An Efficient Implementation of the Forward and Inverse MDCT in MPEG Audio Coding

Vladimir Britanak and K. R. Rao

***Abstract*—The modified discrete cosine transform (MDCT) is employed in subband/transform coding schemes as the analysis/synthesis filter bank based on time domain aliasing cancellation (TDAC). The most efficient implementation of the forward and inverse MDCT computation for layer III in MPEG-1 and MPEG-2 international audio coding standards is proposed. It is based on a new fast algorithm for the forward and inverse MDCT computation in the oddly stacked system. The complete signal flow graphs for the implementation of MDCT and inverse MDCT in layer III are also provided.**

***Index Terms*—Audio coding, modified discrete cosine transform (MDCT), MPEG.**

I. Introduction

THE MODIFIED discrete cosine transform (MDCT) is employed in subband/transform coding schemes as the analysis/synthesis filter bank based on time domain aliasing cancellation (TDAC). Therefore, it is frequently called the "TDAC transform" [1], [2]. The MDCT is the basic processing component for high quality audio compression in the international audio coding standards [7]–[10] and commercial audio coding products [11], [12].

Generally, in almost all existing audio coding systems, the fast MDCT computation is realized by the complex-valued or real-valued FFT algorithms. The most efficient FFT-based algorithm for the fast MDCT computation in terms of arithmetic complexity has been developed by Duhamel *et al.* [3] and it has been adopted in the current audio coding systems. Since this algorithm has been formulated for data sequences with lengths $N = 2^n$, it does not offer the possibility to derive an efficient MDCT computation in layer III of MPEG-1 [7], [9] and MPEG-2 [8], [9] audio coding standards, where the lengths of data blocks are $N \neq 2^n$. Layer III specifies two different MDCT block sizes: a long block ($N = 36$) and a short block ($N = 12$). The original ISO source code implements the MDCT as-is. This requires for $N = 36$, 648 multiplications and 630 additions and for $N = 12$, 72 multiplications and 66 additions.

Recently, a new fast algorithm for the forward and inverse MDCT computation in the oddly stacked system has been proposed. Detailed description of the algorithm can be found in [4]. It is based on the DCT-II/DST-II [6] and their inverses, DCT-III/DST-III, and uses real arithmetic only. The generalized signal flow graph is regular, structurally simple and enables to compute MDCT and its inverse in general for any N divisible by 4 (see Figs. 1 and 2). This fact takes advantage of the new algorithm in some real-time implementations of the MDCT computation, where the length of a data block is not exactly a power of two. In the specific case of $N = 2^n$, $n > 2$, the algorithm requires $(N/4)(n+1)$ multiplications and $(N/4)(3n+3)$ additions (IMDCT computation requires $(N/4)(3n+1)$ additions only).

Manuscript received June 26, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y. Stylianou.

V. Britanak is with the Institute of Control Theory and Robotics, Slovak Academy of Sciences, 842 37 Bratislava, Slovak Republic (e-mail: utrrbrit@savba.sk).

K. R. Rao is with the Department of Electrical Engineering, University of Texas, Arlington, TX 76019-0016 USA (e-mail: krrao@exchange.uta.edu).

Publisher Item Identifier S 1070-9908(01)00285-1.

In this paper, the most efficient implementation of the MDCT and inverse MDCT computation for layer III in MPEG-1 and MPEG-2 international audio coding standards is proposed. It is based on the new fast algorithm for the MDCT computation in the oddly stacked system [4]. The complete signal flow graphs for the implementation of MDCT and inverse MDCT in layer III are also provided.

II. Fast MDCT/IMDCT Computation in MPEG-1 and MPEG-2 Audio

Let $\{x_n\}$, $n = 0, 1, \cdots, N-1$ represents a windowed input data sequence. The MDCT and inverse MDCT (IMDCT) in oddly stacked system are respectively defined as [1], [2]

$$C_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi\left(2n+1+\dfrac{N}{2}\right)(2k+1)}{2N}\right]$$

$$k = 0, 1, \cdots, \frac{N}{2}-1 \tag{1}$$

$$\hat{x}_n = \frac{2}{N}\sum_{k=0}^{(N/2)-1} C_k \cos\left[\frac{\pi\left(2n+1+\dfrac{N}{2}\right)(2k+1)}{2N}\right]$$

$$n = 0, 1, \cdots, N-1. \tag{2}$$

The MDCT sequence (1) possesses the even antisymmetry property given by

$$C_{N-k-1} = -C_k, \qquad k = 0, 1, \cdots, \frac{N}{2}-1 \tag{3}$$

and hence, only $N/2$ coefficients are linearly independent, resulting in $N/2$ unique MDCT coefficients. The notation $\hat{x}_n$ in (2) emphasizes fact that the recovered data sequence by the inverse transforms does not correspond to the original data sequence. In the context of TDAC analysis/synthesis filter banks, the distorted sequence $\hat{x}_n$ is said to be time-domain aliased [1], [2].

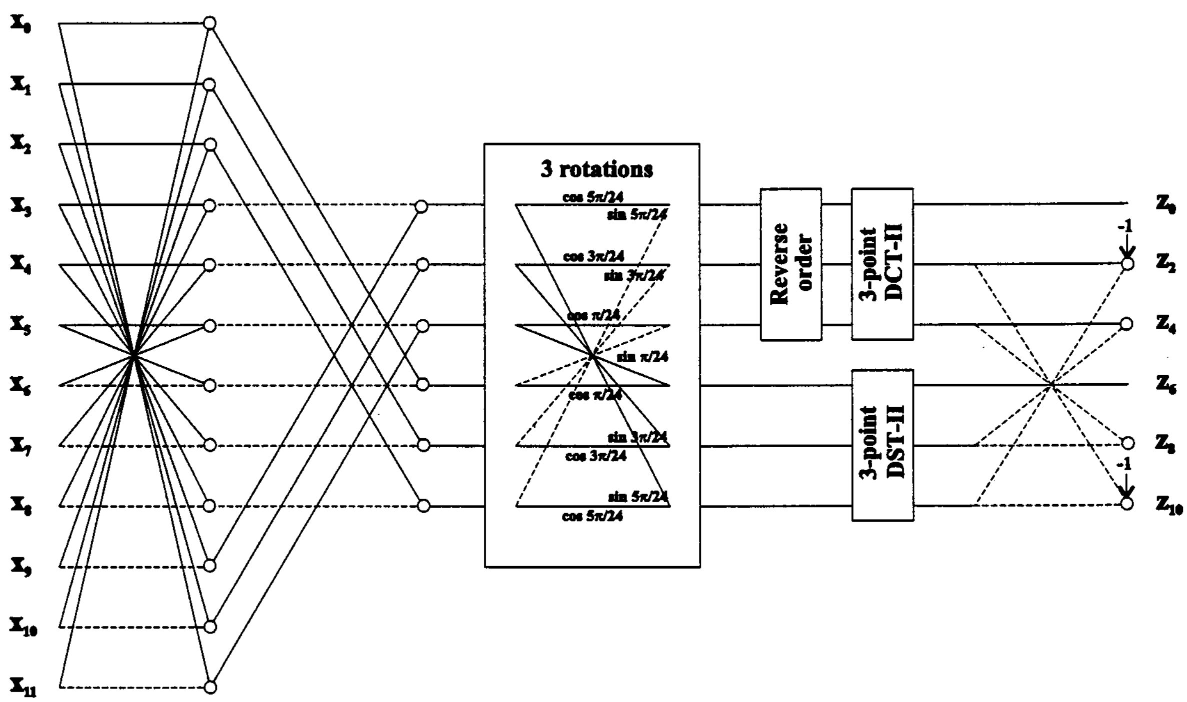

Fig. 1. Signal flow graph for the MDCT/IMDCT computation for $N = 12$.

Because both block sizes in layer III of MPEG-1 and MPEG-2 are divisible by 4, the new fast algorithm for oddly stacked system [4] can be adopted for the efficient MDCT/IMDCT computation in the MPEG family. The corresponding signal flow graphs for the MDCT/IMDCT computation are shown in Figs. 1 and 2 for $N = 12$ and $N = 36$, respectively. Full lines represent transfer factor $+\,1$, while broken lines represent transfer factor $-\,1$ and $\circ$ represents addition. It can be seen that for the fast MDCT computation in layer III we need only efficient 3-point and 9-point DCT-II and DST-II modules. Heideman [5] derived practical and efficient short odd-length DCT-II prime factor modules from corresponding DFT modules. Using a simple relationship between the DCT-II and DST-II [6], the DST-II modules can be easily obtained by simple modification of corresponding DCT-II modules. For the DST-II computation, odd-indexed samples are sign changed, and after the DCT-II computation, the DST-II sequence is in reverse order as is required by the new fast MDCT algorithm.

In the following, the efficient 3-point and 9-point prime factor modules for the forward DCT/DST (DCT-II/DST-II) and inverse DCT/DST (DCT-III/DST-III) computation are presented. In order to unify all modules (forward and inverse), one table of constants is specified and Heideman's forward 3-point DCT module has been modified. Inverse DCT/DST 3-point and 9-point prime factor modules are originally derived.

Table of Constants for the Efficient 3-Point and 9-Point DCT/DST Prime Factor Modules:

$$\begin{array}{llll} u = \dfrac{2\pi}{9} & & & \\ d_1 = \dfrac{\sqrt{3}}{2} & d_2 = 0.5 & d_3 = \cos 4u & d_4 = \cos 2u \\ d_5 = \cos u & d_6 = \sin 4u & d_7 = \sin 2u & d_8 = \sin u \end{array}.$$

Forward 3-Point DCT/DST(DCT-II/DST-II) Module—2 Mults and 4 Adds: The input data sequence is $\{x_0, x_1, x_2\}$, and the output data sequence is $\{C_0^{\rm II}, C_1^{\rm II}, C_2^{\rm II}\}$ or $\{S_2^{\rm II}, S_1^{\rm II}, S_0^{\rm II}\}$

For DST-II:

$$x_1 = -x_1 .$$

For DCT-II:

$$\begin{array}{ll} a_1 = x_0 + x_2 & a_2 = x_0 - x_2 \\ m_1 = d_1\, a_2 & m_2 = d_2\, a_1 \end{array}$$

$$\begin{array}{l} C_0^{\rm II} = S_2^{\rm II} = x_1 + a_1 \\ C_1^{\rm II} = S_1^{\rm II} = m_1 \\ C_2^{\rm II} = S_0^{\rm II} = m_2 - x_1 \end{array}.$$

Inverse 3-Point DCT/DST (DCT-III/DST-III) Module—2 Mults and 4 Adds: The input data sequence is $\{C_0^{\rm II}, C_1^{\rm II}, C_2^{\rm II}\}$ or $\{S_2^{\rm II}, S_1^{\rm II}, S_0^{\rm II}\}$, and the output data sequence is $\{x_0, x_1, x_2\}$

For DCT-III:

$$\begin{array}{ll} m_1 = d_1\, C_1^{\rm II} & m_2 = d_2\, C_2^{\rm II} \\ a_1 = C_0^{\rm II} + m_2 & a_2 = C_0^{\rm II} - C_2^{\rm II} \end{array}$$

$$\begin{array}{ll} x_0 = a_1 + m_1 & x_1 = a_2 \\ x_2 = a_1 - m_1 & \end{array}.$$

For DST-III:

.

$$x_1 = -x_1 .$$

Forward 9-Point DCT/DST (DCT-II/DST-II) Module—10 Mults and 34 Adds: The input data

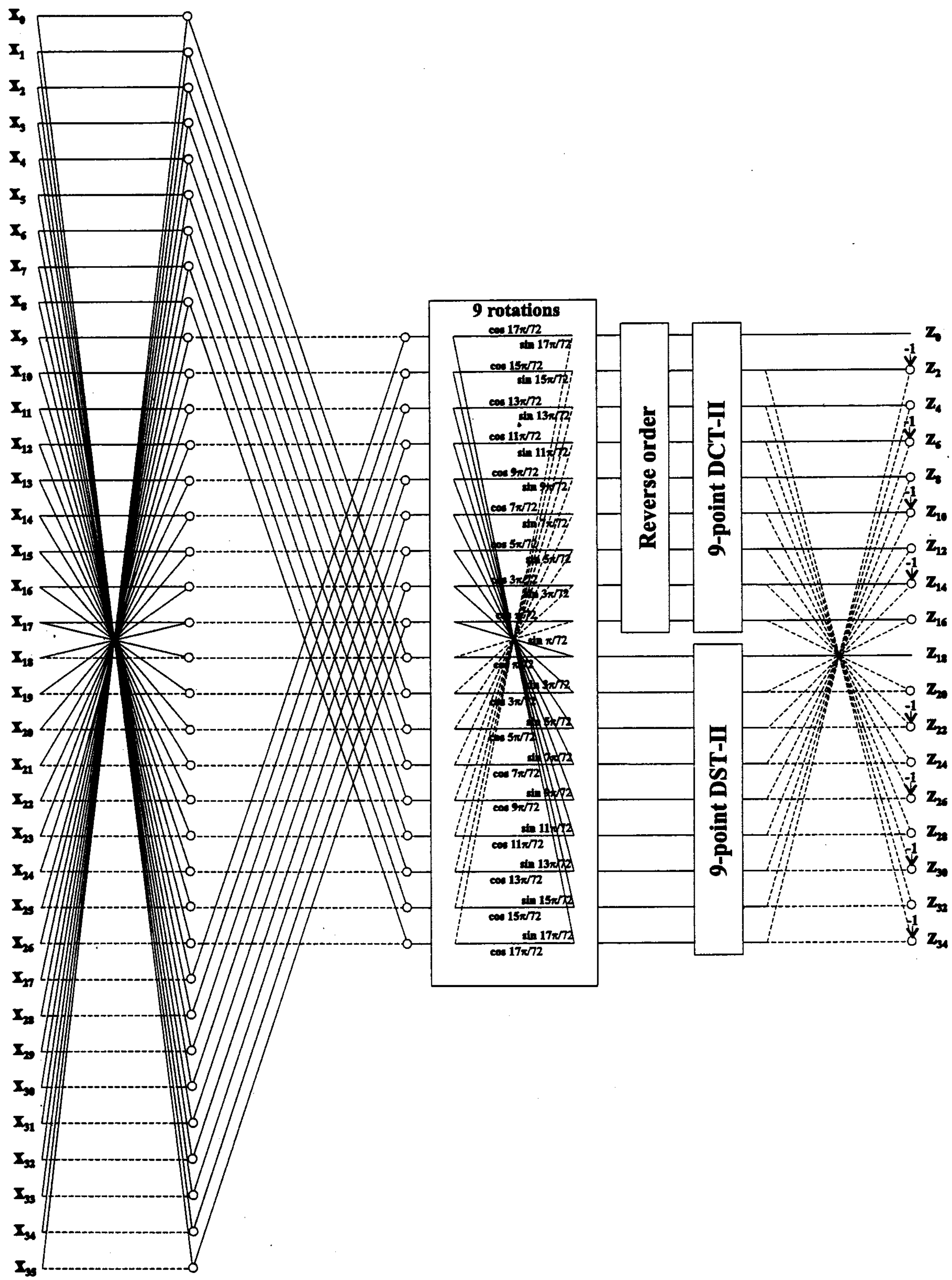

Fig. 2. Signal flow graph for the MDCT/IMDCT computation for $N = 36$.

sequence is $\{x_0, x_1, \cdots, x_8\}$ and output data sequence is $\{C_0^{\rm II}, C_1^{\rm II}, \cdots, C_8^{\rm II}\}$ or $\{S_8^{\rm II}, S_7^{\rm II}, \cdots, S_0^{\rm II}\}$

For DST-II:

$$x_1 = -x_1 \quad x_3 = -x_3 \quad x_5 = -x_5 \quad x_7 = -x_7\,.$$

For DCT-II:

$$\begin{array}{lll} a_1 = x_3 + x_5 & a_2 = x_3 - x_5 & a_3 = x_6 + x_2 \\ a_4 = x_6 - x_2 & a_5 = x_1 + x_7 & a_6 = x_1 - x_7 \\ a_7 = x_8 + x_0 & a_8 = x_8 - x_0 & a_9 = x_4 + a_5 \\ a_{10} = a_1 + a_3 & a_{11} = a_{10} + a_7 & a_{12} = a_3 - a_7 \\ a_{13} = a_1 - a_7 & a_{14} = a_1 - a_3 & a_{15} = a_2 - a_4 \\ a_{16} = a_{15} + a_8 & a_{17} = a_4 + a_8 & a_{18} = a_2 - a_8 \\ a_{19} = a_2 + a_4 & & \end{array}$$

$$
\begin{array}{lll}
m_1 = -d_1\, a_6 & m_2 = d_2\, a_5 & m_3 = d_2\, a_{11} \\
m_4 = -d_3\, a_{12} & m_5 = -d_4\, a_{13} & m_6 = -d_5\, a_{14} \\
m_7 = -d_1\, a_{16} & m_8 = -d_6\, a_{17} & m_9 = -d_7\, a_{18} \\
m_{10} = -d_8\, a_{19} & &
\end{array}
$$

$$
\begin{array}{lll}
a_{20} = x_4 - m_2 & a_{21} = a_{20} + m_4 & a_{22} = a_{20} - m_4 \\
a_{23} = a_{20} + m_5 & a_{24} = m_1 + m_8 & a_{25} = m_1 - m_8 \\
a_{26} = m_1 + m_9 & &
\end{array}
$$

$$
\begin{array}{ll}
C_0^{\mathrm{II}} = S_8^{\mathrm{II}} = a_9 + a_{11} & C_1^{\mathrm{II}} = S_7^{\mathrm{II}} = m_{10} - a_{26} \\
C_2^{\mathrm{II}} = S_6^{\mathrm{II}} = m_6 - a_{21} & C_3^{\mathrm{II}} = S_5^{\mathrm{II}} = m_7 \\
C_4^{\mathrm{II}} = S_4^{\mathrm{II}} = a_{22} - m_5 & C_5^{\mathrm{II}} = S_3^{\mathrm{II}} = m_{25} - m_9 \\
C_6^{\mathrm{II}} = S_2^{\mathrm{II}} = m_3 - a_9 & C_7^{\mathrm{II}} = S_1^{\mathrm{II}} = a_{24} + m_{10} \\
C_8^{\mathrm{II}} = S_0^{\mathrm{II}} = a_{23} + m_6. &
\end{array}
$$

Inverse 9-Point DCT/DST (DCT-III/DST-III) Module—12 Mults and 36 Adds: The input data sequence is $\{C_0^{\mathrm{II}}, C_1^{\mathrm{II}}, \cdots, C_8^{\mathrm{II}}\}$ or $\{S_8^{\mathrm{II}}, S_7^{\mathrm{II}}, \cdots, S_0^{\mathrm{II}}\}$, and the output data sequence is $\{x_0, x_1, \cdots, x_8\}$

For DCT-III:

$$
\begin{array}{lll}
a_1 = C_0^{\mathrm{II}} - C_6^{\mathrm{II}} & a_2 = C_1^{\mathrm{II}} - C_5^{\mathrm{II}} & a_3 = C_1^{\mathrm{II}} + C_5^{\mathrm{II}} \\
a_4 = C_2^{\mathrm{II}} - C_4^{\mathrm{II}} & a_5 = C_2^{\mathrm{II}} + C_4^{\mathrm{II}} & a_6 = C_2^{\mathrm{II}} + C_8^{\mathrm{II}} \\
a_7 = C_1^{\mathrm{II}} + C_7^{\mathrm{II}} & a_8 = a_6 - a_5 & a_9 = a_3 - a_7 \\
a_{10} = a_2 - C_4^{\mathrm{II}} & a_{11} = a_4 - C_8^{\mathrm{II}} &
\end{array}
$$

$$
\begin{array}{lll}
m_1 = d_1\, C_3^{\mathrm{II}} & m_2 = d_2\, C_6^{\mathrm{II}} & m_3 = d_1\, a_{10} \\
m_4 = d_2\, a_{11} & m_5 = d_5\, a_5 & m_6 = d_4\, a_6 \\
m_7 = d_5\, a_8 & m_8 = d_4\, a_5 & m_9 = d_8\, a_9 \\
m_{10} = d_7\, a_7 & m_{11} = d_8\, a_3 & m_{12} = d_7\, a_9
\end{array}
$$

$$
\begin{array}{lll}
a_{12} = C_0^{\mathrm{II}} + m_2 & a_{13} = a_{12} + m_1 & a_{14} = a_{12} - m_1 \\
a_{15} = a_1 + m_4 & a_{16} = m_5 + m_6 & a_{17} = m_7 - m_8 \\
a_{18} = a_{16} + a_{17} & a_{19} = m_9 + m_{10} & a_{20} = m_{11} - m_{12} \\
a_{21} = a_{20} - a_{19} & a_{22} = a_{13} + a_{16} & a_{23} = a_{14} + a_{16} \\
a_{24} = m_{14} + a_{17} & a_{25} = a_{13} + a_{17} & a_{26} = a_{14} - a_{18} \\
a_{27} = a_{13} - a_{18} & &
\end{array}
$$

$$
\begin{array}{lll}
x_0 = a_{22} + a_{19} & x_1 = a_{15} + m_3 & x_2 = a_{24} - a_{20} \\
x_3 = a_{26} - a_{21} & x_4 = a_1 - a_{11} & x_5 = a_{27} + a_{21} \\
x_6 = a_{25} - a_{20} & x_7 = a_{15} - m_3 & x_8 = a_{23} - a_{19}.
\end{array}
$$

For DST-III:

$$x_1 = -x_1 \quad x_3 = -x_3 \quad x_5 = -x_5 \quad x_7 = -x_7\,.$$

Substituting the presented 3-point and 9-point DCT/DST prime factor modules into the signal flow graphs in Figs. 1 and 2 results in the most efficient implementation of the MDCT/IMDCT computation in layer III of MPEG-1 and MPEG-2 audio coding standards. The final unique $(N/2)$ MDCT coefficients using the even antisymmetry property (3) are

$$
\begin{aligned}
C_{2k} &= \frac{\sqrt{2}}{2}\, Z_{2k}, \\
C_{2k+1} &= -\frac{\sqrt{2}}{2}\, Z_{N-2k-2}, \qquad k = 0, 1, \cdots, \frac{N}{4} - 1.
\end{aligned} \tag{4}
$$

The IMDCT computation can be realized simply by reversing the signal flow graphs for the forward MDCT computation and performing inverse operations. The resulting computational complexity for block size $N = 12$ is 13 multiplications and 39 (33) additions and for block size $N = 36$ is 47 (51) multiplications and 165 (151) additions.

III. Conclusions

The most efficient implementation of the MDCT/IMDCT computation for layer III in MPEG-1 and MPEG-2 international audio coding standards has been proposed. This implementation is based on the new fast algorithm for the MDCT/IMDCT computation in the oddly stacked system [4]. It is important to note that the new fast algorithm for the MDCT/IMDCT computation can also be easily adopted in other international standards and commercial products such as the recently established MPEG-2 advanced audio coding (AAC) [10], Sony adaptive transform acoustics coding (ATRAC) and ATRAC2 digital audio coding systems, the perceptual audio coder (AT&T PAC) [11], and the digital audio compression algorithm AC-3 [12].

References

[1] J. P. Princen, A. W. Johnson, and A. B. Bradley, “Subband/transform coding using filter bank designs based on time domain aliasing cancellation,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing ’87*, Dallas, TX, Apr. 1987, pp. 2161–2164.

[2] A. W. Johnson and A. B. Bradley, “Adaptive transform coding incorporating time domain aliasing cancellation,” *Speech Commun.*, vol. 6, pp. 299–308, Dec. 1987.

[3] P. Duhamel, Y. Mahieux, and J. P. Petit, “A fast algorithm for the implementation of filter banks based on time domain aliasing cancellation,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing ’91*, Toronto, ON, Canada, May 1991, pp. 2209–2212.

[4] V. Britanak and K. R. Rao, “New fast algorithms for the unified forward and inverse MDCT/MDST computation,” ICTR SAS, Tech. Rep., May 2000.

[5] M. T. Heideman, “Computation of an odd-length DCT from a real-valued DFT of the same length,” *IEEE Trans. Signal Processing*, vol. 40, pp. 54–61, Oct. 1992.

[6] Z. Wang, “A fast algorithm for the discrete sine transform implemented by the fast cosine transform,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 814–815, Oct. 1982.

[7] “Information Technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s—Part 3: Audio,”, ISO/IEC JTC1/SC29/WG11 MPEG, IS11172-3, (MPEG-1), 1992.

[8] “Information technology—Generic coding of moving pictures and associated audio—Part 3: Audio,”, ISO/IEC JTC1/SC29/WG11 MPEG, IS13818-3, (MPEG-2), 1994.

[9] K. R. Rao and J. J. Hwang, *Techniques and Standards for Digital Image/Video/Audio Coding*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[10] K. Brandenburg and M. Bosi, “MPEG-2 advanced audio coding: Overview and applications,” in *103th AES Conv.: Audio Engineering Soc.*, New York, 1997.

[11] V. K. Madisetti and D. B. Williams, Eds., “Digital audio communications,” in *The Digital Signal Processing Handbook*. Boca Raton, FL: CRC, 1998.

[12] “Digital audio compression (AC-3) standard,” Document of Advanced Television Systems Committee (ATSC), Audio Specialist Group T3/S7, Dec. 20, 1995.