E}}(z) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} z^{(n-m)}\mathbf{P}'_n\mathbf{P}_m = \mathbf{I} \qquad (8)$$

where the second equality comes from the fact that the orthogonality conditions in (3) are equivalent to

$$\sum_{m=0}^{N-1-l} \mathbf{P}'_m\mathbf{P}_{m+l} = \delta(l)\ \mathbf{I}\ . \qquad (9)$$

Therefore, (8) shows that the polyphase component matrix $\mathbf{E}(z)$ is lossless [14], and thus a lapped transform has the PR property. Note that since $\mathbf{E}(z)\tilde{\mathbf{E}}(z) = \mathbf{I}$ implies $\tilde{\mathbf{E}}(z)\mathbf{E}(z) = \mathbf{I}$, (9) can also be written in the form

$$\sum_{m=0}^{N-1-l} \mathbf{P}_m\mathbf{P}'_{m+l} = \delta(l)\ \mathbf{I} \qquad (10)$$

which is an equivalent necessary and sufficient PR condition.

It is also clear from (8) that losslessness of $\mathbf{E}(z)$ implies in the orthogonality conditions in (9). Thus, every filter bank with identical (within time reversal) FIR analysis and synthesis filters, having the perfect reconstruction property, is a lapped transform.

B. Structure of Extended Lapped Transforms

The problem of how to design an appropriate matrix $\mathbf{P}$ lies not only in satisfying (3) and (9) or (10), but also in making each $H_k(z)$ a good band-pass filter. The LOT construction defined in [3], based on the DCT basis functions, is a fast-computable solution, but is valid only for $N = 2$. Another alternative with a fast algorithm, for $N = 2$, is the MLT, which is a TDAC filterbank with a particular window.

The ELT was originally defined in [13] as a cosine-modulated filterbank, in a generalization of the MLT definition in [3]. Its corresponding $\mathbf{P}$ matrix is given by

$$p_{nk} \equiv h(n) \sqrt{\frac{2}{M}} \cos\left[\omega_k \left(n + \frac{M+1}{2}\right)\right] \tag{11}$$

which is a cosine modulated filter bank with modulating frequencies

$$\omega_k \equiv \left(k + \frac{1}{2}\right) \frac{\pi}{M} \tag{12}$$

and the window $h(n)$ must have the symmetry $h(L-1-n) = h(n)$ [3], [13]. It is interesting to note that filter banks with cosine modulation at precisely the same frequencies ω_k as in (12) were originally suggested in [17, 18], and were also considered in [19, 20]. In all of these previous works, however, the phases of the modulating cosines were not adequate for perfect reconstruction, and quasi-PR designs were presented. For the special case $L = 2M$, it was noted in [3, 10, 21] that perfect reconstruction could be obtained. In the following we will see that PR can actually be achieved with filters of any length, as long as $h(n)$ satisfies a PR condition.

Without loss of generality, we will assume that $N = 2K$, i.e., the analysis and synthesis filters have lengths equal to $L = 2KM$, where K will be referred to as the *overlapping factor*. Thus, the original MLT is actually an ELT with $K = 1$.

Let us define the $M \times M$ matrices $\mathbf{\Phi}_l$ and $\mathbf{H}_l$ by

$$[\mathbf{\Phi}_l]_{nk} \equiv \sqrt{\frac{2}{M}} \cos\left[\omega_k \left(n + lM + \frac{M+1}{2}\right)\right] \tag{13}$$

and

$$\mathbf{H}_l \equiv \mathrm{diag}\left\{h(lM),\ h(lM+1),\ \ldots,\ h(lM+M-1)\right\} . \tag{14}$$

In (13) both the row index n and the column index k vary from 0 to $M-1$.

Let us also define the $2KM \times M$ matrix $\mathbf{\Phi}$ by $\mathbf{\Phi}' \equiv [\mathbf{\Phi}_0'\mathbf{\Phi}_1' \cdots \mathbf{\Phi}_{2K-1}']'$, and the window matrix $\mathbf{H} \equiv \mathrm{diag}\{\mathbf{H}_0, \mathbf{H}_1, \ldots, \mathbf{H}_{2K-1}\}$. Thus, the modulating functions are the columns of $\mathbf{\Phi}$, and (11) can be written in matrix form as

$$\mathbf{P}_i = \mathbf{H}_i \mathbf{\Phi}_i \tag{15}$$

which is equivalent to $\mathbf{P} = \mathbf{H\Phi}$.

A fundamental property of the submatrices of the modulating matrix is

$$\begin{aligned} \mathbf{\Phi}_i \mathbf{\Phi}_{i+2m}' &= (-1)^m [\mathbf{I} + (-1)^i \mathbf{J}] \\ \mathbf{\Phi}_i \mathbf{\Phi}_{i+2m+1}' &= \mathbf{0} \end{aligned} \tag{16}$$

The proof of (16) is somewhat long but it is relatively simple, and thus it will be omitted. The basic point behind the proof is that each sinusoid in a row of $\mathbf{\Phi}_i$ is orthogonal to all but two of the rows in $\mathbf{\Phi}_{i+2m}$, and orthogonal to all the rows of $\mathbf{\Phi}_{i+2m+1}$.

Using (16), we can write, after some algebraic manipulations,

$$\sum_{m=0}^{N-1-l} \mathbf{P}_m \mathbf{P}'_{m+l} = \cos\left(\frac{l\pi}{2}\right) \sum_{i=0}^{2K-l-1} \mathbf{H}_i [\mathbf{I} + (-1)^i \mathbf{J}] \mathbf{H}_{i+l} \ . \tag{17}$$

Since the ELT matrix $\mathbf{P}$ in (11), (15) will be a lapped transform only if (10) is satisfied, applying (10) to (17) leads to

$$\sum_{i=0}^{2K-2m-1} \mathbf{H}_i \, \mathbf{H}_{i+2m} = \delta(m) \, \mathbf{I} \tag{18}$$

for $m = 0, 1, \ldots, K-1$. The scalar form of the above equation is

$$\sum_{i=0}^{2K-2m-1} h(n+iM) \, h(n+iM+2mM) = \delta(m) \tag{19}$$

for $n = 0, 1, \ldots, M/2 - 1$, which was previously obtained in [13]. Thus, the set of nonlinear equations in (19) represents the necessary and sufficient conditions on the window $h(n)$ for the generation of an extended lapped transform.

It is important to note that there do exist solutions to (19) in which some of the end samples of $h(n)$ are zero. Therefore, the analysis and synthesis filters may effectively have any length, and not just even multiples of M.

C. Properties

A basic property of the ELT is that the frequency responses of the filter banks are approximately shifted versions of the window frequency response, that is

$$H_k(e^{j\omega}) = \frac{1}{\sqrt{2M}} \left[e^{-j\theta_k} H(e^{j(\omega-\omega_k)}) + e^{j\theta_k} H(e^{j(\omega+\omega_k)}) \right] \tag{20}$$

where $\theta_k \equiv \omega_k(M+1)/2$. Thus, except for $k = 0$ and $k = M-1$, the following approximation is valid:

$$|H_k(e^{j\omega})| \simeq \frac{1}{\sqrt{2M}} |H(e^{j(\omega-\omega_k)})| \, . \tag{21}$$

Therefore, the basic design criteria for $h(n)$ is that $|H(e^{j\omega})|$ should be a good low-pass filter, with a bandwidth of $\pi/2M$, in order to minimize aliasing among the subband signals.

Another interesting property of the ELT basis functions, which is easy to prove, is that for M even

$$f_{M-1-m}(2KM-1-n) = (-1)^{m+K-M/2+1} \, (-1)^n \, f_m(n) \ . \tag{22}$$

Thus, except for a plus or minus sign, half of the filter responses can be obtained from the other half by time reversal and multiplication by $(-1)^n$. Note that, for $M = 2$, this is precisely the property that defines the conjugate quadrature filters (CQF) [22]. Thus, the ELT can also be viewed as a generalization of the CQF concept for an arbitrary number of subbands.

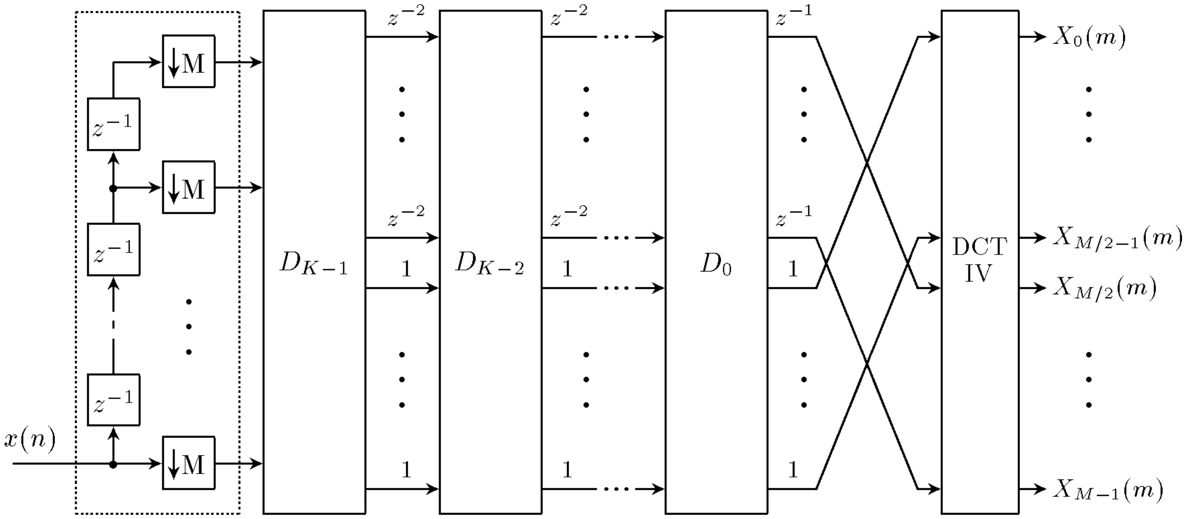

Figure 4: Fast implementation of the direct ELT (analysis filter bank).

3 Fast Algorithms

The ELT structure of (11) has two main advantages over general filter banks. The first, which was pointed out in the previous section, is that a good design for $h(n)$ automatically guarantees that all filters will have a good bandpass response. The second is that the regular structure of the matrix $\mathbf{\Phi}$ leads to a fast algorithm.

In fact, fast algorithms for the ELT have been presented in [3, 23, 24] for $K = 1$ (which corresponds to the MLT) and in [25] for $K = 2$. In this section we will present a fast ELT implementation for any K, based on orthogonal butterflies and a type-IV discrete cosine transform. Structures based on lattice sections and fast transforms were independently derived, and recently introduced in [26, 27].

A. Structure of the Fast Algorithm

The basic idea behind a fast ELT algorithm is to implement the polyphase component matrix $\mathbf{E}(z)$ in Fig. 3 as a cascade of two kinds of matrices: zero-delay orthogonal factors and pure delays. This idea was originally used in the canonical implementations for lossless filter banks in [28]. However, due to the special cosine modulation structure of the ELT, a particular form of the more general structures in [28] leads to a more efficient implementation for the ELT, as we will see in the following.

The structures for the fast direct ELT and inverse ELT are shown in Figs. 4 and 5, respectively. The basic factors of the fast ELT structures are the symmetrical butterfly matrices $\mathbf{D}_k$, which are defined by

$$\mathbf{D}_k \equiv \begin{pmatrix} -\mathbf{C}_k & \mathbf{S}_k\mathbf{J} \\ \mathbf{J}\mathbf{S}_k & \mathbf{J}\mathbf{C}_k\mathbf{J} \end{pmatrix} \tag{23}$$

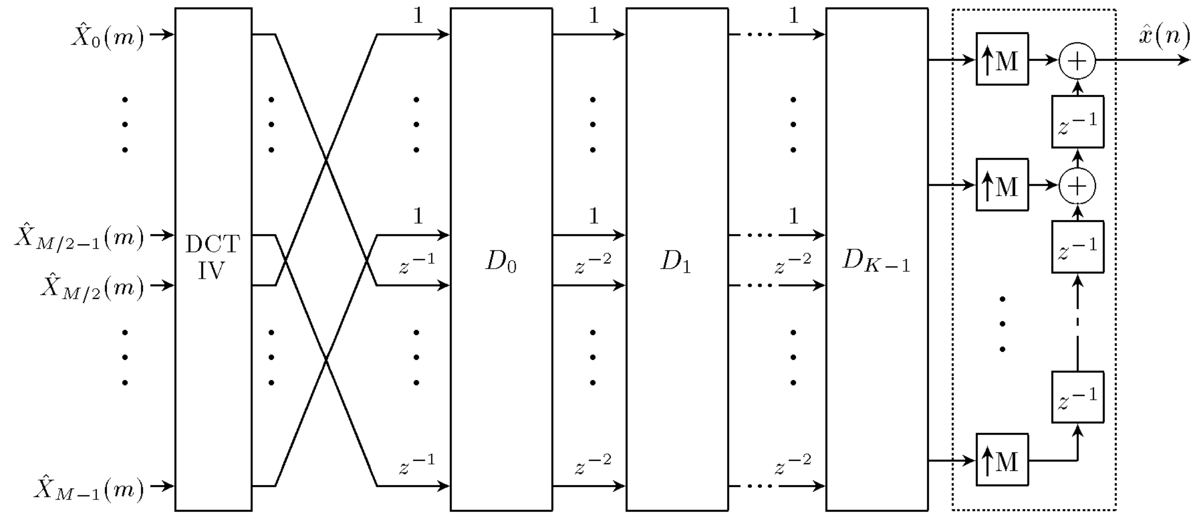

Figure 5: Fast implementation of the inverse ELT (synthesis filter bank).

where

$$\begin{aligned}\mathbf{C}_k &\equiv \operatorname{diag}\{\cos\theta_{0k}, \cos\theta_{1k}, \ldots, \cos\theta_{M/2-1,k}\}\\ \mathbf{S}_k &\equiv \operatorname{diag}\{\sin\theta_{0k}, \sin\theta_{1k}, \ldots, \sin\theta_{M/2-1,k}\}\,. \end{aligned} \tag{24}$$

After the butterfly matrices and the delays, the last factor of the ELT structure is a type-IV DCT operator [29, 30], which has a fast algorithm.

As we can see in Figs. 4 and 5, there are K butterfly matrices in the fast ELT. Thus, there are $KM/2$ butterfly angles θ_{rk}, with $0 \leq r \leq M/2-1$ and $0 \leq k \leq K-1$. Note that $KM/2$ is precisely the number of degrees of freedom in choosing the window $h(n)$, because there are KM distinct values in $h(n)$ and $KM/2$ distinct PR restrictions in (19).

In Appendix A we present the proof of the validity of the fast ELT structures in Figs. 4 and 5, in the sense that for any PR window $h(n)$ there is a set of angles θ_{rk} and vice-versa.

B. Computational Complexity

It is clear that the direct and inverse ELT have the same complexity. Assuming that the DCT-IV is computed using the DCT–to–DCT-IV mapping in [30] (which requires a total of $M+(M/2)\log_2 M$ multiplications and $M+(3M/2)\log_2 M$ additions), and that each butterfly is computed with three multiplications and three additions [30], the total complexity of the fast ELT would be $(M/2)(3K+\log_2 M+2)$ multiplications and $(M/2)(3K+3\log_2 M+2)$ additions. However, the complexity can be reduced by using the DCT-IV algorithm presented in [31], which saves M additions over the algorithm in [30].

An important property of the butterfly matrices in Figs. 4 and 5 that can be used to reduce the computational complexity is that the cascade of the matrices $\mathbf{D}_k$ leads to a cascade connection of butterflies and delays. Therefore, we could scale all the coefficients in butterflies $\mathbf{D}_1$ to $\mathbf{D}_{K-1}$ such that all the diagonal entries would be equal to 1 or -1, and the necessary inverse scaling would be applied to $\mathbf{D}_0$. Thus, instead of using the butterfly

M	K=1		K=2		K=3		K=4		K=5	
	MUL	ADD	MUL	ADD	MUL	ADD	MUL	ADD	MUL	ADD
2	5	5	7	7	9	9	11	11	13	13
4	14	18	18	22	22	26	26	30	30	34
8	32	48	40	56	48	64	56	72	64	80
16	72	120	88	136	104	152	120	168	136	184
32	160	288	192	320	224	352	256	384	288	416
64	352	672	416	736	480	800	544	864	608	928
128	768	1536	896	1664	1024	1792	1152	1920	1280	2048
256	1664	3456	1920	3712	2176	3968	2432	4224	2688	4480
512	3584	7680	4096	8192	4608	8704	5120	9216	5632	9728
1024	7680	16896	8704	17920	9728	18944	10752	19968	11776	20992

Table 1: Computational complexity of the fast ELT algorithm.

matrices $\mathbf{D}_k$ we could use the matrices $\overline{\mathbf{D}}_k$, defined by

$$\begin{aligned}\overline{\mathbf{D}}_k &\equiv \mathrm{diag}\{\mathbf{C}_k^{-1},\, \mathbf{J}\mathbf{C}_k^{-1}\mathbf{J}\}\,\mathbf{D}_k,\ k=1,\ldots,K-1\\ \overline{\mathbf{D}}_0 &\equiv \mathrm{diag}\{\prod_{k=1}^{K-1}\mathbf{C}_k,\, \mathbf{J}\prod_{k=1}^{K-1}\mathbf{C}_k^{-1}\mathbf{J}\}\,\mathbf{D}_0\,.\end{aligned} \tag{25}$$

In this way, $\overline{\mathbf{D}}_0$ would require three multiplications and three additions per butterfly, but in $\overline{\mathbf{D}}_k$ only two multiplications and two additions per butterfly would be necessary. The final computational complexity of the fast ELT would be

$$\begin{aligned}&\frac{M}{2}\,(2K+\log_2 M+3) \qquad \text{multiplications and}\\ &\frac{M}{2}\,(2K+3\log_2 M+1) \qquad \text{additions.}\end{aligned} \tag{26}$$

For the two-band case, i.e. $M=2$, the scaling factors in (25) can be absorbed in the single butterfly that implements the DCT-IV, and thus we would save one extra multiplication and one extra addition. In Table 1 we have a summary of the fast ELT complexity.

We see that the computational complexity of the ELT is close to that of block transforms, and thus much lower than those of other filter banks such as the QMF bank [2] or the PR structures based on canonical factorizations [28]. For example, for $M=32$ and $K=2$, the number of operations (multiplications and additions) per input sample is 16, for the ELT. As a comparison, the DCT requires 9.1 operations per sample, and the PR structures of [28] needs 104 operations per sample. The recently introduced lattice structures for the ELT in [26, 27] lead virtually to the same computational complexity as our butterfly structure.

Only parallel structures were compared in the previous discussion. If M is a power of two, binary tree structures should also be included in the comparison. In particular, a tree based on the two-band lattice structures in [32] would have approximately the same complexity of the fast ELT. However, a very good low-pass filter may lose most of its stopband attenuation after going through many levels of decimation/interpolation. In fact, in order to use a tree with many levels, the original filters should be constrained to satisfy a regularity condition from the wavelet transform theory [33], [34]. If we implement a filter bank as a tree of any of the aforementioned filters designed for $M = 2$ (including the ELT), the equivalent filters at the final resolution may contain small fractal components [33] in their frequency responses, due to lack of regularity. This discussion just touches the point that a tree-structured PR filter bank should be approached from the wavelet theory, but this is a topic that goes beyond the scope of this paper.

C. Choice of the Butterfly Angles

With the fast ELT structures in Figs. 4 and 5, we have $KM/2$ angles that can be freely chosen, and each choice leads to a different window $h(n)$. As we have discussed in the previous section, the frequency responses of the filters in the analysis and synthesis banks will be translated versions of $H(e^{j\omega})$. Therefore, we should try to pick up the butterfly angles such that $H(e^{j\omega})$ is a good low-pass filter, with bandwidth close to $\pi/2M$, in order to achieve good subband separation. There are many possible measures of how good the prototype filter is, and a simple but efficient one is the stopband energy

$$\xi = \frac{1}{\pi} \int_{\omega_s}^{\pi} |H(e^{j\omega})|^2 \, d\omega \tag{27}$$

where ω_s defines the beginning of the stopband, and it should be greater than $\pi/2M$. For $\omega_s = \pi/M$, for example, a small value of ξ implies that there will be virtually no aliasing among subbands that are not adjacent to each other. This error measure is commonly used in filter design, including PR filter banks [28].

Given a set of angles θ_{rk}, it is possible to compute the corresponding window $h(n)$ from the results in Appendix A, and then the stopband error can be evaluated from (27). Note that the integral in (27) can be easily written as a positive semidefinite Toeplitz form on the vector formed by the coefficients of $h(n)$ [35] (we recall that Toeplitz forms can be efficiently evaluated through the FFT), so that no numerical integration procedure is required. Then, a standard non-linear optimization procedure can be used to iteratively refine the angles until a minimum in ξ is achieved.

We have written our own optimization routine, in the C language, based on the Broyden-Fletcher-Goldfarb-Shanno method of iteratively updating an estimate of the inverse Hessian matrix of the objective function [36]. The eigenvalue spread of the Hessian matrix can be worse than five orders of magnitude, and this means that double-precision calculations are required and, more importantly, that convergence will be quite slow. For $K = 4$ and $M > 16$, for example, it took more than 200 iterations for convergence. This ill-conditioning of the optimization problem is intrinsic to the choice of the objective function, and cannot be avoided (the design technique for PR filterbanks in [28] also suffers from this problem).

The main problem in the optimization procedure is not in the slow convergence, but in the initial choice for the angles. Because of the highly non-linear nature of the error function, the optimization algorithm can easily be trapped in a local minimum that is much worse than the true global minimum. The method we have chosen to initialize the

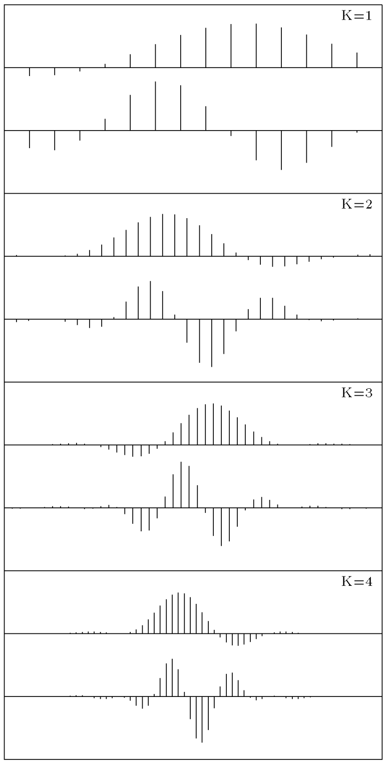

Figure 6: The first two basis functions of the ELT for $M = 8$, for four choices of the overlapping factor K.

angles is based on an auxiliary window $u(n)$, defined by

$$u(n) \equiv \sum_{i=0}^{K-1} a_i \cos\left[f_i\left(n - KM + \frac{1}{2}\right)\right] \quad n = 0, 1, \ldots, 2KM - 1. \tag{28}$$

For any choice of the amplitudes a_i and frequencies f_i in (28), the window $u(n)$ will have the required even symmetry, but it may not satisfy the PR property. Thus, we perform first an optimization of the a_i and f_i parameters so that the PR error, as measured by (19), is minimized. With K sinusoids, as suggested in (28), we can get quite close to a PR window (for $K = 1$ and $K = 2$ we can get exactly a PR window [3, 25]). Then, the relationships in Appendix A are used to obtain an initial set of angles from $u(n)$. With these initial angles, the optimization of ξ is then performed. In practice, we always got good filters $h(n)$ from this procedure, although global optimality is not guaranteed.

In spite of all the difficulties with the optimization, we recall that the number of free angles in the ELT is much smaller than in the canonical PR structures in [28]. For example, for $M = 32$ and $K = 2$ we have to optimize 32 angles, whereas the approach in [28] would require the optimization of 465 angles, a much more difficult task.

As an example, we have in Fig. 6 the first two basis functions of the MLT for $M = 8$, with K varying from one to four. In all cases we have used $\omega_s = 1.2\pi/M$. In Fig. 7 we have their corresponding magnitude responses. As expected, the frequency resolution

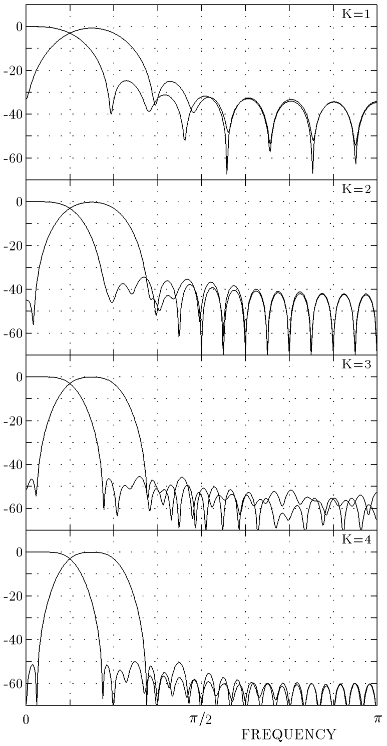

Figure 7: Magnitude frequency responses (in dB) of the filters in Fig. 6.

improves for larger values of the overlapping factor K. In Appendix B we have a table of the butterfly angles for the ELT, with $\omega_s = 1.2\pi/M$, for some vales of M and K.

4 Applications

With its fast algorithm, good frequency resolution, and absence of blocking effects, the ELT can replace traditional block transforms in TC systems, with a little penalty in computational complexity. The ELT can also replace QMF or other non-PR filter banks, and it allows efficient implementation of SBC systems with a large number of bands. Therefore, we believe that the main area of application of the ELT is in signal coding. However, anywhere a block transform or a filter bank can be used, the ELT can also be used, with the potential for a better performance/complexity tradeoff. A good example is adaptive filtering, where block transforms and filter banks have already been employed with good results [37, 38]. In this paper we focus only on the use of the ELT in signal coding, whereas in [31] we will discuss in detail applications of the ELT in signal filtering and spectrum estimation.

Instead of implementing an specific coder with the ELT, let us consider some performance bounds. Whatever adaptive quantization scheme is employed (vector or scalar, AQF or AQB [1]), the coder performance cannot be better than the maximum coding gain of the transform, which is given by the ratio of the arithmetic to the geometric means of

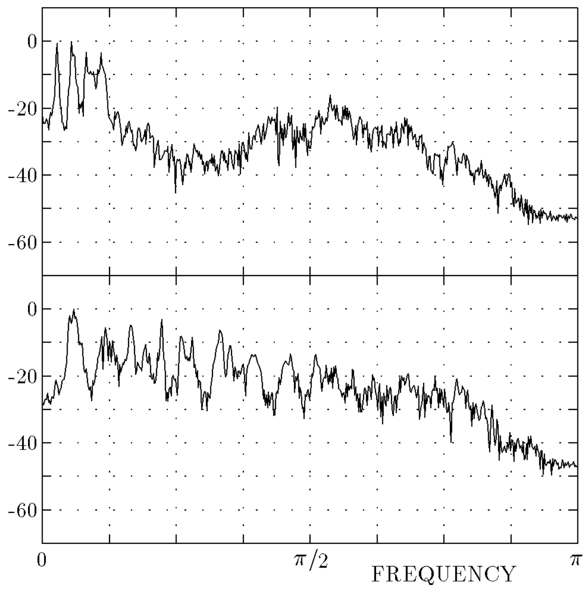

Figure 8: Power spectra, in dB, for two speech signal sampled at 8 kHz. Top: male speaker; bottom: female speaker.

the transform coefficient variances [1, 3, 6]. We have used for our analyses three signal spectra: a first order Markov process with an intersample correlation coefficient $\rho = 0.95$ (which is a good model for images [1]), and the two speech spectra shown in Fig. 8. The speech signals were sampled at 8 kHz, and the spectra were estimated from the average of the power spectra of three 1024-sample segments. For the male speech, most of the utterance contained a vowel and a fricative, whereas the female speech contained a mixture of voiced and unvoiced phonemes.

In Tables 2 to 4 we have the coding gains of the ELT, for several choices for the number of subbands M. As a standard for the comparison, we have included in the tables the coding gains that would be obtained with ideal filter banks (the column labeled $M = \infty$ corresponds to the maximum coding gain that can be achieved for the signal source), and those of the DCT. It is interesting to note that the coding gain generally jumps when we go from the DCT to the ELT with $K = 1$, and then it increases slowly with K. When $K = 4$ we get so close to the maximum coding gain of the ideal filter bank, for any given number of subbands, that for coding applications it would not be necessary to work with overlapping factors greater than four.

Comparing the performances of the DCT for $M = 64$ and the ELT with $M = 16$ and $K = 2$, we see that the coding gains are similar, and in both cases the analysis and synthesis filters would have lengths equal to 64. The processing delay would also be the same in both cases, but the total computational complexity of the analysis and synthesis filter banks would be 1,412 operations for the DCT and 448 operations for the ELT. Furthermore, quantizing 16 ELT coefficients would take fewer computations than quantizing 64 DCT coefficients, whatever adaptive quantization scheme is employed. This gain in computational complexity comes with another gain: the absence of blocking effects.

Bank	Number of Bands, M							
	2	4	8	16	32	64	128	∞
DCT	5.05	7.57	8.83	9.46	9.77	9.94	10.02	
ELT, K=1	5.50	8.11	9.32	9.83	10.02	10.08	10.10	
ELT, K=2	5.76	8.39	9.48	9.90	10.04	10.09	10.10	
ELT, K=3	5.86	8.48	9.55	9.93	10.05	10.09		
ELT, K=4	5.87	8.50	9.56	9.94	10.05	10.10		
Ideal	5.94	8.56	9.59	9.94	10.05	10.10	10.10	10.11

Table 2: Coding gain (dB) for an AR(1) signal with $\rho = 0.95$.

Bank	Number of Bands, M							
	2	4	8	16	32	64	128	∞
DCT	3.46	5.31	7.37	9.18	10.41	11.29	11.86	
ELT, K=1	3.75	6.66	9.14	10.97	11.83	12.32	12.67	
ELT, K=2	3.80	7.68	9.87	11.16	11.87	12.40	12.78	
ELT, K=3	3.84	8.09	10.36	11.46	12.07	12.53		
ELT, K=4	3.86	8.16	10.45	11.53	12.10	12.56		
Ideal	3.88	8.42	10.81	11.87	12.32	12.64	12.93	13.10

Table 3: Coding gain (dB) for a male speech signal.

The coding gains of the ELT are in fact what one could expect, given their filtering performance. Other filter banks such as the pseudo-QMF bank [19] would lead to similar coding gains. The advantages of the ELT are in its PR property and fast implementation. Finally, one important property of the ELT that is not shown in Table 2 is that the coding gain of the ELT is independent of the sign of the correlation coefficient for an AR(1) source. This is a major advantage over the DCT, whose performance degrades significantly for sources having a negative correlation coefficient [39].

Bank	Number of Bands, M							
	2	4	8	16	32	64	128	∞
DCT	1.77	3.07	4.20	5.24	6.06	6.58	7.18	
ELT, K=1	2.30	4.11	5.45	6.16	6.79	7.50	8.13	
ELT, K=2	2.86	4.42	5.75	6.22	7.01	7.76	8.29	
ELT, K=3	3.16	4.57	5.92	6.33	7.13	7.87		
ELT, K=4	3.18	4.59	5.95	6.35	7.16	7.90		
Ideal	3.27	4.60	5.99	6.42	7.26	8.09	8.46	9.07

Table 4: Coding gain (dB) for a female speech signal.

5 Conclusion

We have presented a more general definition of lapped transforms (LTs), and obtained the perfect reconstruction (PR) conditions for arbitrarily long basis functions. The LTs are actually a subclass of all PR FIR filter banks, in which the synthesis filters are identical, within time reversal, to the analysis filters, i.e., the simple orthogonality conditions in (3) are satisfied. We have also shown that the cosine modulated PR filter banks presented in [13] do satisfy those orthogonality conditions, and thus they do belong to the class of lapped transforms. Because these filter banks are an extension of the modulated lapped transform (MLT), for filter lengths that can be greater than twice the number of subbands, we refer to this new family as the extended lapped transform (ELT).

We have presented fast algorithms for the ELT with filter lengths equal to $L = 2KM$, where M is the number of subbands and K is the overlapping factor. The values of the rotation angles in the fast ELT structure are obtained by means of a nonlinear optimization procedure, which has its difficulties, but it needs to be performed only once.

Finally, we have demonstrated the efficiency of the ELT in signal coding applications, where we have shown that the ELT can replace the DCT in transform coding systems, with higher coding gains and absence of blocking effects. In subband coding (SBC), the ELT can replace other filter banks, since the ELT is the PR filter bank with the lowest computational complexity reported to date. In fact, SBC systems with a large number of subbands can be easily implemented in inexpensive DSP chips for many applications.

References

[1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[2] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[3] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 969–978, June 1990.

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

[5] H. S. Malvar and D. H. Staelin, "Reduction of blocking effects in image coding with a lapped orthogonal transform," in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (New York), pp. 781–784, Apr. 1988.

[6] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[7] P. Cassereau, "A New Class of Optimal Unitary Transforms for Image Processing," Master's thesis, Mass. Inst. Tech., Cambridge, MA, May 1985.

[8] H. S. Malvar, “Optimal pre- and post-filters in noisy sampled-data systems,” Tech. Rep. 519, Research Lab. Electronics, Mass. Inst. Tech., Cambridge, MA, Sept. 1986.

[9] H. S. Malvar, “The LOT: a link between block transform coding and multirate filter banks,” in *IEEE Intl. Symp. Circuits Syst.*, (Espoo, Finland), pp. 835–838, June 1988.

[10] J. P. Princen, A. W. Johnson, and A. B. Bradley, “Sub-band/transform coding using filter bank designs based on time domain aliasing cancellation,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Dallas), pp. 2161–2164, Apr. 1987.

[11] J. P. Princen and A. B. Bradley, “Analysis/synthesis filter bank design based on time domain aliasing cancellation,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1153–1161, Oct. 1986.

[12] J. Kovacevic, D. J. Le Gall, and M. Vetterli, “Image coding with windowed modulated filter banks,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Glasgow, Scotland), pp. 1949–1952, May 1989.

[13] H. S. Malvar, “Modulated QMF filter banks with perfect reconstruction,” *Electron. Lett.*, vol. 26, no. 13, pp. 906–907, June 1990.

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

[15] P. P. Vaidyanathan and Z. Doğanata, “The role of lossless systems in modern digital signal processing,” *IEEE Trans. Education*, vol. 32, pp. 181–197, Aug. 1989.

[16] K. Nayebi, T. P. Barnwell III, and M. J. T. Smith, “The time domain analysis and design of exactly reconstructing FIR analysis/synthesis filter banks,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Albuquerque, NM), pp. 1735–1738, Apr. 1990.

[17] J. H. Rothweiler, “Polyphase quadrature filters – a new subband coding technique,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Boston, MA), pp. 1280–1283, Mar. 1983.

[18] P. Chu, “Quadrature mirror filter design for an arbitrary number of equal bandwidth channels,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 203–218, Feb. 1985.

[19] H. Nussbaumer and M. Vetterli, “Computationally efficient QMF filter banks,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (San Diego, CA), pp. 11.3.1–11.3.4, Mar. 1984.

[20] J. Masson and Z. Picel, “Flexible design of computationally efficient nearly perfect QMF filter banks,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Tampa, FL), pp. 541–544, Mar. 1985.

[21] M. Vetterli and D. J. Le Gall, “Perfect reconstruction FIR filter banks: lapped transforms, pseudo QMF’s and paraunitary matrices,” in *IEEE Intl. Symp. Circuits Syst.*, (Espoo, Finland), pp. 2249–2253, June 1988.

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

[23] S. Cramer and R. Gluth, “Computationally efficient real-valued filter banks based on a modified O^2DFT,” in *Proc. European Signal Processing Conf.*, (Barcelona, Spain), pp. 585–588, Sept. 1990.

[24] P. Duhamel, Y. Mahieux, and J. P. Petit, “A fast algorithm for the implementation of filter banks based on time domain aliasing cancelation,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), pp. 2209–2212, May 1991.

[25] H. S. Malvar, “Fast algorithm for modulated lapped transform,” *Electron. Lett.*, vol. 27, no. 9, pp. 775–776, Apr. 1991.

[26] R. D. Koilpillai and P. P. Vaidyanathan, “New results on cosine-modulated filter banks satisfying perfect reconstruction,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), pp. 1793–1796, May 1991.

[27] T. A. Ramstad and J. P. Tanem, “Cosine-modulated analysis-synthesis filterbank with critical sampling and perfect reconstruction,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), pp. 1789–1792, May 1991.

[28] P. P. Vaidyanathan, T. Q. Nguyen, T. Saramäki, and Z. Doğanata, “Improved technique for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1042–1056, July 1989.

[29] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic Press, 1990.

[30] Z. Wang, “On computing the discrete Fourier and cosine transforms,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1341–1344, Oct. 1985.

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

[32] T. Q. Nguyen and P. P. Vaidyanathan, “Two-channel perfect reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 676–690, May 1989.

[33] M. Vetterli, “Wavelets and filter banks: relationships and new results,” in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (Albuquerque, NM), pp. 1723–1726, Apr. 1990.

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

[35] P. P. Vaidyanathan and T. Q. Nguyen, “Eigenfilters: a new approach to least-squares FIR filter design and applications including nyquist filters,” *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 11–23, Jan. 1987.

[36] D. G. Luenberger, *Linear and Non-Linear Programming*. Reading, MA: Addison-Wesley, 2nd ed., 1984. Chap. 9.

[37] D. F. Marshall, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 474–484, Apr. 1989.

[38] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands," in *IEEE Intl. Conf. Acoust., Speech, Signal Processing*, (New York), pp. 1572–1575, Apr. 1988.

[39] A. N. Akansu and R. A. Haddad, "On asymmetrical performance of discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 154–156, Jan. 1990.

Appendix A
Proof of the Fast ELT Implementation

In this Appendix we show that any ELT can be implemented with the fast computational structures of Figs. 4 and 5, that is, for every PR window $h(n)$ there exists a set of angles θ_{rk}, and vice-versa.

We will approach the proof by induction. Assuming that the structure generates an ELT for some overlapping factor K, we will show that it will also generate an ELT for $K+1$. Let us call the polyphase component matrix $\mathbf{E}(z)$ for some K as $\mathbf{E}_K(z)$. Then, it is clear from the regularity of Fig. 5 that

$$z^{-[2(K+1)-1]}\,\mathbf{J}\tilde{\mathbf{E}}_{K+1}(z) = \mathbf{D}_K \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-2}\mathbf{I} \end{pmatrix} z^{-(2K-1)}\,\mathbf{J}\tilde{\mathbf{E}}_K(z) \qquad \text{(A.1)}$$

Using (23), we can rewrite (A.1) as

$$\mathbf{J}\tilde{\mathbf{E}}_{K+1}(z) = \begin{pmatrix} -\mathbf{C}_k & \mathbf{S}_k\mathbf{J} \\ \mathbf{J}\mathbf{S}_k & \mathbf{J}\mathbf{C}_k\mathbf{J} \end{pmatrix} \begin{pmatrix} z^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{J}\tilde{\mathbf{E}}_K(z)\,. \qquad \text{(A.2)}$$

In order to put (A.2) in terms of the window coefficients, we need a few intermediate results. First, it is easy to show, from (13) that

$$\begin{aligned} \mathbf{\Phi}_{2i} &= (-1)^i\,\mathbf{\Phi}_0 \\ \mathbf{\Phi}_{2i+1} &= (-1)^i\,\mathbf{\Phi}_1 \end{aligned} \qquad \text{(A.3)}$$

and that

$$\mathbf{\Phi}_0 = \begin{pmatrix} \mathbf{I} \\ -\mathbf{J} \end{pmatrix} \mathbf{\Lambda}_0 \quad \mathbf{\Phi}_1 = \begin{pmatrix} \mathbf{I} \\ \mathbf{J} \end{pmatrix} \mathbf{\Lambda}_1 \qquad \text{(A.4)}$$

where $\mathbf{\Lambda}_0$ and $\mathbf{\Lambda}_1$ can be easily computed from (15), and are $M/2 \times M$ matrices. Second, we can write the window diagonal submatrices, without loss of generality, as

$$\mathbf{H}_i = \begin{pmatrix} \mathbf{U}_i & \mathbf{0} \\ \mathbf{0} & (-1)^i\,\mathbf{J}\mathbf{V}_i\mathbf{J} \end{pmatrix} \qquad \text{(A.5)}$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ are also diagonal matrices.

Substituting (A.3–A.5) and (15) into (7), we can relate the polyphase component matrix directly to the window factors, in the form

$$\mathbf{J}\tilde{\mathbf{E}}_K(z) = \sum_{r=0}^{K-1} z^{2(K-1-r)}\,(-1)^r \left[\begin{pmatrix} \mathbf{U}_{2r} \\ -\mathbf{J}\mathbf{V}_{2r} \end{pmatrix} \boldsymbol{\Lambda}_0 + z^{-1} \begin{pmatrix} \mathbf{U}_{2r+1} \\ -\mathbf{J}\mathbf{V}_{2r+1} \end{pmatrix} \boldsymbol{\Lambda}_1 \right] . \tag{A.6}$$

When we use (A.6) into (A.2), it is easy to see that $\tilde{\mathbf{E}}_{K+1}(z)$ can also be put in the form of (A.6) (and thus, proving by induction that the recursion in (A.1) generates an ELT), with the window components of $\tilde{\mathbf{E}}_{K+1}(z)$ being given by

$$\begin{aligned} \mathbf{U}_i^{K+1} &= -\mathbf{C}_K\mathbf{U}_i^K + \mathbf{S}_K\mathbf{V}_{i-2}^K \\ \mathbf{V}_i^{K+1} &= -\mathbf{S}_K\mathbf{U}_i^K - \mathbf{C}_K\mathbf{V}_{i-2}^K \end{aligned} \tag{A.7}$$

where the superscripts in the $\mathbf{U}_i$ and $\mathbf{V}_i$ matrices denote the overlapping factor to which they correspond. In order to finish the proof, we need to show that the structure in Figs. 4 and 5 does indeed generate an ELT for some K. Since this has been done in [3, 25] for $K = 1$ and $K = 2$, the proof is complete.

Using the initial conditions

$$\mathbf{U}_0^0 \equiv -\mathbf{I}, \quad \mathbf{V}_{-2}^0 \equiv \mathbf{0} \tag{A.8}$$

we can use (A.7) recursively to generate the window $h(n)$ from the set of butterfly angles (remember that the angles define the matrices $\mathbf{C}_k$ and $\mathbf{S}_k$). The recursions in (A.7) can be easily inverted, in order to generate equations that allow us the computation of the butterfly angles from a PR window $h(n)$. For details, see [31].

In the following, we have the listing of a simple MATLAB program that will generate the window $h(n)$ from a set of butterfly angles.

```
function h = wingen(ang)
%
%     WINGEN.M -  Generation of ELT window, given an array with
%                 the butterfly angles
%
%     Syntax: h = wingen(ang)
%
%     Argument:
%
%     ang - matrix containing the butterfly angles.  It should have
%           m/2 rows and k columns.  The angles must be expressed as
%           fractions of pi.  The first column corresponds to D_0 in
%           the paper, the second to D_1, etc.  Recall that m is the
%           number of subbands, and k is the overlapping factor.
%
%     Output:
%
%     h   - vector of length 2*k*m, with the h(n) coefficients
```

```
%-------------------------------------------
%  Define m, k, and reserve space for window
%
ss = size(ang);
m2 = ss(1);
k  = ss(2);
m  = 2 * m2;
h  = zeros(2*k*m, 1);
%-------------------------------------------
%  Initialize h for k=1
%
c0 = cos(pi*ang(:,1));
s0 = sin(pi*ang(:,1));
h(1:m2)   = -c0;
h(m2+1:m) = -s0;
%----------------------------------------
%  Main recursion to generate window
%
if k > 1
for kk = 2:k
c = cos(pi*ang(:,kk));
s = sin(pi*ang(:,kk));
%---------------------------------------------------------------
% Note that the recursion is performed backwards, i = kk-1:1:0,
% because this allows for in-place generation of the window h.
% Note that only half of the window needs to be generated at each
% step, because of the even symmetry of h(n).  For that to work
% when i = kk-1, we use the relationship
%
%              kk-1      kk   kk-1
%             U    = (-1)    V
%              kk-1           kk-2
%
for i = kk-1:-1:0
u = zeros(m2,1);
v = u;
if i == (kk-1)
sg = (-1)^kk;
mi = m * (kk-2);
u = u - sg * c .* h(mi+m2+1:mi+m);
v = v - sg * s .* h(mi+m2+1:mi+m);
end;
if i < (kk-1)
mi = m * i;
u = u - c .* h(mi+1:mi+m2);
v = v - s .* h(mi+1:mi+m2);
```

```
end;
i2 = i - 2;
if (i2 >= 0)
mi = m * i2 + m2;
u = u + s .* h(mi+1:mi+m2);
v = v - c .* h(mi+1:mi+m2);
end;
mi = m * i;
h(mi+1:mi+m2) = u;
h(mi+m2+1:mi+m) = v;
end; % i
end; % kk
end; % if k
%-------------------------------------------
%  Make V_i = ((-1)^i)*J*V_i*J
%
sg = 1;
for i = 1:k,
j = m * (i-1);
h(j+m2+1:j+m) = sg * h(j+m:-1:j+m2+1);
sg = -sg;
end;
%---------------------------------------------------------------
%  Force symmetry, since only half of the window was generated
%
h(k*m+1:2*k*m) = h(k*m:-1:1);
%----------------
%  Fim. All done.
```

Appendix B
Examples of Butterfly Angles

In this Appendix we present a table of butterfly angles for the structures in Figs. 4 and 5, for some values of M and K. The angles were obtained with the optimization procedure described in Section 3, with the stopband start frequency $\omega_s = 1.2\pi/M$, in all cases. The angles are given in Table 5 as fractions of π, in the correct format for input to the `wingen.m` program described in Appendix A. Tables for other values of M, K, and ω_s can be found in [31].

M	K=1	K=2		K=3			K=4			
2	0.3187	0.5259	0.6546	0.4044	0.4501	0.4209	0.4951	0.5923	0.5568	0.5845
4	0.4144	0.5485	0.6138	0.4382	0.4328	0.4300	0.5214	0.5933	0.5519	0.5421
	0.3119	0.5117	0.7015	0.3845	0.4784	0.4070	0.4811	0.5805	0.5421	0.6304
8	0.4352	0.5619	0.5948	0.4463	0.4210	0.4412	0.5273	0.5837	0.5589	0.5336
	0.3935	0.5368	0.6340	0.4352	0.4481	0.4170	0.5164	0.6019	0.5424	0.5503
	0.3417	0.5187	0.6780	0.4173	0.4705	0.3957	0.4980	0.5972	0.5361	0.5932
	0.2817	0.5056	0.7256	0.3497	0.4884	0.4216	0.4674	0.5651	0.5443	0.6656
16	0.4443	0.5693	0.5858	0.4496	0.4143	0.4470	0.5382	0.5888	0.5529	0.5168
	0.4260	0.5549	0.6041	0.4444	0.4291	0.4354	0.5346	0.6054	0.5420	0.5170
	0.4052	0.5424	0.6237	0.4393	0.4425	0.4228	0.5291	0.6194	0.5340	0.5208
	0.3817	0.5317	0.6446	0.4337	0.4548	0.4096	0.5223	0.6288	0.5282	0.5301
	0.3558	0.5226	0.6666	0.4260	0.4659	0.3975	0.5142	0.6301	0.5243	0.5483
	0.3275	0.5150	0.6897	0.4128	0.4760	0.3903	0.5042	0.6183	0.5228	0.5803
	0.2973	0.5085	0.7134	0.3839	0.4849	0.3982	0.4896	0.5872	0.5265	0.6320
	0.2659	0.5028	0.7378	0.3116	0.4925	0.4491	0.4489	0.5368	0.5565	0.7039

Table 5: Butterfly angles (fractions of π) for the fast ELT structure.