Improved Radix-3 Decimation-in-Frequency Algorithm for the Fast Computation of Forward and Inverse MDCT

Jiasong Wu, Lu Wang, Lotfi Senhadji, Huazhong Shu

▶ **To cite this version:**

Jiasong Wu, Lu Wang, Lotfi Senhadji, Huazhong Shu. Improved Radix-3 Decimation-in-Frequency Algorithm for the Fast Computation of Forward and Inverse MDCT. The 2nd IEEE/IET International Conference on Audio, Language and Image Processing (ICALIP), Nov 2010, Shanghai, China. pp.694-699. inserm-00535839

HAL Id: inserm-00535839

https://inserm.hal.science/inserm-00535839

Submitted on 15 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improved Radix-3 Decimation-in-Frequency Algorithm for the Fast Computation of Forward and Inverse MDCT

Jiasong Wu[1,2,3,4], Lu Wang[1,4], Lotfi Senhadji[2,3,4], and Huazhong Shu[1,4]
[1]LIST, Southeast University, 2 Sipailou, 210096, Nanjing, China
Email: jswu@seu.edu.cn, wanglu@seu.edu.cn, shu.list@seu.edu.cn
[2]INSERM, U642, Rennes, F-35000, France
[3]LTSI, Université de Rennes 1, Campus Beaulieu, Rennes, F-35042, France
Email: lotfi.senhadji@univ-rennes1.fr
[4]Centre de Recherche en Information Biomédicale Sino-français (CRIBs), Rennes, France

Abstract

Shu et al. recently proposed an efficient radix-3 decimation-in-Frequency (DIF) algorithm for the fast computation of forward and inverse modified discrete cosine transform (MDCT). Their algorithm decomposes the MDCT of a sequence with length $N = 2\times3^m$ and the IMDCT of a sequence with length $N = 4\times3^m$ into three N/3-point MDCTs and IMDCTs, respectively. In this paper, we present a refinement of their approach. Comparison with Shu's algorithm shows that the number of arithmetic operations can be reduced from 17.5% to 20.5% for MDCT and from 23.1% to 27.9% for IMDCT when N is varying from 36 to 324.

1. Introduction

The problem of efficient computation of the modified discrete cosine transform (MDCT) and inverse MDCT (IMDCT) [1] has attracted attention of many researchers owing to its wide applications in several international standards and commercial products such as MPEG-1 [2], MPEG-2 [3], and AC-3 [4] in audio coding. In the past decade, many fast algorithms have been reported in the literature (see, e.g., [5], [6] for a survey). It is worth mentioning that the MDCT is equivalent to the modulated lapped transform (MLT) introduced by Malvar [7].

Since the layer III specifies two different block sizes: a long block ($N = 36$) and a short block ($N = 12$), the MDCT implementation via the complex-valued FFT can not be directly achieved. To solve this problem, Britanak [8-11], Lee [12], Cheng [13], and Truong [14] proposed various approaches based on the computation of lower order discrete cosine transform (DCT). Recently, Shu et al. [15] developed a fast radix-3 decimation-in-Frequency (DIF) algorithm for computing the MDCT of a sequence with length $N = 2\times3^m$ and the IMDCT of a sequence with length $N = 4\times3^m$. Wu *et al*. [16] proposed a mixed-radix MDCT algorithm based on radix-2 DIF algorithm and radix-3 decimation-in-time (DIT) algorithm.

In this paper, we propose a refinement of the algorithm presented in [15]. We improve their MDCT algorithm for data sequences with length $N = 4\times3^m$ and extend their IMDCT algorithm to support the length $N = 2\times3^m$. We show that the improved algorithm achieves a substantial reduction of the arithmetic operations compared to Shu's algorithm.

2. Original Shu's MDCT and IMDCT algorithm

The MDCT of an input data sequence $x(n)$, $n = 0, 1, 2, \ldots, N-1$, is defined as [1]

$$X(k) = \sum_{n=0}^{N-1} x(n)\cos\left[\frac{\pi}{2N}\left(2n+1+\frac{N}{2}\right)(2k+1)\right], \quad k = 0, 1, \ldots, N/2-1, \qquad (1)$$

and its inverse transform is given by

$$\tilde{x}(n) = \sum_{k=0}^{N/2-1} X(k)\cos\left[\frac{\pi}{2N}\left(2n+1+\frac{N}{2}\right)(2k+1)\right], \quad n = 0, 1, 2, \ldots, N-1, \qquad (2)$$

where the length N is assumed to be divisible by 6, i.e., $N = 2\times3^m$. In [15], the following approach to MDCT computation is described.

2.1. Computation of $A(k) = X(3k + 1)$, $k = 0, 1, \ldots, N/6 - 1$.

$$A(k) = \sum_{n=0}^{N/3-1} (a'_n - b'_n + c'_n)\cos\phi_{n,k}, \quad (3)$$

with $a_n=x(n)$, $b_n=x(N/3+n)$, $c_n=x(2N/3+n)$, $a'_n = a_{N/3-1-n}$, $b'_n = b_{N/3-1-n}$, $c'_n = c_{N/3-1-n}$, and $\phi_{n,k} = \frac{3\pi}{2N}\left(2n+1+\frac{N}{6}\right)(2k+1)$ for n = 0, 1, …, N/3–1.

2.2. Computation of $B(k) = X(3k) + X(3k + 2)$, k = 0, 1, …, N/6 – 1.

$$B(k) = \sum_{n=0}^{N/3-1} \left[(2c'_n - a'_n + b'_n)\sin\theta_n - \sqrt{3}(a'_n + b'_n)\cos\theta_n\right]\cos\phi_{n,k}. \quad (4)$$

with $\theta_n = (2n+1)\pi / N$.

2.3. Computation of $C(k) = X(3k) – X(3k + 2)$, k = 0, 1, …, N/6 – 1.

$$C(k) = (-1)^k \sum_{n=0}^{N/3-1} \left[(2a_n + b_n - c_n)\cos\theta_n + \sqrt{3}(b_n + c_n)\sin\theta_n\right]\cos\phi_{n,k}. \quad (5)$$

Note that $A(k)$, $B(k)$ and $C(k)$ are all the N/3-length MDCTs. In [15], a radix-3 algorithm was derived for IMDCT of sequences with length $N = 4\times3^m$ which is summarized below.

2.4. Computation of $A'(n) = \tilde{x}(3n+1)$, n = 0, 1, …, N/3 – 1.

$$A'(n) = \sum_{k=0}^{N/6-1} (f_k - g'_k - h_k)\cos\phi_{n,k}, \quad (6)$$

with $f_k=X(k)$, $g_k=X(N/6+k)$, $h_k=X(N/3+k)$, and $g'_k = g_{N/6-1-k}$ for k = 0, 1, …, N/6 – 1.

2.5. Computation of $B'(n) = \tilde{x}(3n) + \tilde{x}(3n+2)$, n = 0, 1, …, N/3 – 1.

$$B'(n) = \sum_{k=0}^{N/6-1} \left[(2f_k + g'_k + h_k)\cos\theta_k + \sqrt{3}(h_k - g'_k)\sin\theta_k\right]\cos\phi_{n,k}, \quad (7)$$

with $\theta_k = (2k+1)\pi / N$.

2.6. Computation of $C'(n) = \tilde{x}(3n) - \tilde{x}(3n+2)$, n = 0, 1, …, N/3 – 1.

$$C'(n) = (-1)^{n+N/12} \sum_{k=0}^{N/6-1} \left[\sqrt{3}(f'_k + g_k)\cos\theta_k - (f'_k - g_k + 2h'_k)\sin\theta_k\right]\cos\phi_{n,k}. \quad (8)$$

with $f'_k = f_{N/6-1-k}$ and $h'_k = h_{N/6-1-k}$.

Note that the assumption of N being a multiple of 12 instead of a multiple of 6 is necessary to obtain equation (8).

3. Improved MDCT and IMDCT algorithm

In practical application, the length of sequences, N, is often divisible by 4. For this reason, we present in this section an improved approach for computing the MDCT of sequence length $N = 4\times3^m$. We also propose an improved algorithm for computing the IMDCT in which the sequence length N needs only to be a multiple of 6. More importantly, the proposed IMDCT algorithm is more efficient than that reported in [15].

3.1. Improved MDCT algorithm

In this subsection, we consider the case where the sequence length N is divisible by 4, i.e., $N = 4\times3^m$, $m \geq 1$. In this case, $A(k)$ and $B(k)$ can be obtained with (3) and (4), and $C(k)$ is calculated in the following manner.

$$\begin{aligned} C(k) &= \sum_{n=0}^{N-1} 2x(n)\cos\left[\frac{\pi}{N}(2n+1)\right] \times\sin\left[\frac{3\pi}{2N}\left(2n+1+\frac{N}{2}\right)(2k+1)\right] \\ &= \sum_{n=0}^{N/3-1} 2x\left(\frac{N}{3}-1-n\right)\cos\left(\frac{2\pi}{3}-\theta_n\right) \times\sin\left[2\pi(2k+1)-\phi_{n,k}\right] \\ &\quad + \sum_{n=0}^{N/3-1} 2x\left(\frac{2N}{3}-1-n\right)\cos\left(\frac{4\pi}{3}-\theta_n\right) \times\sin\left[3\pi(2k+1)-\phi_{n,k}\right] \\ &\quad + \sum_{n=0}^{N/3-1} 2x\left(N-1-n\right)\cos\left(2\pi-\theta_n\right) \times\sin\left[4\pi(2k+1)-\phi_{n,k}\right] \\ &= \sum_{n=0}^{N/3-1} \left[-(2c'_n - a'_n + b'_n)\cos\theta_n - \sqrt{3}(a'_n + b'_n)\sin\theta_n\right]\sin\phi_{n,k}. \end{aligned} \quad (9)$$

If N is a multiple of 12, then Eq. (9) can be rewritten as

$$C\left(\frac{N}{6}-1-k\right) = \sum_{n=0}^{N/3-1} \left[-(2c'_n - a'_n + b'_n)\cos\theta_n - \sqrt{3}(a'_n + b'_n)\sin\theta_n\right]\sin\left[\frac{\pi}{2}\left(2n+1+\frac{N}{6}\right) - \phi_{n,k}\right]$$

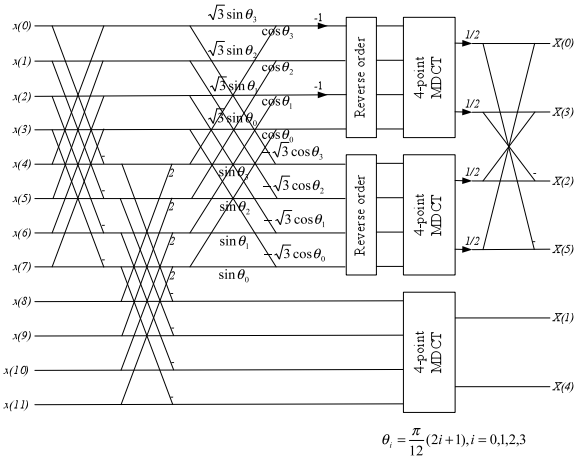

Figure 1. Flow graph of a 12-point MDCT

$$= \sum_{n=0}^{N/3-1} (-1)^{n+N/12+1} \left[(2c'_n - a'_n + b'_n)\cos\theta_n + \sqrt{3}(a'_n + b'_n)\sin\theta_n \right] \cos\phi_{n,k}. \quad (10)$$

Eq. (10) shows that $C(k)$ is the length-$N/3$ MDCT. Moreover, the input sequence of (10) is similar to that of (4) except for the twiddle factors. Figure 1 shows the flow graph of the realization of 12-point MDCT.

3.2. Improved IMDCT algorithm

In this subsection, we show how to calculate the IMDCT in a more efficient way.

3.2.1. Computation of $A'(n) = \tilde{x}(3n+1)$, n = 0, 1, …, $N/3 - 1$.

$$A'(n) = \sum_{k=0}^{N/2-1} X(k)\cos\left[\frac{\pi}{2N}\left(6n+3+\frac{N}{2}\right)(2k+1)\right] = A'_1(n) + A'_2(n) + A'_3(n), \quad (11)$$

where

$$A'_1(n) = \sum_{k=0}^{N/6-1} X(k)\cos\phi_{n,k}, \quad (12)$$

$$A'_2(n) = \sum_{k=N/6}^{N/3-1} X(k)\cos\phi_{n,k}, \quad (13)$$

$$A'_3(n) = \sum_{k=N/3}^{N/2-1} X(k)\cos\phi_{n,k}. \quad (14)$$

For $A'_2(n)$, we have

$$A'_2(n) = \sum_{k=0}^{N/6-1} X(N/3-1-k) \times \cos\left[\pi\left(2n+1+\frac{N}{6}\right) - \phi_{n,k}\right]. \quad (15)$$

Since N is a multiple of 6, we obtain

$$A'_2(n) = \sum_{k=0}^{N/6-1} (-1)^{N/6+1} X(N/3-1-k)\cos\phi_{n,k}. \quad (16)$$

For $A'_3(n)$, we have

$$A'_3(n) = \sum_{k=0}^{N/6-1} (-1)^{N/6+1} X(N/3+k)\cos\phi_{n,k}. \quad (17)$$

Therefore

$$A'(n) = \sum_{k=0}^{N/6-1} \left[f_k + (-1)^{N/6+1}(g'_k + h_k) \right] \cos\phi_{n,k}. \quad (18)$$

3.2.2. Computation of $B'(n) = \tilde{x}(3n) + \tilde{x}(3n+2)$, n = 0, 1, …, $N/3 - 1$.

$$B'(n) = \sum_{k=0}^{N/2-1} 2X(k)\cos\left[\frac{\pi}{N}(2k+1)\right]\cos\phi_{n,k}. \quad (19)$$

Proceeding with the computation of $B'(n)$ in a similar way as for $A'(n)$, we obtain

$$B'(n) = \sum_{k=0}^{N/6-1} \left\{ \left[2f_k + (-1)^{N/6}(g'_k + h_k) \right]\cos\theta_k - (-1)^{N/6}\sqrt{3}(g'_k - h_k)\sin\theta_k \right\} \cos\phi_{n,k}. \quad (20)$$

3.2.3. Computation of $C'(n) = \tilde{x}(3n) - \tilde{x}(3n+2)$, n = 0, 1, …, $N/3 - 1$.

$$C'(n) = \sum_{k=0}^{N/2-1} 2X(k)\sin\left[\frac{\pi}{N}(2k+1)\right]\sin\phi_{n,k}. \quad (21)$$

Similar to $B'(n)$, we have

$$C'(n) = \sum_{k=0}^{N/6-1} \left\{ \left[2f_k + (-1)^{N/6}(g'_k + h_k) \right]\sin\theta_k + (-1)^{N/6}\sqrt{3}(g'_k - h_k)\cos\theta_k \right\} \sin\phi_{n,k}. \quad (22)$$

Eq. (22) can be rewritten as

$$C'\left(\frac{N}{3}-1-n\right) = \sum_{k=0}^{N/6-1} \left\{ \left[2f_k + (-1)^{N/6}(g'_k + h_k) \right]\sin\theta_k + (-1)^{N/6}\sqrt{3}(g'_k - h_k)\cos\theta_k \right\} \sin\left[\frac{3\pi}{2}(2k+1) - \phi_{n,k}\right]$$

$$= \sum_{k=0}^{N/6-1} (-1)^{k+1} \left\{ \left[2f_k + (-1)^{N/6}(g'_k + h_k) \right]\sin\theta_k + (-1)^{N/6}\sqrt{3}(g'_k - h_k)\cos\theta_k \right\} \cos\phi_{n,k}. \quad (23)$$

Eqs. (18), (20) and (23) show that $A'(n)$, $B'(n)$ and $C'(n)$ are all $N/6$-point IMDCTs. Contrary to the algorithm proposed in [15], the assumption of N being a multiple of 12 is not required to derive equation (23). Moreover, the input sequence of (23) is similar to that

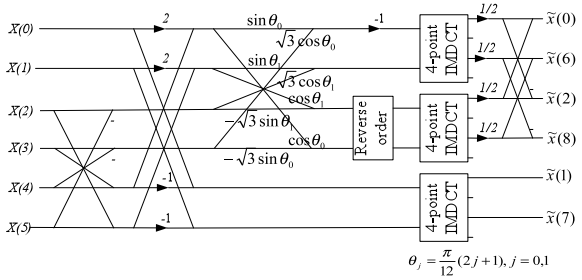

Figure 2. Flow graph of a 12-point IMDCT

of (20) except for the twiddle factors. This permits reducing the number of arithmetic operations required in the calculation of $B'(n)$ and $C'(n)$. The flow graph corresponding to a 12-point IMDCT computation is shown in Figure 2.

4. Computational complexity and comparison results

In this section, we will analyze the computational complexity of the improved radix-3 DIF MDCT and IMDCT algorithms and compare the results with the original algorithms [15]. The comparison of the advantages and disadvantages of these MDCT-based algorithms with the DCT-based algorithms [5-14], we refer the readers to [15] and [16].

4.1. Computational complexity of the proposed MDCT algorithm

Since $A(k)$, $B(k)$ and $C(k)$ are all the $N/3$-length MDCTs, the coefficients $X(k)$ can be obtained from $A(k)$, $B(k)$ and $C(k)$, k = 0, 1, …, $N/6 - 1$, with $N/3$ additions. Hence, we can calculate an N-length MDCT via the computation of three $N/3$-point MDCTs. Note that the computation of $(2c'_n - a'_n + b'_n)\sin\theta_n - \sqrt{3}(a'_n + b'_n)\cos\theta_n$ in (4) and $(2c'_n - a'_n + b'_n)\cos\theta_n + \sqrt{3}(a'_n + b'_n)\sin\theta_n$ in (10) requires four multiplications for each n. However, for $n = 4\times3^m$, when $n = (N/4 - 1)/2$, we have $\theta_n = \pi/4$, $\cos\theta_n = \sin\theta_n = 1/\sqrt{2}$. In this case, two multiplications can be saved. Thus, the computational complexity is given by

$$\begin{cases} M^P_{\mathrm{MDCT}}(N) = 3M^P_{\mathrm{MDCT}}(N/3) + 4N/3 - 2 \\ A^P_{\mathrm{MDCT}}(N) = 3A^P_{\mathrm{MDCT}}(N/3) + 7N/3, \end{cases}$$

for $N = 4\times3^m$, $m \ge 1$. (24)

with $M^P_{\mathrm{MDCT}}(4) = 3$ and $A^P_{\mathrm{MDCT}}(4) = 5$, where M, A and P denote Multiplications, Additions and the Proposed algorithm, respectively.

Table 1. Comparison of the improved MDCT algorithm with Shu's algorithm in terms of arithmetic complexity for $N = 4\times3^m$, $m \ge 1$.

	Shu's algorithm [15]			Improved algorithm			Saved
N	Mul	Add	Total	Mul	Add	Total	(%)
12	25	55	80	23	43	66	17.5
36	123	285	408	115	213	328	19.6
108	513	1215	1728	487	891	1378	20.3
324	1971	4725	6696	1891	3429	5320	20.5

When applying the algorithm presented in [15] for $N = 4\times3^m$, neither multiplication nor addition can be saved. The computational complexity of Shu's algorithm is

$$\begin{cases} M^S_{\mathrm{MDCT}}(N) = 3M^S_{\mathrm{MDCT}}(N/3) + 4N/3 \\ A^S_{\mathrm{MDCT}}(N) = 3A^S_{\mathrm{MDCT}}(N/3) + 10N/3, \end{cases}$$

for $N = 4\times3^m$, $m \ge 1$. (25)

where S denotes Shu's algorithm. Table 1 lists the number of arithmetic operations needed by the two approaches for $N = 4\times3^m$, $m \ge 1$.

4.2. Computational complexity of the proposed IMDCT algorithm

To obtain the input data sequences in (20) and (23), four multiplications are required for each k. However, for $N = 2\times3^m$, $m \ge 1$, when $k = (N/6 - 1)/2$, we have $\theta_k = \pi/6$, $\cos\theta_k = 1/2$, $\sin\theta_k = \sqrt{3}/2$. In this case, we can save three multiplications and two additions. Meanwhile, we can obtain the sequence $\tilde{x}(n)$ from $A'(n), B'(n)$ and $C'(n)$ with $2N/3$ additions. The computational complexity of the proposed approach is given by

$$\begin{cases} M^P_{\mathrm{IMDCT}}(N) = 3M^P_{\mathrm{IMDCT}}(N/3) + 2N/3 - 3 \\ A^P_{\mathrm{IMDCT}}(N) = 3A^P_{\mathrm{IMDCT}}(N/3) + 5N/3 - 2, \end{cases}$$

for $N = 2\times3^m$, $m \ge 1$. (26)

with $M^P_{\mathrm{IMDCT}}(6) = 1$ and $A^P_{\mathrm{IMDCT}}(6) = 4$ (See[16]).

For $N = 4\times3^m$, $m \ge 1$, when $k = (N/4–1)/2$, we have $\theta_k = \pi/4$. In this case, two multiplications can be saved. Moreover, the sequence $\{\tilde{x}(n)\}$ has the following symmetries [13]

$$\begin{cases} \tilde{x}(3N/4 + n) = \tilde{x}(3N/4 - 1 - n) \\ \tilde{x}(N/2 - 1 - n) = -\tilde{x}(n), \end{cases}$$

n = 0, 1, …, $N/4 - 1$, (27)

As a result, only $\tilde{x}(n)$, for n = 0, 1, …, $N/4 - 1$ and n = $N/2$, $N/2 + 1$, …, $3N/4 - 1$, need to be calculated. Therefore, we can only evaluate $A'(n)$, $B'(n)$ and $C'(n)$ for

Table 2. Comparison of the improved IMDCT algorithm with Shu's algorithm in terms of arithmetic complexity for $N = 4\times3^m$, $m \geq 1$.

	Shu's algorithm [15]			Improved algorithm			Saved
N	Mul	Add	Total	Mul	Add	Total	(%)
12	17	35	52	15	25	40	23.1
36	75	183	258	67	123	190	26.4
108	297	783	1080	271	513	784	27.4
324	1107	3051	4158	1027	1971	2998	27.9

$n = 0,1,\cdots,\left\lfloor \frac{N/6-1}{2} \right\rfloor, \frac{N}{6}, \frac{N}{6}+1,\cdots,\frac{N}{6}+\left\lfloor \frac{N/6-1}{2} \right\rfloor$ from which we can obtain $\widetilde{x}(n)$ with $N/3$ additions. The computational complexity is given by

$$\begin{cases} M^P_{\mathrm{IMDCT}}(N) = 3M^P_{\mathrm{IMDCT}}(N/3) + 2N/3 - 2 \\ A^P_{\mathrm{IMDCT}}(N) = 3A^P_{\mathrm{IMDCT}}(N/3) + 4N/3, \end{cases} \quad \text{for } N = 4\times3^m,\ m \geq 1. \qquad (28)$$

with $M^P_{\mathrm{IMDCT}}(4) = 3$ and $A^P_{\mathrm{IMDCT}}(4) = 3$.

In [15], the computational complexity is given by:

$$\begin{cases} M^S_{\mathrm{IMDCT}}(N) = 3M^S_{\mathrm{IMDCT}}(N/3) + 2N/3 \\ A^S_{\mathrm{IMDCT}}(N) = 3A^S_{\mathrm{IMDCT}}(N/3) + 13N/6, \end{cases} \quad \text{for } N = 4\times3^m,\ m \geq 1. \qquad (29)$$

In Table 2, we summarize the computational complexity of the two algorithms with length $N = 4\times3^m$, $m \geq 1$.

5. Conclusions

We have presented an improvement of the recently published algorithm for computing the MDCT and IMDCT based on the calculation of lower order MDCTs and IMDCTs, respectively. The comparison results show that the proposed approach outperforms the original one in terms of the number of arithmetic operations.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 60873048, by the National Basic Research Program of China under Grant 2010CB732503, by a Program of Jiangsu Province under Grant SBK200910055 and the Natural Science Foundation of Jiangsu Province under Grant BK2008279. This work is also supported by the "Eiffel Doctorate" excellence grant of the French Ministry of Foreign and European Affairs.

7. References

[1] J. P. Princen, A. W. Johnson, and A. B. Bradley, "Subband/transform coding using filter bank design based on time domain aliasing cancellation," in *PROC. IEEE ICASSP'87*, Dallas, TX, 1987, pp. 2161-2164.

[2] "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, IS 11172-3, (MPEG-1), 1992.

[3] "Information technology—Generic coding of moving pictures and associated audio—Part 3: Audio,", ISO/IEC, IS 13818-3, (MPEG-2), 1994.

[4] *Digital audio compression (AC-3) standard*, Audio Specialist Group T3/S7, Dec. 1995.

[5] V. Britanak and H.J. Lincklaen Arriens, "Fast computational structures for an efficient implementation of the complete TDAC analysis/ synthesis MDCT/MDST filter banks," *Signal Process*., vol. 89, no. 7, pp. 1379-1394, July 2009.

[6] V. Britanak, "New universal rotation-based fast computational structures for an efficient implementation of the DCT-IV/DST-IV and analysis/synthesis MDCT/MDST filter banks," *Signal Process.*, vol. 89, no. 11, pp. 2213-2232, Nov. 2009.

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

[8] V. Britanak and K. R. Rao, "An efficient implementation of the forward and inverse MDCT in MPEG audio coding," *IEEE Signal Process. Lett.*, vol. 8, pp. 48-51, 2001.

[9] V. Britanak and K. R. Rao, "A new fast algorithm for the unified forward and inverse MDCT/MDST computation," *Signal Process.*, vol. 82, pp. 433-459, 2002.

[10] V. Britanak, "An efficient computing of oddly stacked MDCT/MDST via evenly stacked MDCT/MDST and vice-versa," *Signal Process.*, vol. 85, pp. 1353-1374, 2005.

[11] V. Britanak, "New fast computational structures for an efficient implementation of the forward/backward MDCT in MP3 audio coding standard," *Signal Process.*, vol. 90, no. 2, pp. 536-547, Feb. 2010.

[12] S. W. Lee, "Improved algorithm for efficient computation of the forward and backward MDCT in MPEG audio coder," *IEEE Trans. Circuits Syst. II*, vol. 48, pp.990-994, 2001.

[13] M. H. Cheng and Y. H. Hsu, "Fast IMDCT and MDCT algorithms—a matrix approach," *IEEE Trans. Signal Process.*, vol. 51, pp. 221-229, 2003.

[14] T.K. Truong, P.D. Chen, and T.C. Cheng, "Fast algorithm for computing the forward and inverse MDCT in MPEG audio coding ," *Signal Process.*, vol. 86, no. 5, pp. 1055-1060, May 2006.

[15] H. Shu, X. Bao, C. Toumoulin, and L. Luo, "Radix-3 algorithm for the fast computation of forward and inverse MDCT," *IEEE Signal Process. Lett*., vol. 14, no. 2, pp. 93-96, Feb. 2007.

[16] J.S. Wu, H.Z. Shu, L. Senhadji, and L.M. Luo, "Mixed-radix algorithm for the computation of forward and inverse MDCTs," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 56, no. 4, pp. 784-794, Apr. 2009.