Biorthogonal and Nonuniform Lapped Transforms for Transform Coding with Reduced Blocking and Ringing Artifacts

Henrique S. Malvar

Microsoft Research
One Microsoft Way
Redmond, WA 98052

Revised: October 20, 1997

(to appear: IEEE Transactions on Signal Processing
April 1998, pp. 1043–1053)

Abstract

New lapped transforms are introduced. The LBT (lapped biorthogonal transform), and HLBT (hierarchical lapped biorthogonal transform) are appropriate for image coding, and the MLBT (modulated lapped biorthogonal transform) and NMLBT (nonuniform modulated lapped biorthogonal transform) are appropriate for audio coding. The HLBT has a significantly lower computational complexity than the LOT (lapped orthogonal transform), essentially no blocking artifacts, and less ringing artifacts than the commonly-used DCT (discrete cosine transform). The LBT and HLBT have transform coding gains that are typically between 0.5 and 1.2 dB higher than that of the DCT. Image coding examples using JPEG and embedded zerotree coders demonstrate the better performance of the LBT and HLBT. The NMLBT has less ringing artifacts and better reproduction of transient sounds than the MLT, as shown in audio coding examples. Fast algorithms for both the HLBT and the NMLBT are presented.

EDICS: 2.4.3 – Filter Bank Design and Implementation

I. Introduction

Transform domain signal processing has many practical applications, such as adaptive filtering, scrambling, and coding [1]–[3]. Transform coding (TC) is used in many video, image, and audio coding standards, such as MPEG video, MPEG audio, and JPEG [4]. In such applications, the signal is represented as a linear combination of the transform basis functions, and the coefficients of such combination are called transform coefficients. Signal compression is achieved by efficient quantization and entropy coding of the coefficients [4].

Signals such as audio and images have spectral characteristics that vary significantly in time and space. Therefore, transform domain representations are usually performed in blocks of samples. With small enough blocks, the signals within each block can be considered as having approximately constant spectra. In TC, the encoder breaks the signal into blocks of M samples (or $M \times M$ for images). For each block, a direct transform operator is used to compute the transform coefficients for that block. The resulting transform coefficients are quantized (usually, via scalar quantizers [2]) and entropy encoded. The quantization step size is chosen such that the output of the entropy encoder fits within the desired bit rate. At the decoder, each block is reconstructed by first decoding the transform coefficients and then applying the inverse transform operator. The inverse transform reconstructs the block as a linear combination of the transform basis functions, weighted by the reconstructed transform coefficients.

Two kinds of reconstruction artifacts are typical in TC, mainly at low bit rates: blocking (or tiling) and ringing. Blocking artifacts arise because the concatenation of the reconstructed blocks generates signal discontinuities across block boundaries. Ringing artifacts arise because the quantization errors on the transform coefficients generate signal reconstruction errors that last for the entire block duration.

Blocking artifacts can be significantly reduced with lapped transforms (LTs) [1]. LT basis functions have two key properties: (i) they are longer than the block size, and (ii) they decay smoothly to near zero at their boundaries. The lapped orthogonal transform (LOT) [5], which is useful for image coding applications, has basis function with linear phase (even or odd symmetry). For blocks with M samples, the LOT bases have length $L = 2M$. A plot of the first two basis LOT functions for $M = 8$ ($L = 16$) is shown in Fig. 1. The LOT can be computed via the discrete cosine transform (DCT) and some additional butterflies [1],[5].

The modulated lapped transform (MLT) [1], which is a particular form of a cosine-modulated filter bank [1],[2], has even less blocking effects than the LOT. This is because the MLT window forces its basis functions to decay asymptotically to zero at their boundaries. The MLT bases are not linear phase, but that is not an issue for applications such as audio coding [6].

Ringing artifacts occur in transient signals, such as edges in images or plosive sounds in audio. One disadvantage of lapped transforms is that their longer basis functions lead to more ringing artifacts than block transforms such as the DCT. In image coding, ringing is perceived as ghosts around edges, and in audio coding ringing leads to pre-echo [7].

In this paper we present new families of lapped transforms that were designed with the purpose of reducing both blocking and ringing artifacts. For a given block size *M*, these new LTs have low-frequency basis functions that are longer than those of the DCT, and high frequency basis function that are effectively shorter than those of the DCT.

To generate the new multiresolution lapped transforms, it is necessary first to develop biorthogonal versions of the LOT and MLT. The lapped biorthogonal transform (LBT) and the hierarchical lapped biorthogonal transform (HLBT) are discussed in Section II. Section III introduces the modulated biorthogonal transform (MLBT) and the nonuniform modulated biorthogonal transform (NMLBT). Image coding examples with the HLBT are presented in Section IV, and audio coding examples with the NMLBT are presented in Section V. Section VI presents conclusions and future directions.

II. The Lapped Biorthogonal Transform

The LOT can significantly reduce blocking artifacts in TC of images [1]. In most cases, however, some residual artifacts are still visible [5], because the basis functions do not decay exactly to zero at their boundaries, as shown in Fig. 1. One way to force the LOT bases to decay to zero is to use biorthogonal transforms. In a lapped biorthogonal transform (LBT), the $2M \times M$ direct and inverse transform matrices $\mathbf{P}_a$ and $\mathbf{P}_s$ do not satisfy orthogonality constraints, but as a pair they still satisfy the orthogonality and lapped orthogonality conditions

$$\mathbf{P}_a^T \mathbf{P}_s = \mathbf{I} \quad \text{and} \quad \mathbf{P}_a^T \mathbf{W} \mathbf{P}_s = \mathbf{I} \tag{1}$$

where $\mathbf{W}$ is the one-block shift operator [1,5], and the superscript T denotes matrix transposition. We can say that the LOT is an LBT in which we add the additional constraint $\mathbf{P}_a = \mathbf{P}_s$.

It is clear that LBTs have more degrees of freedom than LOTs. Aase and Ramstad [8] have shown that these extra degrees of freedom can be used to simultaneously increase coding gain and smoothness of the synthesis basis functions (and thus reducing blocking artifacts). The optimal LBTs in [8] were obtained by constrained optimization of the elements of $\mathbf{P}_a$ and $\mathbf{P}_s$, and so they do not have a fast computational algorithm.

Fast-computable LBTs can be obtained by direct manipulation of the branch transmittances of the fast LOT flowgraph of [5]. Young and Kingsbury [9] suggested rescaling the DC term of the intermediate DCT coefficients by a factor of $\sqrt{2}$, and Chan [10] suggested rescaling all the oddly-symmetrical DCT coefficients, and optimizing such scaling factors for maximum coding gain. We propose an LBT definition based on a special case of Chan's generalized lapped transform.

We define the direct and inverse LBT by the flowgraph shown in Fig. 2. It is obtained from the LOT flowgraph in [5] by multiplying the first oddly-symmetrical DCT coefficient (i.e., the first AC coefficient) by $\sqrt{1/2}$ in the inverse transform (synthesis), and by $\sqrt{2}$ in the direct transform (analysis). The traditional LOT corresponds to $c = 1$ in Fig. 2.

Some basis functions of the LBT are shown in Fig. 3. They resemble quite closely those obtained in [8]. We note that the particular choices for c above generate synthesis basis functions whose asymptotic end values are exactly zero. That leads to less blocking artifacts than the LOT, as discussed in Section IV.

For coding applications, an important measure of performance for a particular transform is the transform coding gain G_{TC}, which is defined as the ratio of the reconstructed error variance for straight PCM quantization to the reconstructed error variance for transform coding [2]:

$$G_{TC} = \frac{\mathrm{E}[e^2]_{PCM}}{\mathrm{E}[e^2]_{TC}} \tag{2}$$

For high-rate coding and biorthogonal transforms, G_{TC} can be computed by the formula [8]

$$G_{TC} = \left\{ \prod_{i=1}^{M} \left[\left(\frac{\sigma_{y_i}^2}{\sigma_x^2} \right) \| f_i \|^2 \right] \right\}^{-(1/M)} \tag{3}$$

where σ_x^2 is the variance of the input signal, $\sigma_{y_i}^2$ is the variance of the i-th transform coefficient, and $\| f_i \|^2$ is the norm of the i-th synthesis basis function (the i-th column of $\mathbf{P}_s$). It is usual to express G_{TC} in decibels (dB).

A commonly used input signal model for image coding is a first-order Gauss-Markov process with intersample autocorrelation coefficient $\rho = 0.95$ [2],[5],[8]. For such signals, the G_{TC} is 8.83 dB for the DCT, 9.22 dB for the fast LOT, 9.52 dB for the fast LBT of Fig. 2, and 9.63 dB for the optimal LBT of [8]. We see that our fast LBT performs quite closely to the optimal (for which there is no fast algorithm).

The G_{TC} formula in Eqn. (3) is usually employed to compare the performance of transforms, even

though in most applications the high bit rate assumption does not apply. The usual thinking is that the relative G_{TC} performances of different transforms should scale proportionally at low bit rates. We will see later in this section that this is not the case, and so a new G_{TC} formula that takes into account the bit rate is introduced in the Appendix.

Nonuniform Transforms

The LBT is a better alternative to the LOT for coding applications. It achieves higher coding gain and less blocking artifacts at the small cost of one additional multiplication in the direct and inverse transforms. However, the LBT leads to as much ringing artifacts as the LOT, because of the long high-frequency functions. An efficient way to reduce the ringing artifacts is to start with an LBT of half the desired block size, and combine two blocks via the hierarchical structure described in [11]. Young and Kingsbury used such a hierarchical construction in [9], with very good results for video coding.

The hierarchical lapped biorthogonal transform (HLBT) is defined, for a given block size M, as a hierarchical transform formed with an LBT of size $M/2$ in the first level, and a length-2 DCT in the second level, as shown in Fig. 4.The basis function of the HLBT are shown in Fig. 5. The first two functions, #0 and #1, have length $1.5M = 12$ (shorter than those of the LBT, which have length $2M = 16$). All other basis functions have length $M = 8$, i.e. half the length of the LBT functions. We note that the DC synthesis function #0 is quite smooth, decaying asymptotically to zero at its boundaries. This is due to the use of LBTs in the first level of the hierarchy in Fig. 4. The analysis DC function #0 does not decay smoothly to zero, but that is not an issue because the blocking artifacts are generated by the synthesis functions. If we had used an LOT for the first level of the hierarchy in Fig. 4, the analysis and synthesis functions would be identical, and both DC basis functions would not decay to zero at the boundaries (furthermore, they would also have discontinuities within the block).

The computational complexity of the HLBT is lower than that of the LOT. The structure of Fig. 4 leads to 16 multiplies and 42 adds per block [12], whereas the LOT uses 22 multiplies and 54 adds per block [1] (compared to 13 multiplies and 29 adds for the DCT).

The transform coding gain G_{TC} of the HLBT, for the first-order Gauss-Markov process with intersample autocorrelation coefficient $\rho = 0.95$ is 9.10 dB, that is, only 0.12 dB below the LOT and 0.27 dB above the DCT. Using the formulas in the Appendix, we have computed G_{TC} at low bit rates for the DCT, LOT, LBT, and HLBT. The results are shown in Fig. 6. We note that the HLBT has a higher coding gain than the LOT at rates below 0.5 bits/sample. Therefore, for low bit rate image coding applications, the

HLBT is better than the LOT in three aspects: reduced blocking and ringing artifacts, lower computational complexity, and higher coding gain (i.e. less quantization noise).

III. The Modulated Biorthogonal Transform

For audio coding applications, frequency selectivity of the basis functions is an important property. The better the selectivity the less the audible effects of uncanceled aliasing due to coding [1]. Thus, the modulated lapped transform (MLT) is better suited than the LOT for audio coding applications. The *M*-channel MLT is defined as a particular instance of the oddly-stacked TDAC [6] cosine-modulated filter bank:

$$\begin{aligned} p_a(n,k) &= h_a(n)\sqrt{\frac{2}{M}}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ p_s(n,k) &= h_s(n)\sqrt{\frac{2}{M}}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \end{aligned} \tag{4}$$

where $p_a(n,k)$ is the *n,k*-th element of the direct transform matrix $\mathbf{P}_a$ and $p_s(n,k)$ is the *n,k*-th element of the inverse transform matrix $\mathbf{P}_s$. The frequency index *k* varies from 0 to *M*-1, and time index *n* varies from 0 to 2*M*-1. The modulating cosine functions in Eqn. (4) are windowed by $h_a(n)$ for the direct transform (analysis filter bank), and by $h_s(n)$ for the inverse transform (synthesis filter bank). Assuming symmetric and identical windows

$$h_a(n) = h_s(n) = h_s(2M-1-n) \tag{5}$$

then the filter bank in Eqn. (4) achieves perfect reconstruction (which leads to orthogonal basis functions) if the Princen-Bradley condition is satisfied [1],[6]:

$$h_s^2(n) + h_s^2(n+M) = 1 \tag{6}$$

The MLT is defined by the unique window that makes $\sum_n p_a(n,k) = 0$ for all $k \neq 0$ (that is, DC signals are captured entirely by the first basis function), a necessary condition for maximum coding gain [1]. That window is given by

$$h_s(n) = \sin\left[\left(n+\frac{1}{2}\right)\frac{\pi}{2M}\right] \tag{7}$$

A plot of the first two (*k* = 0 and 1) MLT basis functions for *M* = 64 subbands is shown in Fig. 7.

To generate biorthogonal MLTs within the formulation in Eqn. (4), we need to relax the constraint of identical analysis and synthesis windows, as suggested by Smart and Bradley [13], Cheung and Lim [14], Jawerth and Sweldens [15], and Matviyenko [16]. Assuming a symmetrical synthesis window $h_s(n) = h_s(2M-1-n)$, and applying the biorthogonality conditions in Eqn. (1) to Eqn. (4), it is easy to verify that Eqn. (4) generates a modulated lapped biorthogonal transform (MLBT) if the analysis window $h_a(n)$ satisfies the generalized Princen-Bradley conditions [13]–[16]

$$h_a(n) = \frac{h_s(n)}{h_s^2(n) + h_s^2(n+M)}, \quad n = 0, 1, \ldots, M-1 \tag{8}$$

and $h_a(n) = h_a(2M-1-n)$.

If we try to optimize the windows for maximum transform coding gain G_{TC} (using the equations in the Appendix), we arrive at the result that the optimal windows converges to the MLT window of Eqn. (7) as $\rho \to 1$. Therefore, unlike the LBT of Section II, the extra degrees of freedom of the MLBT cannot be used to significantly improve the coding gain. They can be used, however, to improve the frequency selectivity of the synthesis basis functions responses, as in the optimized bases of Matviyenko [16]. They can also be used as a building block for nonuniform MLTs, as we will see later in this Section.

We define the MLBT as the modulated lapped transform of Eqn. (4) with the synthesis window

$$h_s(n) = \frac{1 - \cos\left[\left(\frac{n+1}{M}\right)^{\alpha}\pi\right] + \beta}{2+\beta}, \quad n = 0, 1, \ldots, M-1 \tag{9}$$

and the analysis window defined by Eqn. (8). The parameter α controls mainly the width of the window, whereas β controls its end values. A plot of the analysis and synthesis windows for $\alpha = 0.85$ and $\beta = 0$ is shown in Fig. 8, and Fig. 9 shows some of the basis functions. The MLT window (which can be approximated closely by Eqn. (9) with $\alpha = 0.627$ and $\beta = 0$) is also shown in Fig. 8.

Fig. 10 shows the frequency responses of some of the basis functions of the MLT and MLBT. The main advantage of the MLBT over the MLT is an increase of the stopband attenuation of the synthesis functions, at the expense of a reduction in the stopband attenuation of the analysis functions.

The first sidelobe level of the MLBT frequency responses can be improved by controlling the parameter β. For example, for $\beta = 0.25$ the first sidelobe level of the synthesis responses improves from –27 dB to –30 dB (approximating quite closely the optimal functions of [16]), but the rate of decay of the sidelobes

also reduces, with the lower sidelobe level for the frequency range in Fig. 10 changing from –53 dB to –42 dB. In audio coding applications, decaying stopband gains such as the ones in Fig. 10 are better than the quasi-equiripple responses of [16], because aliasing distortions are more perceptible to the ear for frequency bands that are farther apart.

For $M = 8$, the transform coding gain G_{TC} of the MLBT is actually lower than that of the MLT. For a Gauss-Markov signal with $\rho = 0.95$, the MLBT has a coding gain of only 8.85 dB, significantly lower than that of the LBT of Section II. Thus, the MLBT would not be suitable for image coding applications, but it is well suited for audio coding, as shown in Section V.

A fast computational algorithm for the MLBT can be easily derived from the MLT algorithm of [1], by simply using the corresponding windows for the direct and inverse transforms. The resulting flowgraph is shown in Fig. 11. We note that the butterfly window operators in Fig. 11 are not orthogonal.

The main reason for the particular choice of the MLBT window in Eqn. (9) is the generation of good basis functions in the nonuniform filterbank structure to be discussed next.

Nonuniform Transforms

As discussed above, the main application for modulated transforms is in speech and audio coding applications. For those, blocking and ringing artifacts are even more disturbing than in image coding, due to the high sensitivity of the human ear. Blocking artifacts lead to disturbing periodic clicks, and ringing artifacts lead to reverberation and pre-echo.

The MLT is essentially free from blocking artifacts, but it still leads to noticeable ringing artifacts in audio coding, mainly at low bit rates [17]. They are more noticeable during high-frequency transients, whose durations are shorter than the MLT basis functions. One approach to alleviate the problem, used in the MPEG-2 audio coding standard [18], is to switch to a shorter block length M during transient sounds. Such a switching strategy requires special nonsymmetric windows during the switching periods [18] and increases encoding complexity.

To reduce the ringing artifacts of the MLT/MLTBT, we need to generate shorter high-frequency basis functions. That is equivalent to generating a nonuniform filter bank, in which the high-frequency subbands have larger bandwidths [1]. The hierarchical transform approach [11] that was used in Section II can be employed, but only if an MLT or better is used at the second level of the tree; a DCT would lead to poor stopband rejection of the low-frequency functions [17]. With an MLT in the second level, the basis functions would be about twice as long, increasing delay for real-time processing.

An alternative approach to generate a nonuniform MLT is to merge high-frequency subbands, as suggested originally by Cox [19] and applied specifically for cosine-modulated filter banks by Lee and Lee in [20]. The nonuniform modulated lapped biorthogonal transform (NMLBT) is defined by the flowgraph in Fig. 12. The use of the +1/–1 butterflies makes the NMLBT a perfect reconstruction transform, unlike those in [19] and [20].

As shown in Fig. 12, the first N basis functions are not modified, and each pair of the remaining $N–M$ functions is combined to generate a new pair. The original two basis functions were centered at frequencies $(r+1/2)\pi/M$ and $(r+3/2)\pi/M$, with bandwidths π/M. Their combination via the +1/–1 butterflies generates two new basis functions that are both centered at $(r+1)\pi/M$ with bandwidths $2\pi/M$. Their difference now becomes time localization, as shown in Fig. 13.

We can view the NMLBT construction of Fig. 12 as a dual of the HLBT construction of Fig. 4. With the HLBT, we start with two length-$M/2$ LBTs and combine low-frequency functions via +1/–1 butterflies to generate longer low-frequency functions. With the NMLBT, we start with length-M MLBTs and combine high-frequency functions via +1/–1 butterflies to generate shorter high-frequency functions. The HLBT approach leads to perfect time domain separation and a loss of frequency resolution, whereas the NMLBT approach leads to good frequency resolution but imperfect time domain separation.

In Fig. 13, the ratio of the peak absolute value of a basis function in its central region to its peak values in the central region of its pair is approximately 7.7. That ratio can be viewed as a stopregion attenuation of about 17.7 dB. If we had used the same construction starting with the MLT, the stopregion attenuation would be only 5.8 (15.2 dB).

An example of the frequency responses of the NMLBT is shown in Fig. 14. We note that the high-frequency basis functions have better frequency resolution than the low-frequency ones, because they came from merging of two adjacent bands of the original filterbank. One advantage of the NMLBT over the dual-bank approach suggested by Princen [21] is that perfect reconstruction is preserved, and no special transition filter is needed. Applications of the NMLBT to audio coding are discussed in Section V.

IV. Image Coding Examples

We tested the transforms in Section II with two DCT-based image coding algorithms: the JPEG coder [22], and the embedded zerotree DCT coder (EZDCT) recently introduced by Xiong et.al. [23]. The EZDCT coder replaces the wavelet transform of the SPIHT (set partitioning in hierarchical trees) coder of Said and Pearlman [24] (one of the best image coders reported to date) by DCTs, with appropriate coefficient ordering. As discussed in [12], for both coders we tested the lapped transforms by simply replacing the DCT by

the LOT, LBT, and HLBT, without any other change in the coder.

The coders were tested with the “lena2” image (a 256x256 cut from the well known 512x512 “lena” image). The peak signal-to-noise ratio (PSNR) results are shown in Fig. 15. For the JPEG coder, the HLBT has a PSNR improvement of 0.4 dB over the DCT, whereas the LBT shows a PSNR gain of about 0.7 dB at rates around 0.5 bits/sample. These results are consistent with the coding gain calculations of Section II.

For the embedded zerotree coder, the performance of the HLBT is quite close to that of the LBT. Both perform closely (within 0.8 and 0.4 dB, respectively) to the optimized wavelet-based SPIHT coder, but the HLBT-based embedded coder is faster. Compared to the DCT, the HLBT improves the embedded coder by 0.6 dB.

Fig. 16 shows 160x160 portions of the reconstructed images, for the rate of 0.5 bits/sample. In the top row (JPEG results), we see that all lapped transforms have less blocking than the DCT. The LOT still shows some artifacts, and the LBT is virtually free from blocking, but both show more ringing artifacts than the DCT. The HLBT has less blocking and less ringing than the DCT. The embedded zerotree coded images in Fig. 16 show the significant improvement achieved with the biorthogonal lapped transforms. The results with the HLBT and LBT images are quite similar, as expected from the curves in Fig. 15, and they both represent a visible improvement over the DCT-coded image.

V. Audio Coding Examples

To test the performance of the NMLBT for audio coding, we simulated an audio encoding system in the following way. For each input signal block, the direct transform is computed, and all transform coefficients are uniformly quantized. The quantization step size for the block is chosen as the product of a constant parameter γ times the block standard deviation. The quantized coefficients are inverse transformed and added to the reconstructed signal. The parameter γ is chosen such that the measured entropy of the quantized transform coefficients for all blocks equals a prescribed value.

As a test signal, we used an audio waveform sampled at 16 kHz, containing four concatenated sound segments. Each segment had a length of about 1.5 s, and came from the following sources: castanets, horns, female speech, and male speech. We ran the coder simulation with three transforms: the MLT for $M = 64$ (MLT_{64}), the MLT for $M = 32$ (MLT_{32}), and the NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The quantized coefficient entropy was fixed at 1.5 bits/sample by proper choice of γ in each case. The average segmental signal-to-noise ratio (SSNR) was 18.0, 17.1, and 17.8 dB, respectively. So, although 75% of the basis functions of the NMLBT had about the same length of the MLT_{32}, the NMLBT performance was

much closer to that of the MLT_{64}.

An example of the results for the castanets segment is shown in Fig. 17. In terms of SSNR for the segment shown, the NMLBT is about 0.5 dB better than the MLT_{64}, and 1.1 dB better than the MLT_{32}. The MLT_{32} has the shortest pre-echo, but the highest reconstruction error in the pre-echo region. The NMLBT has less pre-echo artifacts than the MLT_{64}.

Results for a portion of the female segment are shown in Fig. 18. To stress the transient signal performance, that segment shows the onset of an utterance. The SSNR for the NMLBT is about the same as that for the MLT_{64}, and about 1.5 dB higher than that for the MLT_{32}. As with the castanets signal, the NMLBT leads to less pre-echo artifacts than the MLT_{64},

The better performance of the NMLBT for transient signals comes at only a small penalty in performance for approximately stationary signals. This can be seen in Fig. 19, which shows the results for a portion of the male segment. The SSNR of the NMLBT is only 0.2 dB below that of the MLT_{64}, and about 1.1 dB higher than that for the MLT_{32}.

From the audio coding results, it is clear that the NMLBT is a good alternative to the MLT, leading to better reproduction of transient sounds, with less pre-echo. That is achieved with only a small penalty in SNR performance (around 0.1–0.2 dB) for stationary signals.

For adaptive audio coding, an interesting property of the NMLBT is that the choice of N can change on a block-by-block basis. During stationary regions, we can make $N = 0$, which turns the NMLBT into a length-M MLBT. During transient sounds, we can choose N among a few predetermined values, for best reproduction of the block. By setting $N = M$, for example, we can effectively halve the length of all basis functions. In that way, we have a time-varying adaptive transform which maintains good frequency and time resolutions at all times, without the use of transitional filter banks. Since the SNR performance of the NMLBT is better than the MLT for the same block size M for transient signals, the adaptive approach above would lead to a better SNR performance than a fixed MLT.

VI. Conclusions and Further Directions

We have introduced new families of lapped transforms, based on biorthogonal constructions. The lapped biorthogonal transform (LBT) is generated from the LOT flowgraph by a simple modification, which leads to higher transform coding gains and less blocking artifacts. The hierarchical lapped biorthogonal transform (HLBT), built from the LBT, has two levels of resolution, which leads to shorter high-frequency

functions and smoother low-frequency functions than the LOT. For image coding applications, the HLBT is a better choice than the DCT, because the HLBT has a higher coding gain, much less blocking and less ringing artifacts. These advantages come at a computational overhead of only 30% (compared to 80% for the LOT or LBT) over the DCT. For the embedded zerotree coder, replacing the DCT by the HLBT can improve the PSNR performance by more than 0.5 dB. An HLBT-based embedded zerotree coder approaches the performance of the best wavelet image coders reported to date, with the advantage of keeping the bulk of its computation on the DCTs that are part of the HLBT flowgraph. Thus, an embedded zerotree HLBT coder can leverage existing DCT software or hardware, and runs faster than the EZW or SPIHT coders (because good wavelet decompositions use more multiplies and adds per sample than the HLBT).

The modulated lapped biorthogonal transform (MLBT) is obtained by relaxing the constraint of identical analysis and synthesis windows in the MLT filterbank. The use of different windows does not improve on transform coding gain (for low-order autoregressive Gauss-Markov signals), but leads to improved stopband attenuation. Using a simple subband merging technique, the nonuniform modulated lapped biorthogonal transform (NMLBT) achieves two levels of resolution, by effectively halving the time duration and doubling the bandwidth of the high-frequency basis functions. Audio coding simulations show that an NMLBT-based coder can achieve SNR performance close to that of the MLT, but with a better reproduction of transient sounds. The NMLBT also allows for efficient implementation of time-varying transforms with good frequency and time resolutions at all times, without the use of transitional filters.

There are many directions in which further research could improve on the results presented in this paper. For the HLBT, JPEG encoding performance may be improved by optimization of the quantization tables, for example. For the NMLBT, new window designs and optimized coefficients for merging of subbands may lead to better time/frequency resolution trade-offs.

Appendix

Computation of Coding Gain for Low Bit Rates

Consider a Gaussian input signal x with variance σ_x^2 and block autocorrelation matrix $\mathbf{R}_{xx}$. The variance of the transform coefficients are given by $\sigma_{y_i}^2 = \mathbf{P}_{ai}^{\mathrm{T}}\mathbf{R}_{xx}\mathbf{P}_{ai}$, where $\mathbf{P}_{ai}$ is the i-th column of the direct transform matrix $\mathbf{P}_a$ [1].

Assuming the i-th transform coefficient is quantized with B_i bits, its quantization distortion is given by $\sigma_{q_i}^2 = 2^{-2B_i}\sigma_{y_i}^2$, assuming ideal quantizers [2]. Under the mild assumption that the quantization noises q_i are uncorrelated, the total output noise variance is given by [8]

$$\sigma_n^2 = \sum_{i=0}^{M-1} \sigma_{q_i}^2 \left\| f_i \right\|^2 \tag{A.1}$$

where $\left\| f_i \right\|^2 = \left\| \mathbf{P}_{si} \right\|^2$ is the norm of the i-th synthesis basis function.

An optimal bit allocation procedure finds the set of B_i that minimizes σ_n^2, subject to the condition that the total bit rate should match the B available bits for each block:

$$\sum_{i=0}^{M-1} B_i = B \tag{A.2}$$

Using standard nonlinear optimization techniques, the solution is a simple variation of the usual log-variance formula [2]

$$B_i = \begin{cases} \lambda + \frac{1}{2}\log_2(\sigma_{y_i}^2 \left\| f_i \right\|^2), & \text{if positive} \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

where λ is a Lagrange multiplier that is chosen such that Eqn. (A.2) is satisfied. The transform coding gain is then given in dB by $G_{TC}(B) = 10\log_{10}(2^{-2B}\sigma_x^2/\sigma_n^2)$, i.e. the ratio of the noise for straight quantization of the input signal to the output noise in Eqn. (A.1).

References

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

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

[3] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice Hall, 1992.

[4] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video, and Audio Coding.* Englewood Cliffs, NJ: Prentice Hall, 1996.

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

[6] J. Princen, A. W. Johnson, and A. B. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Dallas, Apr. 1987, pp. 2161–2164.

[7] H. S. Malvar, "Extended cosine bases and application to audio coding," *Computational and Applied Math.,* vol. 15, no. 2, pp. 111–-123, 1996.

[8] S. O. Aase and T. A. Ramstad, "On the optimality of nonunitary filter banks in subband coders," *IEEE Trans. on Image Processing,* vol. 4, pp. 1585–1591, Dec. 1995.

[9] R. W. Young and N. G. Kingsbury, "Video compression using lapped transforms for motion estimation/compensation and coding," *Proc. SPIE Conf. on Visual Communications and Image Processing*, Boston, Nov. 1992, pp. 276–288.

[10] S. C. Chan, "The generalized lapped transform (GLT) for subband coding applications," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, May 1995, pp. 1508–1511.

[11] H. S. Malvar, "Efficient signal coding with hierarchical lapped transforms," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Albuquerque, Apr. 1990, pp. 1519–1522.

[12] H. S. Malvar, "Lapped biorthogonal transforms for transform coding with reduced blocking and ringing artifacts," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Munich, Apr. 1997, pp. 2421–2424.

[13] G. Smart and A. B. Bradley, "Filter bank design based on time domain aliasing cancellation with non-identical windows," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Adelaide (Australia), Apr. 1994, pp. III-185–III-188.

[14] S. Cheung and J. S. Lim, “Incorporation of biorthogonality into lapped transforms for audio compression,” *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, May 1995, pp. 3079–3082.

[15] B. Jawerth and W. Sweldens, “Biorthogonal smooth local trigonometric bases,” *J. Fourier Analysis and Appl.*, vol. 2, no. 2, pp. 109–133, 1995.

[16] G. Matviyenko, “Optimized local trigonometric bases,” *Appl. and Computational Harmonic Analysis,* vol. 3, no. 4, pp. 301–323, 1996.

[17] L. F. C. Vargas and H. S. Malvar, “ELT-based wavelet coding of high-fidelity audio signals,” *Proc. IEEE Int. Symp. Circuits and Systems*, Chicago, May 1993, pp. 124–127.

[18] D. Pan, “A tutorial on MPEG audio compression,” *IEEE Multimedia,* vol. 2, no. 2, pp. 60–74, Summer 1995.

[19] R. V. Cox, “The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1090–1096, Oct. 1986.

[20] J.-J. Lee and B. G. Lee, “A design of nonuniform cosine modulated filter bank,” *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 42, pp. 732–737, Nov. 1995.

[21] J. Princen, “The design of nonuniform modulated filterbanks,” *IEEE Trans. on Signal Processing,* vol. 43, pp. 2550–2560, Nov. 1995.

[22] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1992.

[23] Z. Xiong, O. G. Guleryuz, and M. T. Orchard, “A DCT-based embedded image coder,” *IEEE Signal Processing Letters,* vol. 3, Nov. 1996, pp. 289–290.

[24] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology,* vol. 6, June 1996, pp. 243–250.

Figures

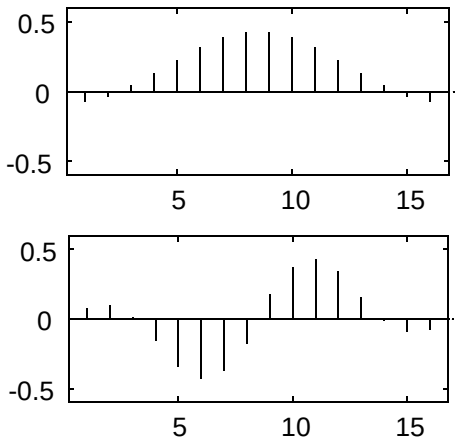

Figure 1. The first two LOT basis functions for $M = 8$.

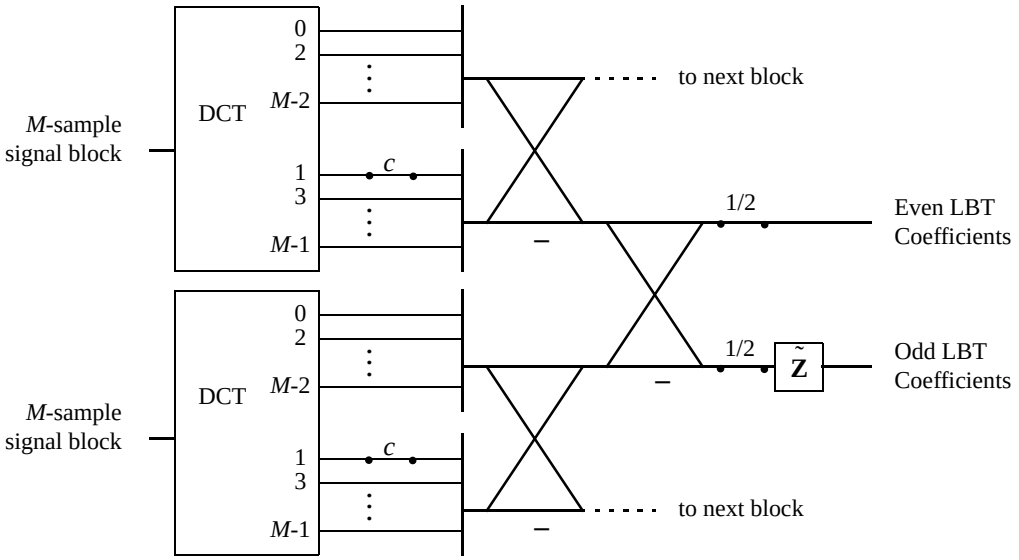

Figure 2. Flowgraph of the LBT, with $\tilde{\mathbf{Z}}$ defined in [1],[5]. For the direct transform (left to right) $c = \sqrt{2}$ and for the inverse transform (right to left) $c = \sqrt{1/2}$.

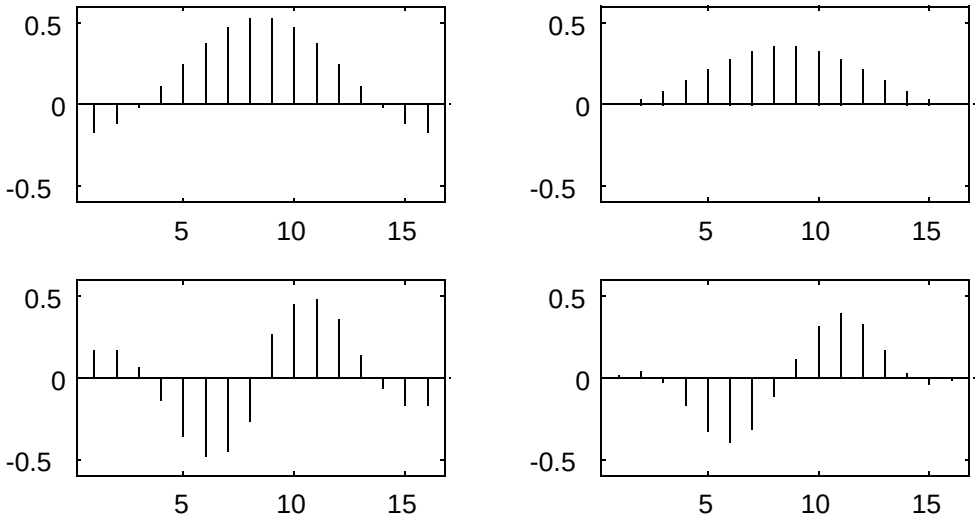

Figure 3. The first two LBT basis functions for $M = 8$. Left: analysis (direct transform); right: synthesis (inverse transform).

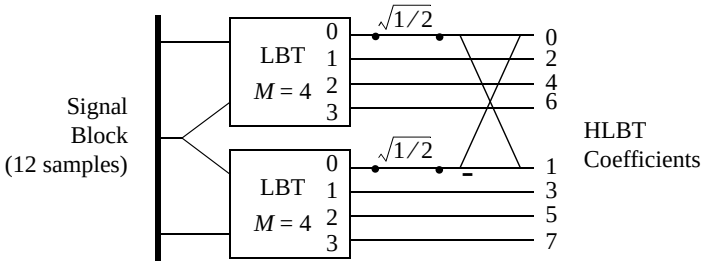

Figure 4. Simplified block diagram of the HLBT for $M = 8$. The length-2 DCT in the second level corresponds to the +1/-1 butterfly with the $\sqrt{1/2}$ scaling factors (those can be replaced by factors equal to one in the inverse transform and 1/2 in the direct transform, for example).

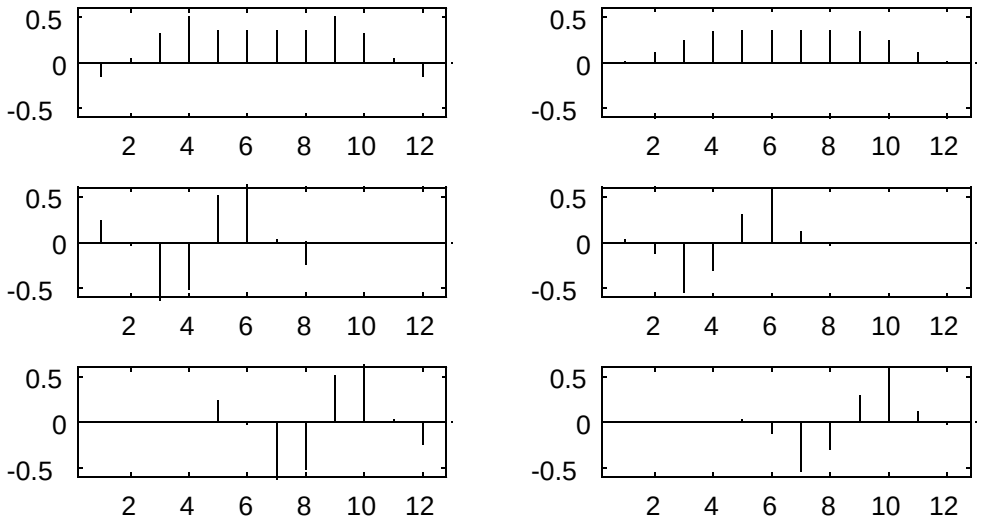

Figure 5. HLBT basis functions # 0, 2, and 3, for $M = 8$. Left: analysis (direct transform); right: synthesis (inverse transform).

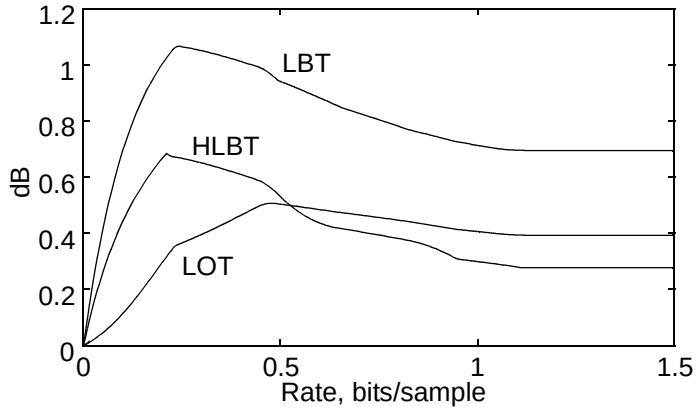

Figure 6. Improvement in transform coding gain over the DCT, as a function of bit rate, for several lapped transforms. Gauss-Markov input signal with $\rho = 0.95$. For bidimensional signals (e.g. images) the gain is approximately doubled [1], [2].

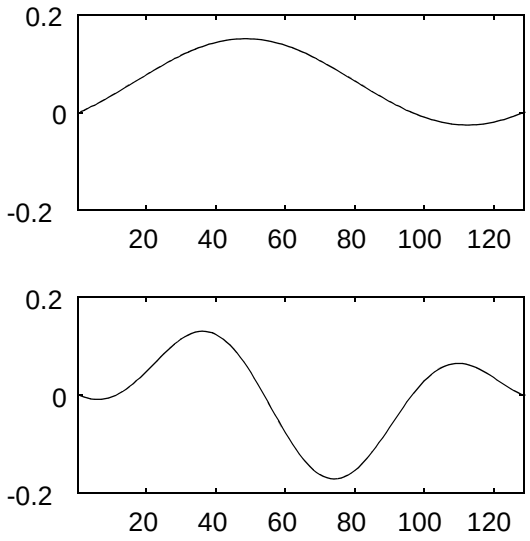

Figure 7. The first two MLT basis functions for $M = 64$.

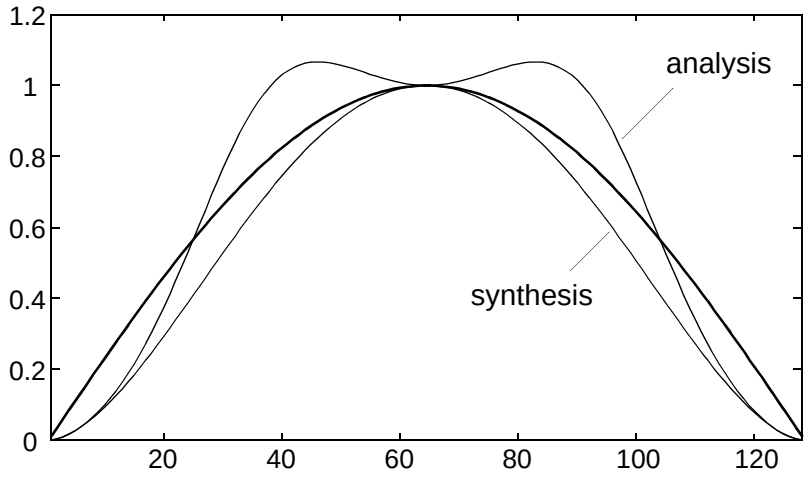

Figure 8. MLBT windows for $M = 64$, $\alpha = 0.85$, and $\beta = 0$. For comparison, the MLT sine window is shown as a heavier line.

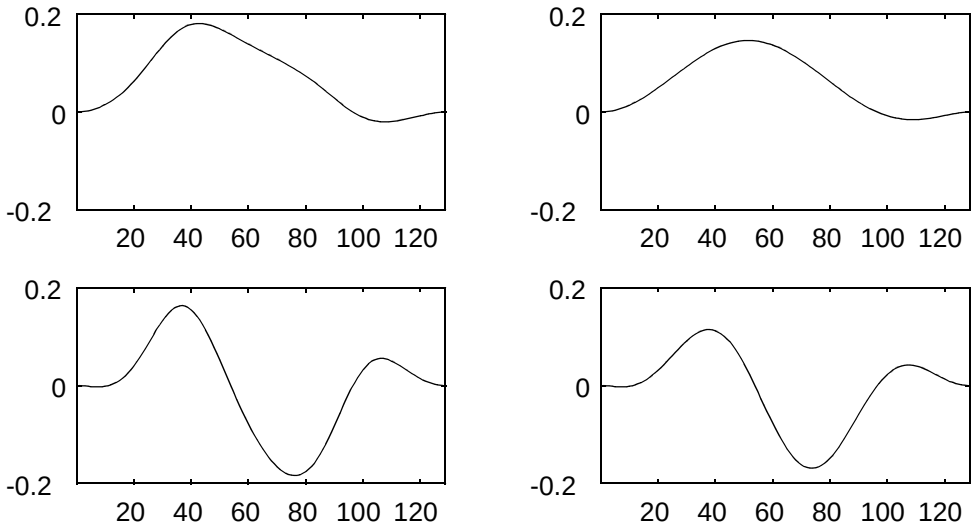

Figure 9. The first two MLBT basis functions for $M = 64$, $\alpha = 0.85$, and $\beta = 0$. Left: analysis (direct transform); right: synthesis (inverse transform).

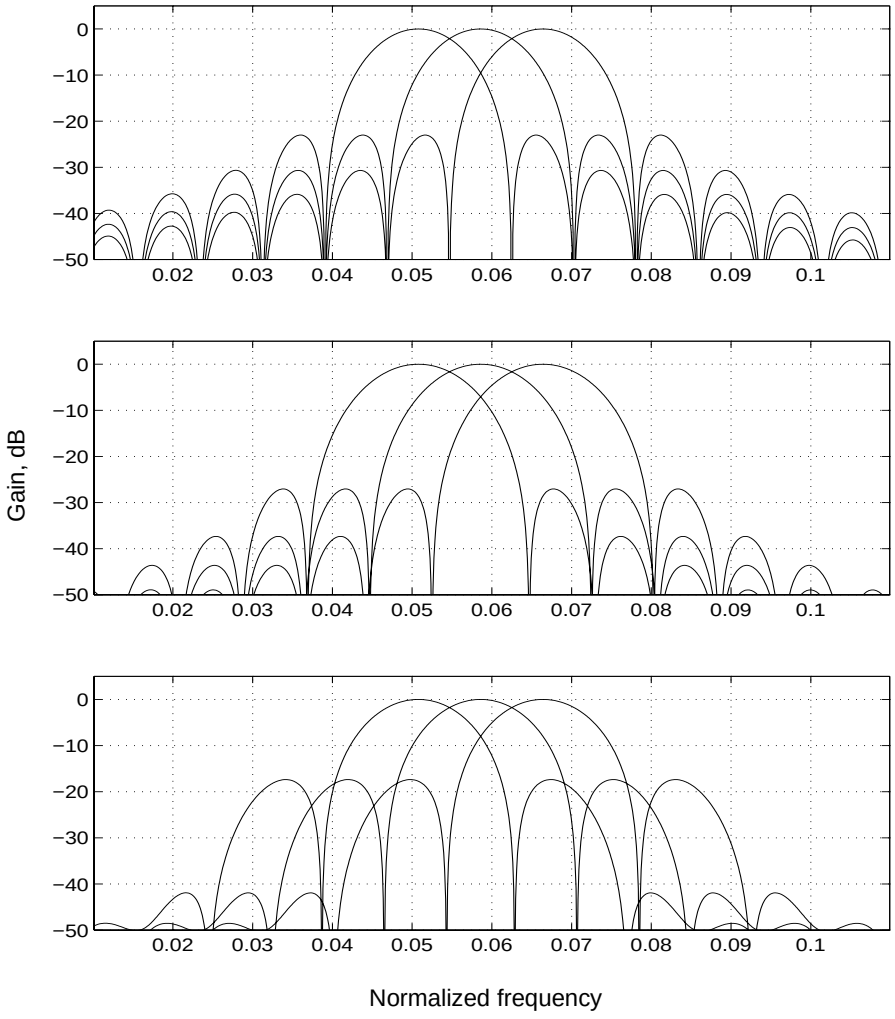

Figure 10. Frequency responses for basis functions #7, #8, and #9, for M = 64. Top: MLT; middle: MLBT, synthesis (α = 0.85, β = 0); bottom: MLBT, analysis.

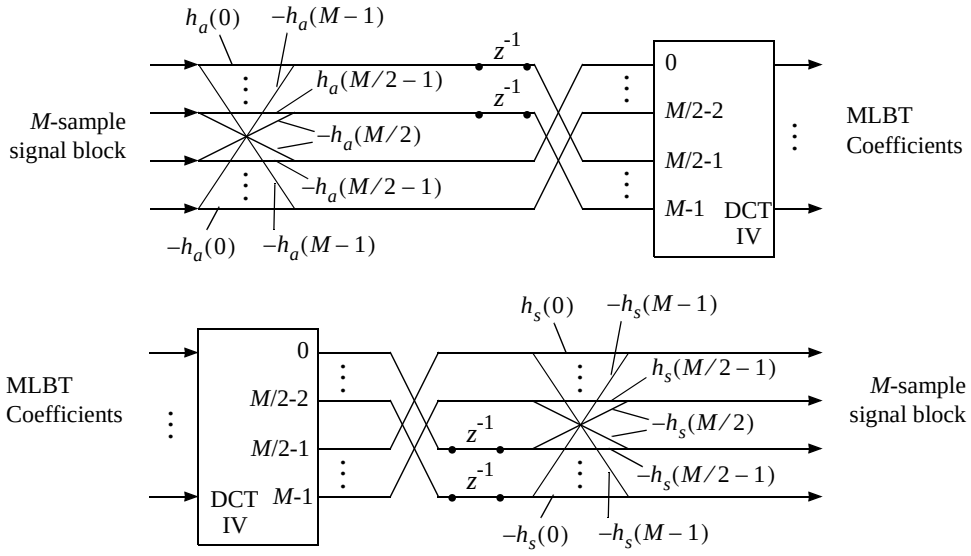

Figure 11. Flowgraph of the fast MLBT. Top: direct transform (analysis); bottom: inverse transform (synthesis).

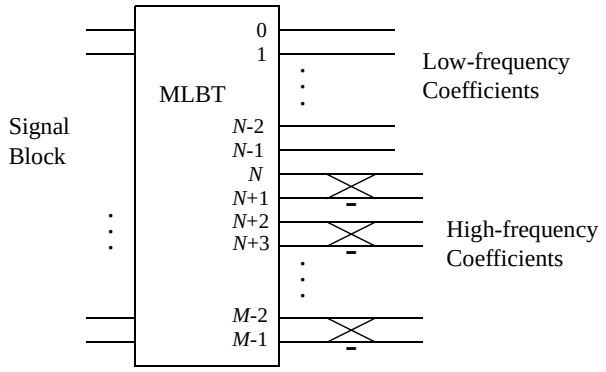

Figure 12. Simplified block diagram of the NMLBT. Each pair of high-frequency coefficients is merged into two new coefficients that correspond to the same subband and different time localizations.

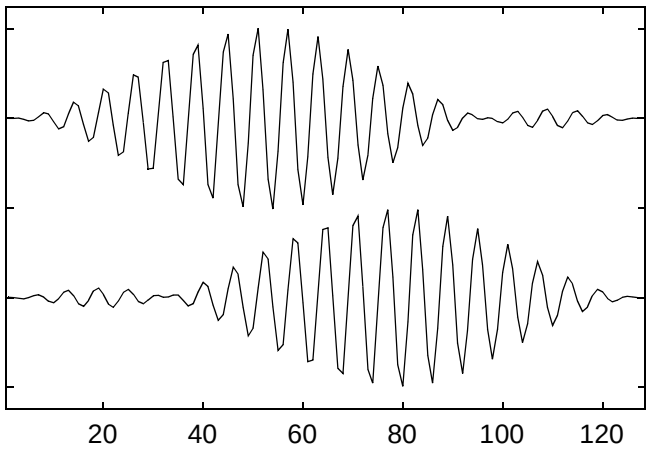

Figure 13. An example of NMLBT synthesis basis functions #20 and #21, for $M = 64$ and $N = 16$, based on an MLBT with $\alpha = 0.85$, and $\beta = 0$. They correspond to the same frequency subband, but have different time localizations.

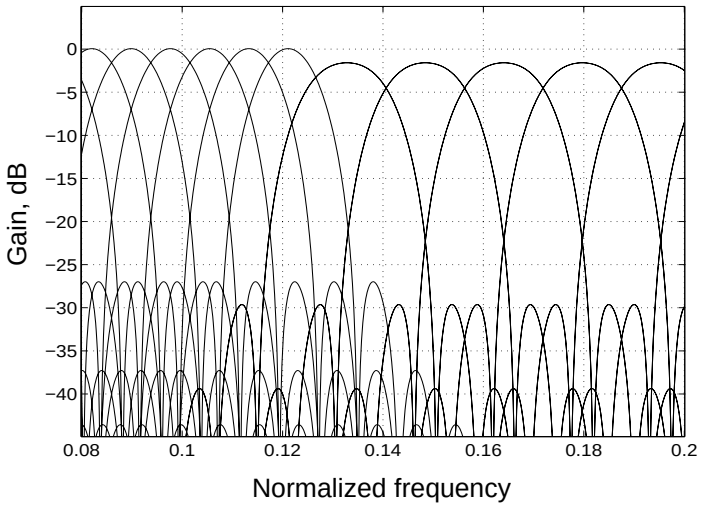

Figure 14. Frequency responses of some of the subbands of the NMLBT synthesis filterbank, for $M = 64$ and $N = 16$, based on an MLBT with $\alpha = 0.85$, and $\beta = 0$.

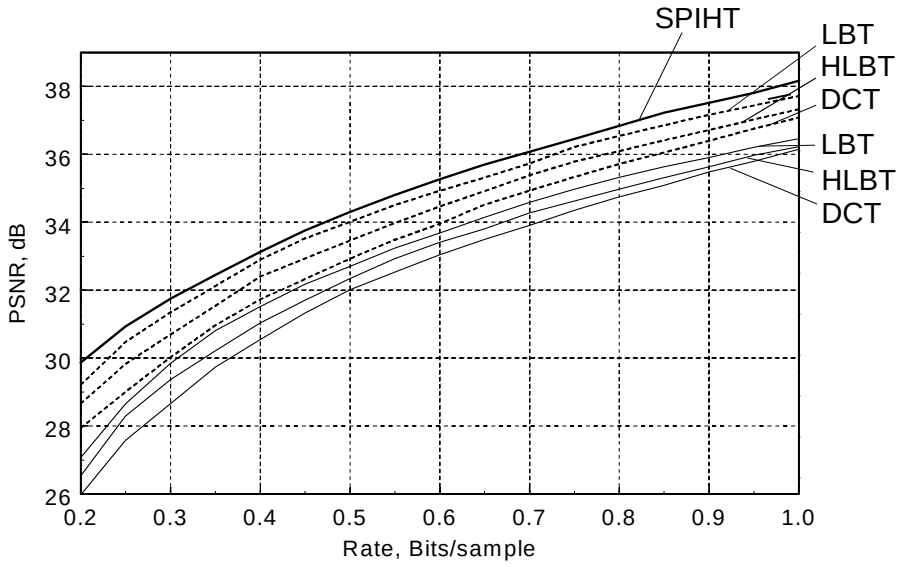

Figure 15. Peak SNR for Performance for transform-based coders with N = 8, for the "lena2" image. Dashed lines: embedded zerotree; solid lines: JPEG. Reference: SPIHT (top solid line).

Figure 16. Coding examples for image “lena2” at 0.5 bits/pixel. Top row, left to right: original, JPEG encoding with DCT, and JPEG/LOT, PSNR (dB) = 32.0 and 32.3. Middle row: JPEG/LBT, JPEG/HLBT, and EZDCT with PSNR (dB) = 32.7, 32.4, and 32.9. Bottom row: EZ/LOT, EZ/LBT, and EZ/HLBT, with PSNR (dB) = 33.4, 34.0, and 33.5.

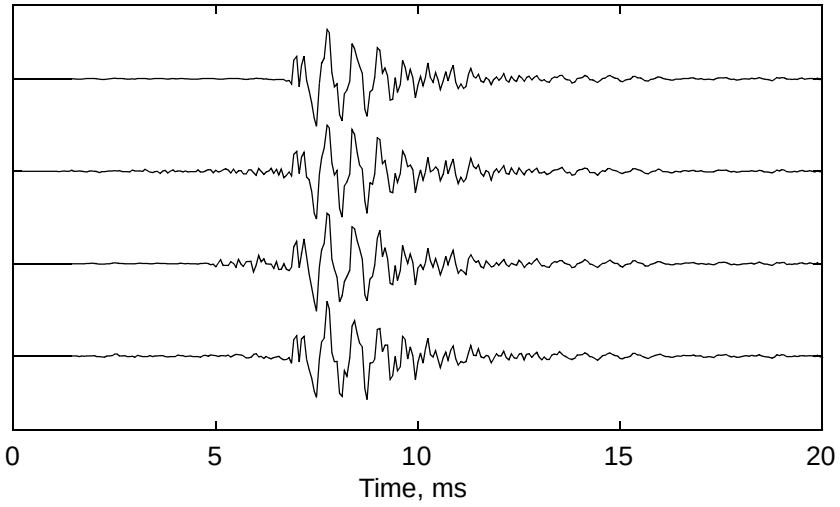

Figure 17. Audio coding results at 1.5 bits/sample for a castanets sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 14.7, 14.1, and 15.2 dB, respectively.

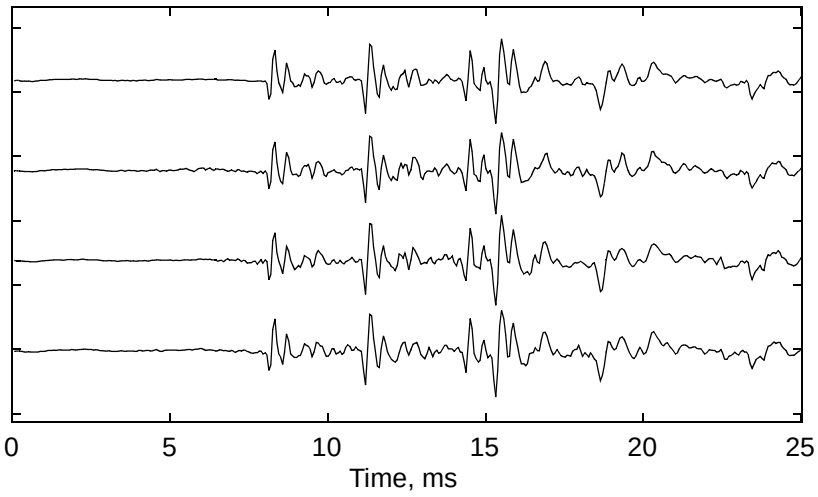

Figure 18. Audio coding results at 1.5 bits/sample for a female speech sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 16.2, 14.6, and 16.1 dB, respectively.

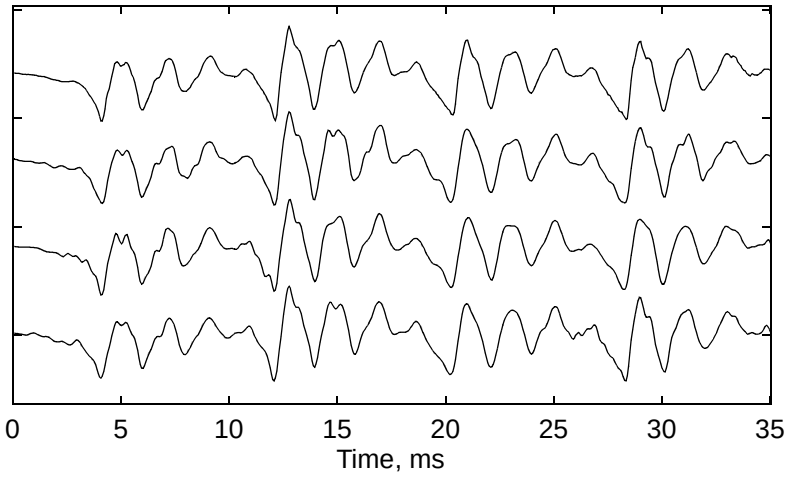

Figure 19. Audio coding results at 1.5 bits/sample for a male speech sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 19.6, 18.3, and 19.4 dB, respectively.

Footnotes

Manuscript received _____; revised _____. The associate editor coordinating the review of this paper and approving it for publication was Prof. Keshab K. Parhi.

The author was with PictureTel Corporation, 100 Minuteman Road, Andover, MA 01810; he is now with Microsoft Corporation, One Microsoft Way, Redmond, WA 98052.

Figure Captions

Figure 1. The first two LOT basis functions for $M = 8$.

Figure 2. Flowgraph of the LBT, with $\tilde{\mathbf{Z}}$ defined in [1],[5]. For the direct transform (left to right) $c = \sqrt{2}$ and for the inverse transform (right to left) $c = \sqrt{1/2}$.

Figure 3. The first two LBT basis functions for $M = 8$. Left: analysis (direct transform); right: synthesis (inverse transform).

Figure 4. Simplified block diagram of the HLBT for $M = 8$. The length-2 DCT in the second level corresponds to the +1/-1 butterfly with the $\sqrt{1/2}$ scaling factors (those can be replaced by factors equal to one in the inverse transform and 1/2 in the direct transform, for example).

Figure 5. HLBT basis functions # 0, 2, and 3, for $M = 8$. Left: analysis (direct transform); right: synthesis (inverse transform).

Figure 6. Improvement in transform coding gain over the DCT, as a function of bit rate, for several lapped transforms. Gauss-Markov input signal with $\rho = 0.95$. For bidimensional signals (e.g. images) the gain is approximately doubled [1], [2].

Figure 7. The first two MLT basis functions for $M = 64$.

Figure 8. MLBT windows for $M = 64$, $\alpha = 0.85$, and $\beta = 0$. For comparison, the MLT sine window is shown as a dotted line

Figure 9. The first two MLBT basis functions for $M = 64$, $\alpha = 0.85$, and $\beta = 0$. Left: analysis (direct transform); right: synthesis (inverse transform).

Figure 10. Frequency responses for basis functions #7, #8, and #9, for $M = 64$. Top: MLT; middle: MLBT, synthesis ($\alpha = 0.85$, $\beta = 0$); bottom: MLBT, analysis.

Figure 11. Flowgraph of the fast MLBT. Top: direct transform (analysis); bottom: inverse transform (synthesis).

Figure 12. Simplified block diagram of the NMLBT. Each pair of high-frequency coefficients is merged into two new coefficients that correspond to the same subband and different time localizations.

Figure 13. An example of NMLBT synthesis basis functions #20 and #21, for $M = 64$ and $N = 16$, based on an MLBT with $\alpha = 0.85$, and $\beta = 0$. They correspond to the same frequency subband, but have different time localizations.

Figure 14. Frequency responses of some of the subbands of the NMLBT synthesis filterbank, for $M = 64$ and $N = 16$, based on an MLBT with $\alpha = 0.85$, and $\beta = 0$.

Figure 15. Peak SNR for Performance for transform-based coders with N = 8, for the "lena2" image. Dashed lines: embedded zerotree; solid lines: JPEG. Reference: SPIHT (top solid line).

Figure 16. Coding examples for image "lena2" at 0.5 bits/pixel. Top row, left to right: original, JPEG encoding with DCT, and JPEG/LOT, PSNR (dB) = 32.0 and 32.3. Middle row: JPEG/LBT, JPEG/HLBT, and EZDCT with PSNR (dB) = 32.7, 32.4, and 32.9. Bottom row: EZ/LOT, EZ/LBT, and EZ/HLBT, with PSNR (dB) = 33.4, 34.0, and 33.5.

Figure 17. Audio coding results at 1.5 bits/sample for a castanets sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 14.7, 14.1, and 15.2 dB, respectively.

Figure 18. Audio coding results at 1.5 bits/sample for a female speech sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 16.2, 14.6, and 16.1 dB, respectively.

Figure 19. Audio coding results at 1.5 bits/sample for a male speech sound sampled at 16 kHz. From top to bottom: original and coded with the MLT for $M = 64$, MLT for $M = 32$, and NMLBT for $M = 64$, $N = 16$, $\alpha = 0.85$, and $\beta = 0$. The segmental SNR for the reconstructed signals are 19.6, 18.3, and 19.4 dB, respectively.