Improved integer transforms for lossless audio coding

Conference Paper · December 2003

DOI: 10.1109/ACSSC.2003.1292354 · Source: IEEE Xplore

Gerald Schuller
Technische Universität Ilmenau

IMPROVED INTEGER TRANSFORMS FOR LOSSLESS AUDIO CODING

Ralf Geiger[1], *Yoshikazu Yokotani*[1,2], *Gerald Schuller*[1]

Fraunhofer IIS AEMT, Ilmenau, Germany[1]
University of Texas at Arlington, Electrical Engineering Dept., Arlington, TX, U.S.A.[2],
Email: ggr@emt.iis.fhg.de[1], yoshi@msp.uta.edu[2], shl@emt.iis.fhg.de[1]

ABSTRACT

Lifting scheme based integer transforms are very powerful tools to construct lossless coding schemes. These transforms such as the Integer Fast Fourier Transform (IntFFT) and the Integer Modified Discrete Cosine Transform (IntMDCT) are integer approximations of the original floating-point transforms, and hence there is an approximation error in the transform domain. This paper will propose structures for improved integer transforms in terms of improved approximation accuracy and computational efficiency. Experimental results will show that clear improvements in these two points are achieved in lossless audio coding.

1. INTRODUCTION AND GOAL

Integer transforms are used for lossless coding applications such as in audio [1] and image coding [2]. Integer transforms can map integers to integers and are reversible. Hence, they are a lossless process over the forward and inverse transforms. In addition, they inherit the properties of the original transforms. Thus, in general, a lossless codec can be realized by simply cascading the integer transform with an entropy coding scheme.

Integer transforms can be obtained by using a structure of lifting steps with constant lifting coefficients and rounding after coefficient multiplication, as seen in Fig. 2. The lifting scheme [3] has been applied to construct a number of integer transforms such as the Integer Fast Fourier Transform (IntFFT) [4], the Integer Discrete Cosine Transform (IntDCT) [2], and the Integer Modified Discrete Cosine Transform (IntMDCT) [1, 5].

The main application for the IntMDCT is the area of lossless audio coding. In [6] and [7], scalable lossless enhancements of MDCT-based perceptual lossless audio codecs are proposed. Figure 1 shows the basic structure of such a codec.

Here the IntMDCT is useful because the MDCT is the mainly used transform or filter bank in perceptual audio coders, like MPEG-1 Layer 3 (MP3), or MPEG-4 AAC, its successor. Currently MPEG is working on an extension towards lossless coding, and IntMDCT based solutions are proposed. Here the obtained compression ratio is especially important and previously used IntMDCT structures have come to a limit. This shows that a new structure is needed.

Integer transforms have two measures of quality for coding. The first is the approximation accuracy. Since integer transforms have integer values in the frequency domain, they can only be approximations of the original floating point transforms. The approximation error results from an accumulation of the rounding errors of the lifting steps. The coding efficiency, and hence the compression ratio, is limited by this approximation error, because especially at higher frequencies, where the audio signal usually has very low energy, it shows up as "noise floor" which has to be coded.

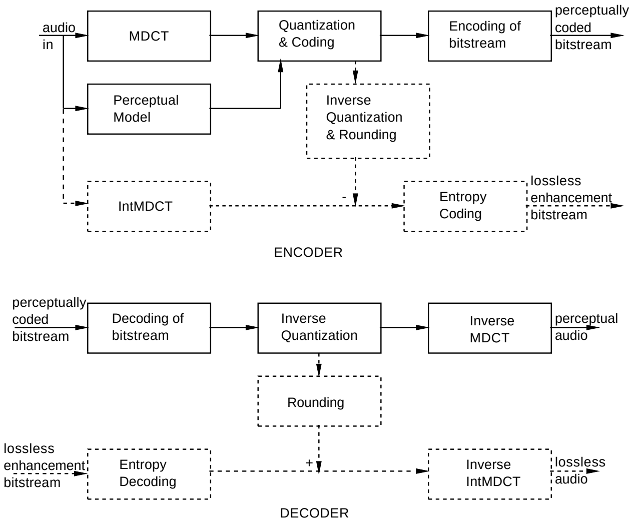

Figure 1: Perceptual audio coding scheme (solid lines) and scalable lossless enhancement (dashed lines)

The second measure is the computational complexity. This is important because in audio coding large size transforms are necessary, and in general the complexity of integer transforms is higher than for the original floating point transforms.

2. PRESENT STATE

Transforms like the DFT, DCT, or the MDCT can be formulated in terms of so-called Givens rotations. The lifting scheme can be applied to get an invertible integer approximation of each Givens rotation:

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

The integer approximation is achieved by applying a rounding function after each addition. This 3 step lifting structure is illustrated in Fig. 2.

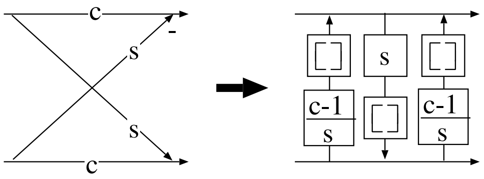

Figure 2: 3 step lifting scheme to implement an integer approximation of a Givens rotation. [ ] symbolize rounding.

Transforms like the FFT contain many trivial rotations with only factors +1/-1. For integer transforms these trivial rotations have to be modified because they imply a scaling of the signal, resulting in increase of energy. Hence these simple rotations are implemented using the 3 step lifting structure with non-trivial coefficients. The consequence is that the computational complexity of the integer transforms is higher than for the original floating point transform. An important factor for the approximation error of the 3 step lifting scheme is the total number of lifting steps needed for the integer transform.

For the IntMDCT, the lifting scheme is first applied to the steps of windowing and time-domain aliasing. This concept can also be generalized and applied to low-delay filter banks, as shown in [8]. The remaining block transform is the DCT type 4 (DCT$_{\mathrm{IV}}$). Figure 3 illustrates this decomposition.

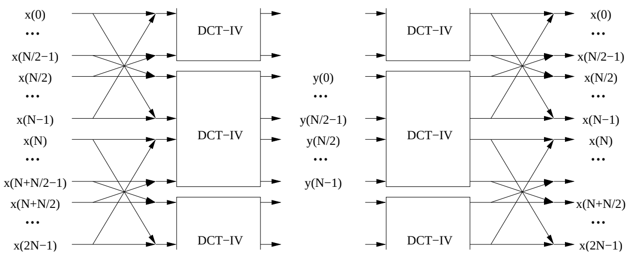

Figure 3: Decomposition of MDCT and inverse MDCT into Givens rotations and DCT$_{\mathrm{IV}}$

The DCT$_{\mathrm{IV}}$ can be implemented using the FFT, see [9]. For an integer approximation, the lifting scheme is applied in all the stages. With this connection between the MDCT and the FFT, an improvement of the IntFFT also improves the IntMDCT.

To improve the IntMDCT according to the two quality measures we will describe two techniques. The first one, providing refinements of the lifting scheme, can improve the computational efficiency, and to some extent the approximation accuracy as well. The second approach, the multi-dimensional lifting, is more focused on improved approximation accuracy.

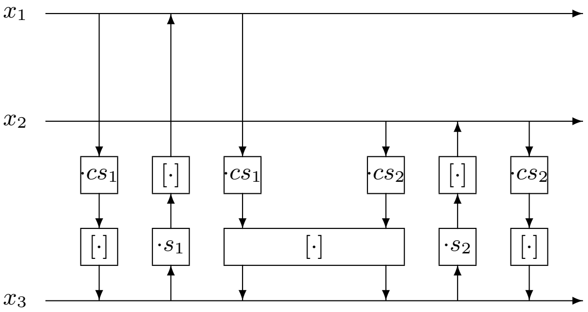

Figure 4: Combined rounding in two succeeding lifting steps

3. REFINEMENTS OF THE LIFTING SCHEME

3.1. Combined Rounding

Whenever two succeeding lifting steps occur in a way that two rounded values are added to the same third value, subsequently, the rounding operations can be combined. This can reduce the overall rounding error. Figure 4 illustrates this procedure.

3.2. 2 Step Lifting

The technique described in the following is based on the combination of the conventional 3 step lifting scheme and 2 step lifting scheme [2]. When the IntFFT is used to implement the IntMDCT [10], this technique is useful, since, as mentioned earlier, the FFT has trivial rotations and each consecutive two of them can be implemented with 4 lifting steps with only 2 rounding operations as depicted in Fig. 5. Because of reduction of the rounding operations, the approximation error due to the trivial rotations can be reduced. Also, it can be seen that the lifting coefficients simply become 0.5 and the multiplication with the coefficient is done by a shift operation. As a result, this contributes to a significant improvement of computational efficiency. Moreover, it is possible to combine two rounding operations in some part of the implementation so that the approximation error could be further reduced. Here, it should be noted that the 3 step lifting scheme is applied to all of the other Givens rotations.

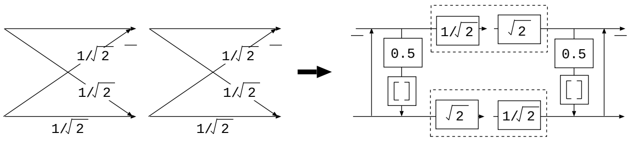

Figure 5: 2 step lifting scheme to implement 2 successive trivial butterflies.

4. MULTI-DIMENSIONAL LIFTING

Apart from Givens roatations, the lifting scheme can also be used for an invertible integer approximation of certain scaling operations. In [11] the following lifting decomposition of a 2 × 2 scaling matrix with determinant 1 is presented:

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

This decomposition provides the basic idea for the new approach. The equation still holds when all the values are replaced by $n \times n$ matrices. So, for any invertible $n \times n$ matrix T and for the $n \times n$ identity matrix I_n the following decomposition of $2n \times 2n$ block matrices is possible:

$$\begin{pmatrix} T & 0 \\ 0 & T^{-1} \end{pmatrix} = \begin{pmatrix} -I_n & 0 \\ T^{-1} & I_n \end{pmatrix} \begin{pmatrix} I_n & -T \\ 0 & I_n \end{pmatrix} \begin{pmatrix} 0 & I_n \\ I_n & T^{-1} \end{pmatrix} \quad (1)$$

Apart from some simple operations like permutations or multiplication with -1 all the three blocks of this decomposition have the following general structure:

$$\begin{pmatrix} I_n & 0 \\ A & I_n \end{pmatrix}$$

with an $n \times n$ matrix A.

For this $2n \times 2n$ block matrix a generalized lifting scheme can be applied, called "multi-dimensional lifting" in the following. Similar to the conventional lifting scheme, these $2n \times 2n$ matrices can be used for invertible integer approximations of the transform T in the following way: The first half of the integer input values is processed by the matrix A and then rounded to integer values before adding them to the second half of the values.

The inverse of the block matrix is given by

$$\begin{pmatrix} I_n & 0 \\ A & I_n \end{pmatrix}^{-1} = \begin{pmatrix} I_n & 0 \\ -A & I_n \end{pmatrix}$$

So this process can be inverted without any error by simply applying the same matrix A and the same rounding, and subtracting the resulting values instead of adding them. As the first half of the values is not modified in the forward step, they are still available for the inverse step. No special restrictions apply to the matrix A, e.g. it does not necessarily have to be invertible.

4.1. The Stereo IntMDCT

The most straight-forward way of using the multi-dimensional lifting approach for the IntMDCT is to apply the $\mathrm{DCT_{IV}}$ to two blocks of signals simultaneously. These blocks can either be from two succeeding blocks or from the left and the right channel of a stereo audio signal. The decomposition in equation (1) is applied to the $\mathrm{DCT_{IV}}$ matrix. Since the inverse of the $\mathrm{DCT_{IV}}$ is again the $\mathrm{DCT_{IV}}$, the decomposition in equation (1) becomes:

$$\begin{pmatrix} \mathrm{DCT_{IV}} & 0 \\ 0 & \mathrm{DCT_{IV}} \end{pmatrix} =$$

$$\begin{pmatrix} -I_N & 0 \\ \mathrm{DCT_{IV}} & I_N \end{pmatrix} \begin{pmatrix} I_N & -\mathrm{DCT_{IV}} \\ 0 & I_N \end{pmatrix} \begin{pmatrix} 0 & I_N \\ I_N & \mathrm{DCT_{IV}} \end{pmatrix} \quad (2)$$

So, apart from permutations and multiplications with -1, the application of the $\mathrm{DCT_{IV}}$ to two blocks of signals can be performed with three multi-dimensional lifting steps. This process is illustrated in figure 6, including the rounding operations for the integer approximation.

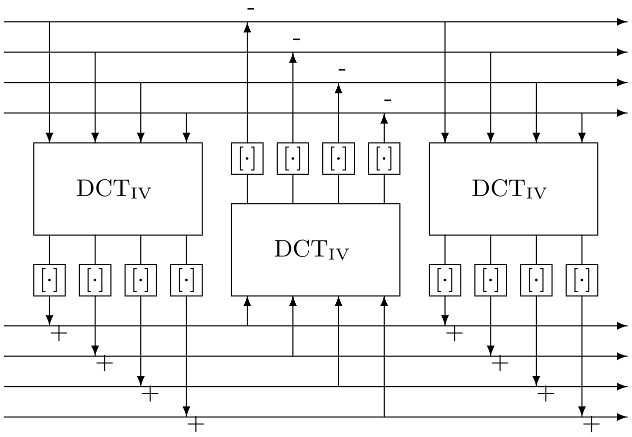

Figure 6: Invertible integer approximation of two blocks of $\mathrm{DCT_{IV}}$ by three multi-dimensional lifting steps

With this approach two $\mathrm{DCT_{IV}}$ transforms of length N can be implemented in an invertible integer fashion with only $3N$ rounding steps, i.e. $3N/2$ rounding steps per transform.

The $\mathrm{DCT_{IV}}$ in the three multi-dimensional lifting steps can have an arbitrary implementation, e.g. floating-point or fixed-point based. It does not need to be invertible. It just has to be performed in the same way in the forward and inverse IntMDCT. This makes this approach also suitable for high transform lengths of e.g. 1024, as used in audio coding applications. The overall computational complexity is about 1.5 times the computational complexity of the non-integer implementation of the two $\mathrm{DCT_{IV}}$ transforms. This is still lower than for the conventional lifting-based integer implementations, which are about twice as complex as the conventional $\mathrm{DCT_{IV}}$, as these implementations have to implement the trivial +/− butterflies based on lifting to achieve an energy conservation, see [12].

4.2. The Mono IntMDCT

The approach presented so far always needs to calculate two $\mathrm{DCT_{IV}}$ transforms simultaneously. This can e.g. be achieved by calculating the $\mathrm{DCT_{IV}}$ of two succeeding blocks of the audio signal. In the case of a two-channel stereo signal this can also be achieved by calculating the $\mathrm{DCT_{IV}}$ of the left and the right channel simultaneously. The first version introduces an additional delay of one block into the system, the second version is only possible for stereo signals.

If both the delay and the stereo version is not desired, this approach is still possible, but some additional stages of Givens rotations are necessary.

The $\mathrm{DCT_{IV}}$ of length N

$$\mathrm{DCT_{IV}^{(N)}} = \left(\sqrt{\frac{2}{N}} \cos \frac{(2k+1)(2l+1)\pi}{4N} \right)_{k,l=0,\ldots,N-1}$$

can be decomposed into two $\mathrm{DCT_{IV}}$ of length $N/2$ and certain pre- and post-modulation stages. In the following this decomposition is described.

Define the $N \times N$ matrices L and M by

$$\begin{pmatrix} L_{k,k} & L_{k,N-1-k} \\ L_{N-1-k,k} & L_{N-1-k,N-1-k} \end{pmatrix} =$$

$$\begin{pmatrix} \cos(\frac{2k+1}{4N}\pi) & -\sin(\frac{2k+1}{4N}\pi) \\ -\sin(\frac{2k+1}{4N}\pi) & -\cos(\frac{2k+1}{4N}\pi) \end{pmatrix} \quad k = 0, \ldots, N/2-1$$

$$L_{k,l} = 0 \qquad \text{else}$$

$$M = \frac{1}{\sqrt{2}} \begin{pmatrix} I_{N/2} & I_{N/2} \\ -I_{N/2} & I_{N/2} \end{pmatrix}$$

and the $N \times N$ permutation matrices P and Q by

$$P_{4k,4k} = P_{4k+1,4k+1} = P_{4k+2,4k+3} = P_{4k+3,4k+2} = 1$$
$$k = 0, \ldots, N/4-1$$
$$P_{k,l} = 0 \quad \text{else}$$

i.e. every second pair of values is swapped, and

$$Q_{k,2k} = Q_{N/2+k,2k+1} = 1 \quad k = 0, \ldots, N/2-1$$
$$Q_{k,l} = 0 \qquad \text{else}$$

i.e. the even indices are arranged in the first half, the odd indices are arranged in the second half.

With these matrices the $\mathrm{DCT_{IV}}$ of length N can be decomposed into

$$\mathrm{DCT}_{\mathrm{IV}}^{(N)} = L \begin{pmatrix} \mathrm{DCT}_{\mathrm{IV}}^{(N/2)} & 0 \\ 0 & \mathrm{DCT}_{\mathrm{IV}}^{(N/2)} \end{pmatrix} MQP$$

Figure 7 illustrates this decomposition.

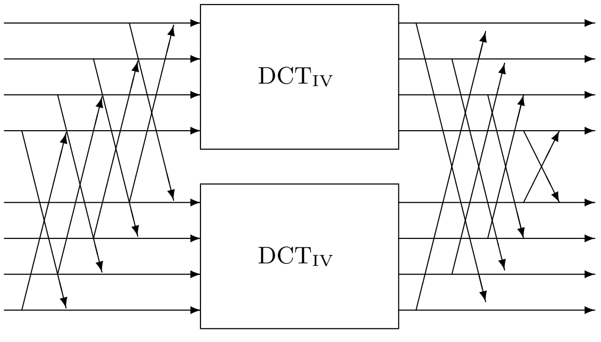

Figure 7: $\mathrm{DCT_{IV}}$ of length N by two $\mathrm{DCT_{IV}}$ of length $N/2$ and two stages of Givens rotations

Now the two $\mathrm{DCT_{IV}}$ of length $N/2$ can be decomposed into three multi-dimensional lifting steps of length $N/2$ using equation (2). The matrices L and M can both be considered as $N/2$ Givens rotations. The lifting implementation of the matrices L and M can be adopted to the $\mathrm{DCT_{IV}}$ stage to further reduce the overall number of rounding operations. The matrix M can be implemented using the multi-dimensional lifting steps

$$\begin{pmatrix} I_{N/2} & 0 \\ -\frac{1}{2} I_{N/2} & I_{N/2} \end{pmatrix} \begin{pmatrix} I_{N/2} & I_{N/2} \\ 0 & I_{N/2} \end{pmatrix}$$

The remaining scaling factors $\sqrt{2}$ and $1/\sqrt{2}$ can be handled by the $\mathrm{DCT_{IV}}$ stage. Furthermore, the remaining $N/2$ rounding operations for M and $N/2$ of the $3N/2$ rounding operations for L can be combined with the rounding operations in the $\mathrm{DCT_{IV}}$ stage, as described in the previous section. So, overall only $5N/2$ rounding operations are necessary for this invertible integer approximation of the $\mathrm{DCT_{IV}}$ of length N.

Including the windowing stage, the total number of rounding operations for this IntMDCT is $4N$, i.e. 4 rounding operations per sample.

5. RESULTS

The approximation accuracy of the combined 2-step lifting based and the multi-dimensional lifting based IntMDCT are evaluated by applying the transforms to the audio material used for the lossless coding activities of the ISO MPEG group [13]. The audio material consists of recordings of the New York Symphonic Ensemble and Jazz recordings. The evaluation is done for 48 kHz / 16 bit and 96 kHz / 24 bit. The performance of the different transforms is evaluated by mean squared error (MSE) maximum error and an entropy estimation, calculated by $\sum_k \log_2(2|y_k|+1)$. The number of instructions reflects the number of additions and multiplications.

The size of the IntMDCT is set to be 1024. Table 2 shows the results for the conventional lifting based IntMDCT, the combined 2-step lifting based, the multi-dimensional lifting based IntMDCT in the mono case, and, as a reference, the rounded MDCT, which does not allow lossless operation. The resulting MSE and maximum error values are similar for both input format, so only the overall values are displayed.

It can be observed that the computational efficiency is significantly improved by the combined 2-step lifting approach and to some extent the entropy of the resulting IntMDCT coefficients is reduced. More importantly, the approximation error is largely reduced by the multi-dimensional lifting approach, and the estimated entropy comes close to the theoretical limit given by the rounded MDCT.

The multi-dimensional lifting based IntMDCT was also evaluated in a codec for scalable lossless enhancement of MPEG-4 AAC, see [7] for a detailed structure of this codec. Table 2 summarizes the compression results in bits per sample for the AAC-based lossless enhancement, the lossless-only mode, and, as a comparison, the prediction-based lossless coder Monkey's Audio [14]. It can be observed that the compression performance in the lossless-only mode comes very close to the performance of Monkey's Audio. Additionally the codec provides a scalable mode, where the enhancement can clearly benefit from the perceptual core codec, compared to a simulcast solution.

6. CONCLUSIONS

In this paper we have presented two possible improvements for lifting based integer transforms, especially focusing lossless audio coding applications. The first approach of combined 2-step lifting allows for a low computational complexity of the integer transform, and to some extent also im-

	Lifting	Combined	Multi-dim.	Rounded
	based	2-step lifting	lifting	MDCT
	IntMDCT	IntMDCT	IntMDCT	(not lossless)
Rounding operations per sample	22.5	8.65	4	1
Instructions per sample	45	27	32	20
MSE	1.97	1.49	0.48	0
max. Error	8	10	4	0
Entropy estimation 48 kHz 16 bit	$1.180 \cdot 10^8$	$1.177 \cdot 10^8$	$1.166 \cdot 10^8$	$1.160 \cdot 10^8$
Entropy estimation 96 kHz 24 bit	$4.145 \cdot 10^8$	$4.143 \cdot 10^8$	$4.125 \cdot 10^8$	$4.113 \cdot 10^8$

Table 1: Comparison of convention lifting based IntMDCT, combined 2-step lifting based IntMDCT, multi-dimension lifting (MDL) based IntMDCT and rounded MDCT (not lossless)

	48 kHz	48 kHz	96 kHz	192 kHz
	16 bit	24 bit	24 bit	24 bit
AAC	1.3	1.3	0.8	0.5
Enhancement	6.5	14.4	11.0	9.2
AAC + Enhancement	7.8	15.7	11.8	9.7
Lossless-only	7.5	15.3	11.6	9.5
Monkey's Audio 3.97	7.2	15.2	11.5	9.4
Simulcast (AAC + Monkey's Audio)	8.5	16.5	12.3	9.9

Table 2: Compression results (in bits per sample) for AAC-based lossless enhancement, lossless-only mode, Monkey's Audio, and a simulcast solution

proves the approximation accuracy. The second approach of multi-dimensional lifting achieves a very high approximation accuracy, especially for long transforms, typical for audio coding applications. It is also evaluated in the context of a scalable lossless enhancement of MPEG-4 AAC, providing a flexible perceptual and lossless codec with high compression efficiency.

7. REFERENCES

[1] R. Geiger, T. Sporer, J. Koller, and K. Brandenburg. Audio coding based on integer transform. *111st AES convention Preprint 5471*, 2001.

[2] J. Liang and T. D. Tran. Fast multiplierless approximations of the DCT with the lifting scheme. *IEEE Trans. on Signal Processing*, 49(12):3032–3044, December 2001.

[3] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[4] S. Oraintara, Y. J. Chen, and T. Nguyen. Integer fast fourier transform. *IEEE Trans. on Signal Processing*, 50:607–618, March 2002.

[5] T. Krishnan and S. Oraintara. A fast and lossless forward and inverse structure for the MDCT in MPEG audio coding. *Proc. of the International Symposium on Circuits and Systems*, May 2002.

[6] R. Geiger, J.Herre, J. Koller, and K. Brandenburg. IntMDCT - A link between perceptual and lossless audio coding. *Proc. IEEE International Conf. on Acoustics, Speech, and Signal Processing*, 2:1813–1816, 2002.

[7] R. Geiger, G. Schuller, J. Herre, R. Sperschneider, and T. Sporer. Scalable Perceptual and Lossless Audio Coding based on MPEG-4 AAC. In *AES 115th Convention*, New York, October 2003.

[8] R. Geiger and G. Schuller. Integer low delay and MDCT filter banks. *Proc. of the Asilomar Conf. on Signals, Systems, and Computers*, 2002.

[9] H. S. Malvar. *Signal Processing with Lapped Transforms.* Archtech House, 1992.

[10] Y. Yokotani and S. Oraintara. Lossless audio compression using integer modified discrete cosine transform. In *Proc. ISPACS 2003*, Japan, December 2003.

[11] R. Geiger and G. Schuller. Integer low delay and MDCT filter banks. *Proc. of the Asilomar Conf. on Signals, Systems, and Computers*, 2002.

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

[13] ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group. Final Call for Proposals on MPEG-4 Lossless Audio Coding. (N5208), October 2002.

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