INTEGER LOW DELAY AND MDCT FILTER BANKS

Ralf Geiger, Gerald Schuller

Fraunhofer AEMT, Ilmenau, Germany
{ggr,shl}@emt.iis.fhg.de

ABSTRACT

Recently lifting-based integer approximations of filter banks have received much attention, especially in the field of image coding. This paper focuses on the application of these techniques to cosine modulated filter banks for audio coding, including not only the Modified Discrete Cosine Transform (MDCT) but also low delay filter banks. Applications of these integer filter banks include lossless audio coding and backward compatible lossless enhancement of MDCT-based perceptual audio coding schemes, such as MPEG-2/4 AAC.

1. INTRODUCTION, GOAL

Perceptual audio coders, such as MPEG-2/4 AAC [1, 2] or MPEG-1 layer-3 (MP3) are used to transmit audio data or to store it on solid state players, for instance. On the other hand, for archiving purposes, and for editing or producing audio content, for instance, lossless coding is more suitable. To streamline the production/archiving/transmission process it would be useful to have a lossless coder with an embedded perceptual coder (or a perceptual coder with an ”enhancement layer” to obtain lossless coding).

Modern perceptual audio coders usually use cosine modulated filter banks such as the Modified Discrete Cosine Transform (MDCT) to obtain a block-wise frequency representation of the audio signal. These filter banks usually produce floating point values even for integer input samples, which are then quantized according to perceptual criteria. Applying these floating point filter banks to lossless audio coding leads to problems. Simply rounding the integer subband values lead to round errors in the reconstructed signal. One approach could be to make the quantization fine enough to allow restoring the original integer values. This has the disadvantage of an unnecessary high bit-rate. An approach to obtain an embedded perceptual coder is to decode the perceptually coded signal in the encoder, compute the error to the original, and to code and transmit this error as enhancement layer in the time domain. This has the disadvantage that perceptual decoders often don’t have a bit-exact conformance, which would destroy the lossless property.

A possible solution for filter bank based lossless audio coding is the use of integer-to-integer filter banks with perfect reconstruction. They can be used to generate an enhancement layer to obtain lossless coding. For these filter banks it is desirable to have energy conservation on average, to avoid an unnecessary high bit-rate. In this paper we will describe an integer-to-integer version of cosine modulated filter banks, such as MDCT and low delay filter banks.

2. PRESENT STATE

A powerful tool for obtaining integer-to-integer filter banks is given by the so-called lifting scheme [3]. This technique allows to approximate Givens Rotations by mapping integers to integers in a reversible way. To achieve this, a Givens Rotation is decomposed into lifting steps in the following way:

$$\begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}=\begin{pmatrix}1 & \frac{\cos\alpha-1}{\sin\alpha}\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \sin\alpha & 1\end{pmatrix}\begin{pmatrix}1 & \frac{\cos\alpha-1}{\sin\alpha}\\ 0 & 1\end{pmatrix}$$

Figure 1 illustrates this decomposition. In every lifting step a rounding function can be included to stay in the integer domain. This rounding doesn’t affect the perfect reconstruction property, because every lifting step can be inverted by subtracting the value that has been added. This is illustrated in Figure 2.

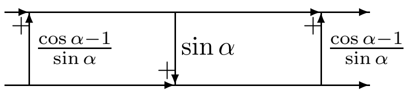
+
$\frac{\cos\alpha-1}{\sin\alpha}$
$\sin\alpha$
+
+
$\frac{\cos\alpha-1}{\sin\alpha}$

Fig. 1. Givens rotation using three lifting steps

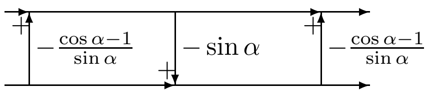
+
$-\frac{\cos\alpha-1}{\sin\alpha}$
$-\sin\alpha$
+
+
$-\frac{\cos\alpha-1}{\sin\alpha}$

Fig. 2. Inverse Givens rotation using three lifting steps

So every filter bank that can be decomposed into Givens Rotations or into only lifting steps can be approximated by a lossless integer-to-integer version. For DCT-based filter banks focusing on image coding this technique was described in [4, 5]. The lifting scheme can also be utilized for the Fast Fourier Transform (FFT), as shown in [6]. In [7, 8, 9] a lifting scheme based integer-to-integer MDCT has been presented. In [8] this IntMDCT was used to obtain a scalable perceptual and lossless audio coding scheme.

Besides the MDCT also non-orthogonal versions of modulated filter banks have been considered for audio coding applications due to their possibility to reduce the overall delay, or to more closely adapt to psycho-acoustic requirements. These filter banks can be designed with a decomposition of its structure into a DCT and a cascade of pre-processing steps. In a polyphase representation these pre-processing steps show up as maximum-delay and zero-delay matrices and a diagonal factor matrix [10], see Figure 3. The maximum-delay and zero-delay matrices can be seen as a set of lifting steps. This makes them very suitable for an integer-to-integer implementation.

3. NEW APPROACH

In this paper a general approach for integer-to-integer approximations of cosine modulated filter banks for audio coding applications will be presented, and the feasibility of this approach for MDCT and low delay filter banks will be investigated. As it turns out the zero-delay and maximum-delay pre-processing steps are inherently suitable for integer-to-integer filter banks. The diagonal matrix of the pre-processing step can also be decomposed into a suitable set of lifting steps. This makes it convenient to design, for instance, integer-to-integer low delay filter banks, or an integer MDCT. We will show what restriction the integer-to-integer requirement imposes on the pre-processing diagonal factor matrix, and what limitations can be concluded from it to integer-to-integer cosine modulated filter banks. For instance, the decomposition of the diagonal matrix into lifting steps leads to constraint which shows that only cosine modulated filter banks with identical baseband prototypes for analysis and synthesis can be approximately energy conserving integer-to-integer filter banks. In our case we mean with approximately energy conserving that the determinant of the polyphase matrix is one or a pure delay. This also means that the polyphase matrix can be decomposed into lifting steps, without extra factors, since each lifting steps polyphase matrix has a determinant of one.

Our filter bank impulse responses are

$$h_k(n) = h(n) \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi}{N}(k+0.5)(n+0.5+n_a)\right) \tag{1}$$

$$g_k(n) = g(n) \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi}{N}(k+0.5)(n+0.5-N+n_s)\right) \tag{2}$$

with $k = 0, \ldots, N-1$, and where $h(n), g(n)$ are our analysis and synthesis baseband impulse responses respectively. We can write the polyphase matrices for the analysis $\mathbf{P_a}(z)$ and synthesis $\mathbf{P_s}(z)$ as [10]

$$\mathbf{P_a}(z) = \mathbf{S}^{n_a}(z) \cdot \mathbf{F_a}(z) \cdot \mathbf{T} \tag{3}$$

$$\mathbf{P_s}(z) = \mathbf{T}^{-1} \cdot \mathbf{F_s}(z) \cdot \mathbf{S}^{n_s}(z) \tag{4}$$

where $\mathbf{S}(z)$ is a shift matrix that advances a block or vector by one sample and $\mathbf{diag}$ is an $N \times N$ diagonal matrix.

$$\mathbf{S}(z) := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & 0 & \ddots & 0 \\ 0 & \vdots & \vdots & & 1 \\ z & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\begin{aligned} \mathbf{F_a}(z) = \mathbf{diag}[P_0(-z^2), \ldots, P_{M-1}(-z^2)] \cdot \mathbf{J} + \\ + z^{-1} \cdot \mathbf{diag}[P_{2M-1}(-z^2), \ldots, P_M(-z^2)] \end{aligned} \tag{5}$$

$$\begin{aligned} \mathbf{F_s}(z) = \mathbf{diag}[P'_0(-z^2), \ldots, P'_{M-1}(-z^2)] \cdot \mathbf{J} \\ - z^{-1}\mathbf{diag}[P'_M(-z^2), \ldots, P'_{2M-1}(-z^2)], \end{aligned} \tag{6}$$

where

$$P_i(z) = \sum_{m=-\infty}^{\infty} h(m2N + i - n_a)(-1)^m z^{-m} \tag{7}$$

$$P'_i(z) = \sum_{m=-\infty}^{\infty} g(m2N + i - n_s)(-1)^m z^{-m} \tag{8}$$

$\mathbf{F_a}(z)$ and $\mathbf{F_a}(z)$ can be further decomposed into simpler matrices. They can be seen as building blocks for filter banks. These matrices already consist of lifting steps.

The first type is Zero-Delay matrices. They increase the filter length but not the system delay

$$\mathbf{L}_i(z) := \mathbf{J} + \mathbf{diag}(l_0^i, \ldots, l_{M/2-1}^i, 0, \ldots, 0) \cdot z^{-1},$$

Since it consists of lifting steps, the inverse is simply

$$\mathbf{L}_i^{-1}(z) = \mathbf{J} - \mathbf{diag}(0, \ldots, 0, l_{M/2-1}^i, \ldots, l_0^i) \cdot z^{-1}.$$

The second type is Maximum-Delay matrices. These also increase the filter length, but especially the system delay, to obtain filter banks with higher system delay. They need a multiplication with z^{-1} for causality.

$$\mathbf{L}_i(z^{-1}) \cdot z^{-1}.$$

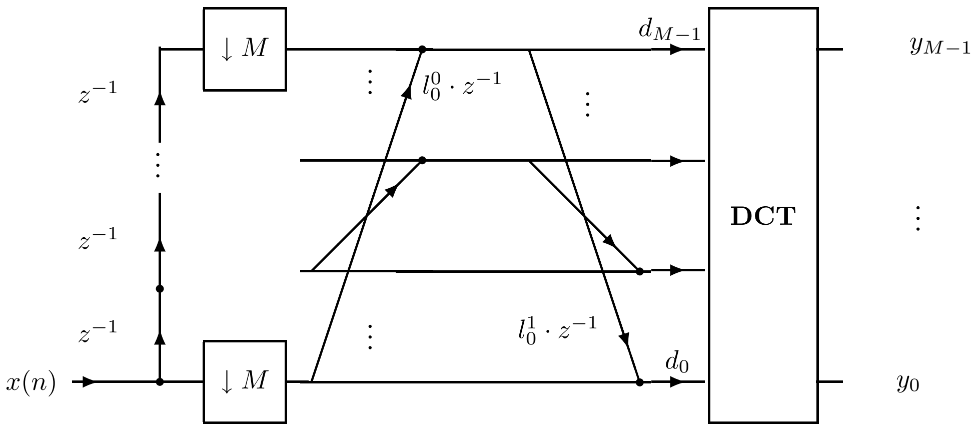

Fig. 3. Example of a low delay analysis filter bank. $x(n)$ is the input time signal, y_0 to y_{M-1} are the subband signals. The coefficients l_0^0 and l_0^1 are part of the zero delay pre-processing steps, d_0 and d_{M-1} are part of the diagonal factor matrix.

The inverse is

$$\mathbf{L}_i^{-1}(z^{-1}) \cdot z^{-1},$$

The last piece for the design is a diagonal coefficient matrix $\mathbf{D}$

$$\mathbf{D} = \mathbf{diag}(d_0, \ldots, d_{M-1}).$$

With these basic matrices, the following product or decomposition is a general form for $\mathbf{F_a}(z)$ [10], which includes low delay filter banks and conventional filter banks like the Modified Discrete Cosine Transform (MDCT), also known as TDAC filter bank or lapped orthogonal transform.

$$\mathbf{F_a}(z) = \prod_{j=\nu-1}^{0} \mathbf{L}_{\mu+j}(z) \cdot \prod_{i=\mu-1}^{0} \left(\mathbf{L}_i(z^{-1}) \cdot z^{-1}\right) \cdot \mathbf{D}. \quad (9)$$

where ν is the number of zero delay matrices and μ is the number of maximum delay matrices. The indices below the product signs mean that the product is counted backwards. The coefficients for the zero delay matrices, the maximum delay matrices, and the diagonal matrix, are obtained using numerical optimization. The resulting structure or implementation for a low delay filter bank with $\nu = 2$ and $\mu = 0$ can be seen in Fig. 3. The inverse for the synthesis, with a suitable delay for causality, is

$$\mathbf{F_s} = \mathbf{F_a}^{-1}(z) = \mathbf{D}^{-1} \cdot \prod_{i=0}^{\mu-1} \left(\mathbf{L}_i^{-1}(z^{-1}) \cdot z^{-1}\right) \prod_{j=0}^{\nu-1} \mathbf{L}_{\mu+j}^{-1}(z). \quad (10)$$

The MDCT, for instance, results for $\nu = \mu = 1$ and $n_a = n_s = N/2$.

We have now almost completely decomposed the filterbank into lifting steps. Matrices $\mathbf{L}(z)$ consist of lifting steps, $\mathbf{S}(z)$ consists of reordering and advance steps, and the transform matrix $\mathbf{T}$ is usually the Discrete Cosine Transform, for the MDCT it is the DCT type 4, which can also be represented using lifting steps [4, 11, 7]. What remains is the diagonal matrix $\mathbf{D}$. This matrix can also decomposed into lifting steps, if $d_i = 1/d_{N-1-i}$ for $i = 0, \ldots, N/2-1$. Its decomposition can be seen in the following:

$$\begin{bmatrix} -1 & 0 \\ d^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & d^{-1} \end{bmatrix} = \begin{bmatrix} d & 0 \\ 0 & d^{-1} \end{bmatrix}.$$

3.1. Representable class of filter banks

Representing the filter bank with lifting steps only leads to this restriction on the diagonal matrix $\mathbf{D}$. What does this restriction mean? Take a look at the condition for perfect reconstruction:

$$P_i'(z) = \frac{s \cdot z^{-d} P_i(z)}{z^{-2} P_{N+i}(z) P_{2N-1-i}(z) - P_i(z) P_{N-1-i}(z)} \quad (11)$$

$$P_{N+i}'(z) = \frac{s \cdot z^{-d} P_{N+i}(z)}{z^{-2} P_{N+i}(z) P_{2N-1-i}(z) - P_i(z) P_{N-1-i}(z)} \quad (12)$$

where $s = \pm 1$, and d is a delay.

Our analysis and synthesis are identical except for the sign s, $g(n) = s \cdot h(n)$, if $P_i'(z) = s \cdot P_i(z)$ for $i = 0, \ldots, 2N-1$. Eq. 11, 12 show that this is the case if

$$z^{-2} P_{N+i}(z) P_{2N-1-i}(z) - P_i(z) P_{N-1-i}(z) = s \cdot z^{-d} \quad (13)$$

for $i = 0, \dots, N-1$.

The left side of this equation (13) is the determinant of the following 2×2 submatrix of $\mathbf{F_a}(z)$ (5),

$$\begin{bmatrix} z^{-1}P_{2N-1-i}(z^2) & P_{N-1-i}(z^2) \\ P_i(z^2) & z^{-1}P_{N+i} \end{bmatrix}$$

We call its determinant $\det_{2x2}(\mathbf{F_a}(z))$

$$\det_{2x2}(P(z)) = z^{4m} \det_{2x2}(D)$$

To determine this determinant, take a look at the determinants of our decomposition. $\det_{2x2}$ of the zero delay matrices is 1, for the maximum delay matrices it is z^{-4}, for $\mathbf{S}(z)$ it is $-z$. Our restriction from the lifting representation for $\mathbf{D}$, $d_i = 1/d_{N-1-i}$, means exactly that

$$\det_{2x2}(\mathbf{D}) = 1.$$

Because of the bi-diagonal structure of $\mathbf{L}$, the $\det_{2x2}(\mathbf{F_a}(z))$ is the product of the $\det_{2x2}$ of the matrices of the decomposition. We can now conclude that $\det_{2x2}(\mathbf{F_a}(z)) = z^{-4\mu}$, where μ is the number of maximum delay matrices. With eq. (11), (12) and $d = 4\mu$ it follows that

$$g(n) = s \cdot h(n).$$

This means, the restriction of using lifting steps only, to obtain integer to integer filter banks means that we obtain only filter banks where the analysis and synthesis baseband impulse responses are identical except for a sign. Observe that for practical applications this is not a severe restriction. Cosine modulated filter banks with perfect reconstruction can have different window functions or baseband impulse responses $h(n)$, $g(n)$, but for most applications (as in audio coding) it is desirable to have them identical.

4. RESULTS

The performance of the lifting based integer-to-integer low delay filter banks, and the case of the MDCT filter bank can be evaluated based on the approximation error. Figure 4 shows the output of the float MDCT filter bank and the IntMDCT for an example signal. It can be seen that they are very similar, as desired. The difference is the noise floor caused by the rounding of the integer-to-integer filter bank. This noise floor is well below the quantization error caused by a perceptual coder. This means that the integer MDCT is well suited for a bit-rate efficient enhancement layer.

4.1. Concept of a Scalable System

Figure 5 illustrates the concept of the proposed scalable architecture which consists of a conventional perceptual base layer coder and a lossless enhancement coder based on the IntMDCT.

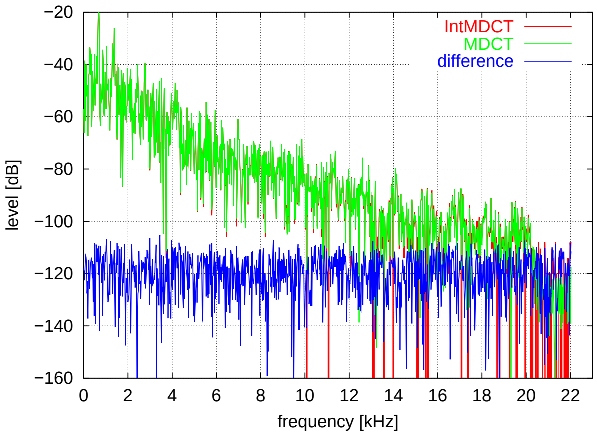

Fig. 4. IntMDCT, MDCT and difference spectra of a usual audio signal (SQAM CD, track 64)

In the proposed encoder the quantized MDCT spectrum is used to predict the IntMDCT spectrum. It is rounded to integer values and only the difference to the IntMDCT values has to be entropy coded. This produces both a lossy (perceptually coded) bitstream and a lossless enhancement bitstream which carries the necessary information to reconstruct the input signal exactly.

In the decoder the quantized MDCT spectrum is reconstructed from the lossy coded bitstream. By applying the inverse MDCT, the perceptually coded audio signal can be obtained. If the enhancement bitstream is decoded, the original IntMDCT spectrum can be obtained. Finally the inverse IntMDCT is applied to obtain the losslessly decoded audio signal.

This scalable system has been implemented based on MPEG-4 AAC using the additional coding tools Window Switching and Mid/Side (MS) Coding. The Window Switching tool allows to increase the temporal resolution for transient signal by reducing the number of MDCT subbands from 1024 to 128. The MS tool is used to exploit redundancy between stereo channels and to avoid binaural unmasking. In the lossless enhancement layer the MS operation is done by applying a lifting-based Givens rotation with angle $\pi/4$ to allow energy conservation. The resulting difference values in the frequency domain are coded using Huffman coding similar to MPEG-4 AAC core coder.

In table 1 bitrate results for different configurations for the entire SQAM CD [12] (including all zero samples) are summarized. In the last column the core coder was disabled to measure the performance of this system in a lossless-only mode. It can be seen that a certain bitrate overhead is introduced by the scalable approach. But this is partially due to

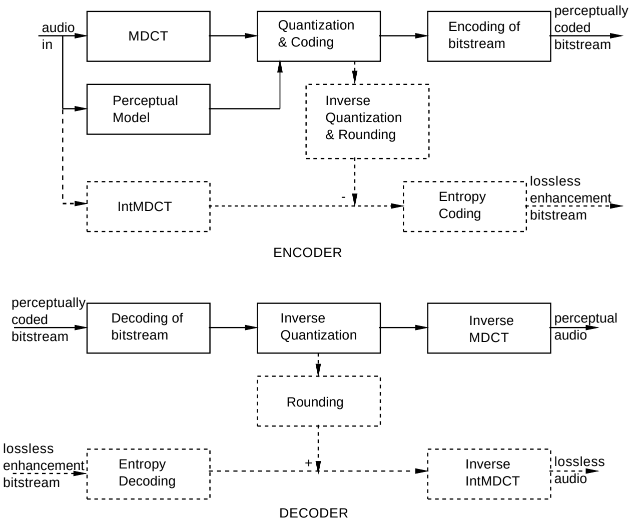

Fig. 5. MDCT-based perceptual audio coding scheme (solid lines) and scalable lossless enhancement (dashed lines)

the fact that the chosen test material contains a lot of zero samples which are coded very inefficiently with the constant rate AAC core coder.

AAC (kbps(stereo))	128	192	0
AAC (bit/sample)	1.5	2.2	0
Enhancement (bit/sample)	4.0	3.7	4.7
Total (bit/sample)	5.5	5.9	4.7

Table 1. Bitrate results for scalable coding system

As a reference for typical prediction-based lossless audio coding schemes, Monkey's Audio [13] was used. For the selected test material this coding scheme achieved an average bitrate of 4.6 bit/sample.

5. CONCLUSIONS

Integer-to-integer cosine modulated filter banks, including low delay filter banks and MDCT filter banks, can be built using the lifting scheme. From the approximate energy conservation we conclude that we can and need to construct them using lifting steps only. The latter leads to the restriction that the analysis and synthesis baseband impulses $h(h)$ and $g(n)$ are identical. For many practical applications, like in audio coding, this is not a severe restriction, but rather a desired property. These integer filter banks can be used to build an efficient lossless enhancement layer of a perceptual coder. We implemented such a system based on the IntMDCT and the MPEG-4 AAC perceptual coder. Our results show that the enhancement layer is indeed quite efficient, for instance in comparison to a state-of-the-art lossless only coder.

6. REFERENCES

[1] "Generic Coding of Moving Pictures and Associated Audio: Advanced Audio Coding," International Standard 13818-7, ISO/IEC Moving Pictures Expert Group, ISO/IEC JTC1/SC29/WG11, 1997.

[2] "Coding of Audio-Visual Objects: Audio," International Standard 14496-3, ISO/IEC Moving Pictures Expert Group, ISO/IEC JTC1/SC29/WG11, 1999.

[3] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," Tech. Rep., Bell Laboratories, Lucent Technologies, 1996.

[4] K. Komatsu and K. Sezaki, "Reversible Discrete Cosine Transform," in *Proc. ICASSP*, 1998, vol. 3, pp. 1769–1772.

[5] T. D. Tran, "The LiftLT: fast lapped transforms via lifting steps," *IEEE Signal Processing Letters*, vol. 7, pp. 145–149, June 2000.

[6] S. Oraintara, Y. Chen, and T. Nguyen, "Integer Fast Fourier Transform (INTFFT)," in *Proc. ICASSP*, 2001.

[7] R. Geiger, T. Sporer, J. Koller, and K. Brandenburg, "Audio Coding based on Integer Transforms," in *111th AES Convention*, New York, 2001.

[8] R. Geiger, J. Herre, J. Koller, and K. Brandenburg, "IntMDCT - A link between perceptual and lossless audio coding," in *Proc. ICASSP 2002*, Orlando, 2002.

[9] T. Krishnan and S. Oraintara, "Fast and lossless implementation of the forward and inverse mdct computation in mpeg audio coding," in *ISCAS 2002*, Scottsdale, Arizona, May 2002.

[10] G. Schuller and T. Karp, "Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case," *IEEE Transactions on Signal Processing*, March 2000.

[11] T. D. Tran, "The BinDCT: fast multiplierless approximation of the DCT," *IEEE Signal Processing Letters*, vol. 7, pp. 141–145, June 2000.

[12] *SQAM (Sound Quality Assessment Material)*, European Broadcasting Union (EBU), Geneva, 1988.

[13] M. T. Ashland, "Monkey's Audio - a fast and powerful lossless audio compressor," http://www.monkeysaudio.com.