Fast Decomposition of Filterbanks for the State-of-the-Art Audio Coding

Shih-Way Huang, Tsung-Han Tsai, *Member, IEEE*, and Liang-Gee Chen, *Fellow, IEEE*

***Abstract*—This letter derives fast decomposition for the quadrature mirror filterbanks (QMFs) of the low power spectral band replication (SBR) tools in the MPEG high efficiency advanced audio coding (HE AAC) decoder. In contrast with the standard method where computation-intensive matrix operations are employed in the QMF, the proposed method decomposes the matrix operations into conventional discrete cosine transform of type II and III (DCT-II and DCT-III) and simple permutations for easy implementation. The computational complexity can be also reduced effectively by using fast algorithms for DCT.**

***Index Terms*—Audio coding, fast algorithm.**

I. INTRODUCTION

THE state-of-the-art audio coding, MPEG-4 high efficiency advanced audio coding (HE AAC) [1], includes core AAC to be compatible with the conventional AAC, and new spectral band replication (SBR) tools are embedded to reach high audio quality at much lower bitrates with increasing complexity. Reference [2] presents low power SBR with aliasing minimizing tools to reduce the complexity of the decoder by processing real-valued data instead of complex-valued data. However, the complexity is still high without optimization. From the complexity analyses in [2], filterbanks are the most computationally dominant parts in the decoder. There are three different filterbanks in an HE AAC decoder. One is based on inverse modified discrete cosine transform (IMDCT) in the core AAC decoder, another is the analysis quadrature mirror filterbank (AQMF) in new SBR tools, and the other is the synthesis quadrature mirror filterbank (SQMF) in the same SBR tools. Because of mature researches of the core AAC decoder, there are already fast algorithms for IMDCT. Therefore, the other QMF in the new SBR tools requires effective techniques to reduce complexity.

This letter derives fast algorithms to reduce the complexity of the computation-intensive matrix operations in AQMF and SQMF. The proposed methods comprise only conventional discrete cosine transform of type II and III (DCT-II and DCT-III) [3] and simple permutations. By referring to the fast DCT in [4], the computational complexity can be reduced up to 2.7% and 7.8% with respect to the original multiplications and additions. Moreover, we provide the easily implemented solution of HE AAC with the same unified DCT computations in filterbanks.

Manuscript received February 17, 2005; revised May 9, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Markus Pueschel.

S.-W. Huang and L.-G. Chen are with the Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: shihway@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

T.-H. Tsai is with the Department of Electrical Engineering, National Central University, Chung-Li 32001, Taiwan, R.O.C. (e-mail: han@ee.ncu.edu.tw).

Digital Object Identifier 10.1109/LSP.2005.855550

II. REVIEW

The matrix operations (scale factors are neglected) in the QMF are defined as follows.

In AQMF

$$X(k) = \sum_{n=0}^{63} u(n) \cos\left[\frac{\pi}{64}(n-48)(2k+1)\right] \quad \text{for} \quad k = 0,1,\ldots,31; \quad n = 0,1,\ldots,63. \tag{1}$$

In SQMF

$$v(n) = \sum_{k=0}^{63} X(k) \cos\left[\frac{\pi}{128}(n-32)(2k+1)\right] \quad \text{for} \quad k = 0,1,\ldots,63; \quad n = 0,1,\ldots,127. \tag{2}$$

In order to circumvent the situations in which the output signals have a sampling rate twice that of the input, the downsampled 32-channel SQMF can be employed instead of the 64-channel SQMF.

In downsampled SQMF

$$v(n) = \sum_{k=0}^{31} X(k) \cos\left[\frac{\pi}{64}(n-16)(2k+1)\right] \quad \text{for} \quad k = 0,1,\ldots,31; \quad n = 0,1,\ldots,63 \tag{3}$$

where n is the time index, and k is the frequency index.

From the above equations, 2048 multiplications and 2016 additions are required for the matrix operation in AQMF; 8192 multiplications and 8064 additions are required for that in 64-channel SQMF; and 2048 multiplications and 1984 additions are required for that in 32-channel downsampled SQMF. This makes the QMF computation intensive and difficult to be implemented. Hence, fast algorithms for QMF are required to reduce the complexity.

III. DEVELOPMENT METHODS

The concept of the AQMF is similar to the analysis subband of MPEG-1 audio encoding [5], whereas the concept of the SQMF is similar to the synthesis subband of MPEG-1 audio decoding. However, the AQMF and SQMF are variants with respect to the counterparts in MPEG-1 audio coding because of different choices of the prototype filters. In [6] and [7], fast algorithms for the matrix operations in analysis and synthesis subband of MPEG-1 audio are presented. Both papers have the similar concept to decompose the original equations into a conventional DCT type with a corresponding permutation. In the similar way, we decompose (1)–(3) into a conventional DCT type with a corresponding permutation. The derivation is shown in the following.

A. AQMF

First, let (4) be defined as

$$X(k)=\sum_{n=0}^{63}u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right] \quad \text{for}\quad k=0,1,\ldots,31;\quad n=0,1,\ldots,63. \tag{4}$$

The relationship between $u'(n)$ in (4) and $u(n)$ in (1) is derived. Let $\mathrm{j}=\mathrm{n}-48$; then, (1) is rewritten as

$$\begin{aligned}X(k)&=\sum_{j=-48}^{15}u(j+48)\cos\left[\frac{\pi}{64}j(2k+1)\right]\\&=\sum_{j=-48}^{-1}u(j+48)\cos\left[\frac{\pi}{64}j(2k+1)\right]\\&\quad+\sum_{j=0}^{15}u(j+48)\cos\left[\frac{\pi}{64}j(2k+1)\right].\end{aligned}$$

Substituting $\mathrm{m}=\mathrm{j}+64$ into the above yields

$$\begin{aligned}X(k)&=\sum_{m=16}^{63}u(m-16)\cos\left[\frac{\pi}{64}(m-64)(2k+1)\right]\\&\quad+\sum_{j=0}^{15}u(j+48)\cos\left[\frac{\pi}{64}j(2k+1)\right]\\&=\sum_{m=16}^{63}(-u(m-16))\cos\left[\frac{\pi}{64}m(2k+1)\right]\\&\quad+\sum_{j=0}^{15}u(j+48)\cos\left[\frac{\pi}{64}j(2k+1)\right].\end{aligned}\tag{5}$$

Compare (5) with (4); then, the relationship is obtained by

$$u'(n)=\begin{cases}u(n+48), & n=0,1,\ldots,15\\ -u(n-16), & n=16,17,\ldots,63.\end{cases}\tag{6}$$

Second, let (7) be defined as the 32-point DCT-III [3]

$$X(k)=\sum_{n=0}^{31}u''(n)\cos\left[\frac{\pi}{64}n(2k+1)\right] \quad \text{for}\quad k=0,1,\ldots,31;\quad n=0,1,\ldots,31. \tag{7}$$

Then, the relationship between $u''(n)$ in (7) and $u'(n)$ in (4) is derived as follows:

$$\begin{aligned}X(k)&=u'(0)\cos\left[\frac{\pi}{64}0(2k+1)\right]\\&\quad+\sum_{n=1}^{31}u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]\\&\quad+u'(32)\cos\left[\frac{\pi}{64}32(2k+1)\right]\\&\quad+\sum_{n=33}^{63}u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]\\&=u'(0)+\sum_{n=1}^{31}u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]\\&\quad+\sum_{n=1}^{31}u'(64-n)\cos\left[\frac{\pi}{64}(64-n)(2k+1)\right]\\&=u'(0)+\sum_{n=1}^{31}u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]\\&\quad-\sum_{n=1}^{31}u'(64-n)\cos\left[\frac{\pi}{64}n(2k+1)\right]\\&=u'(0)+\sum_{n=1}^{31}(u'(n)-u'(64-n))\\&\quad\times\cos\left[\frac{\pi}{64}n(2k+1)\right].\end{aligned}\tag{8}$$

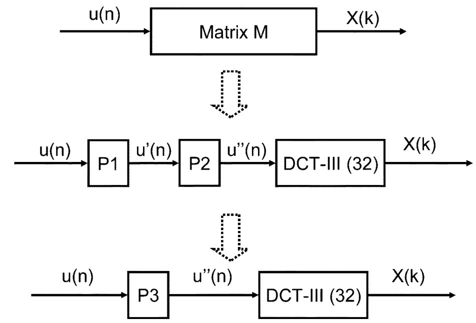

Fig. 1. Decomposition of the matrix operation in AQMF.

Compare (8) with (7); then, the relationship is obtained by

$$u''(n)=\begin{cases}u'(0), & n=0\\ u'(n)-u'(64-n), & n=1,2,\ldots,31.\end{cases}\tag{9}$$

Combining (6) and (9) yields

$$u''(n)=\begin{cases}u(48), & n=0\\ u(n+48)+u(48-n), & n=1,2,\ldots,15\\ -u(n-16)+u(48-n), & n=16,17,\ldots,31.\end{cases}\tag{10}$$

The above derivation is illustrated in Fig. 1, where (10) stands for the permutation and is represented as $P3$. Fig. 2 details the signal flow graph of the $P3$ in the proposed decomposition. Finally, this result can be equivalently written in sparse matrix factorizations. Equation (1) is rewritten as $X=uM$. Let I be the identity, $\bar{I}$ be the identity with columns reversed, and C_{32}^{III} be the 32-point DCT-III. Consequently, matrix M can be equivalently written as follows:

$$M=\begin{bmatrix}0 & -I_{16}\\ 0 & A_{16\times16}\\ A_{16\times16} & B_{16\times16}\\ I_{16} & 0\end{bmatrix}C_{32}^{III}$$

where

$$A_{16\times16}=\begin{bmatrix}0 & 0 & \cdots & 0\\ 0 & & & \\ \vdots & & \bar{I}_{15} & \\ 0 & & & \end{bmatrix}$$

and

$$B_{16\times16}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \\ 0 & 0 & \cdots & 0\end{bmatrix}.$$

B. SQMF

First, let (11) be defined

$$v'(n)=\sum_{k=0}^{63}X(k)\cos\left[\frac{\pi}{128}(n)(2k+1)\right] \quad \text{for}\quad k=0,1,\ldots,63;\quad n=0,1,\ldots,127. \tag{11}$$

Then, the relationship between $v'(n)$ in (11) and $v(n)$ in (2) is derived.

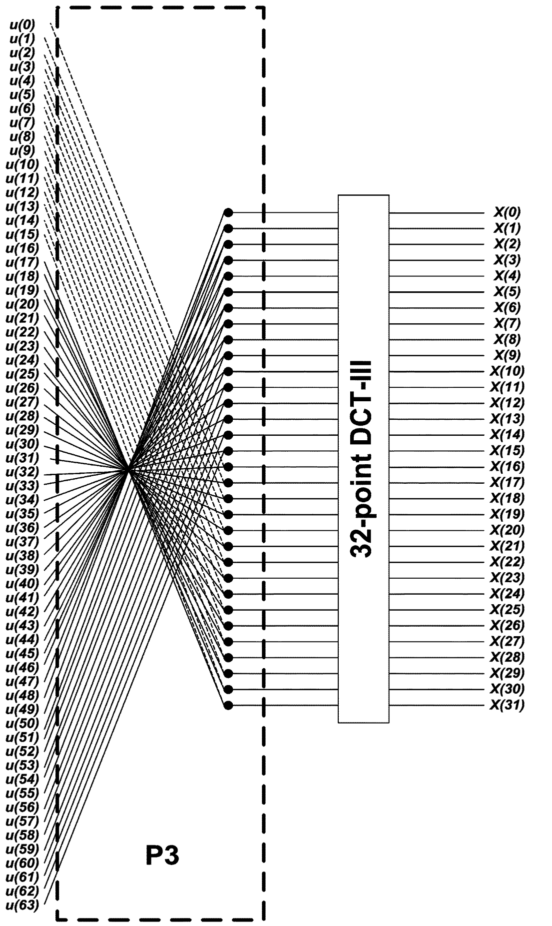

Fig. 2. Signal flow graph of the proposed decomposition in AQMF.

For $n=0,1,\ldots,31$, $v(n)=\sum_{k=0}^{63}X(k)\cos[\pi/128(n-32)(2k+1)]=\sum_{k=0}^{63}X(k)\cos[\pi/128(-(n-32))(2k+1)]$.

For $n=32,33,\ldots,128$, the relationship is intuitive. Therefore

$$v(n)=\begin{cases}v'(32-n), & n=0,1,\ldots,31\\ v'(n-32), & n=32,33,\ldots,127.\end{cases}\qquad(12)$$

Second, let (13) be defined as the 64-point DCT-II [3]

$$v''(n)=\sum_{k=0}^{63}X(k)\cos\left[\frac{\pi}{128}(n)(2k+1)\right]\quad\text{for}\quad k=0,1,\ldots,63;\quad n=0,1,\ldots,63.\qquad(13)$$

Then, the relationship between $v''(n)$ in (13) and $v'(n)$ in (11) is derived.

For $n=0,1,\ldots,63, v''(n)=v'(n),$ for $n=64, v''(n)=0,$ and for $n=65,66,\ldots,127$

$$\begin{aligned}v'(n)&=\sum_{k=0}^{63}X(k)\cos\left[\frac{\pi}{128}(n)(2k+1)\right]\\&=-\sum_{k=0}^{63}X(k)\cos\left[\frac{\pi}{128}(128-n)(2k+1)\right].\end{aligned}$$

Therefore

$$v'(n)=\begin{cases}v''(n), & n=0,1,\ldots,63\\ 0, & n=64\\ -v''(128-n), & n=65,66,\ldots,127.\end{cases}\qquad(14)$$

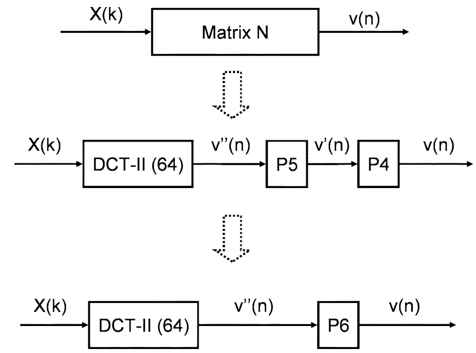

Fig. 3. Decomposition of the matrix operation in SQMF.

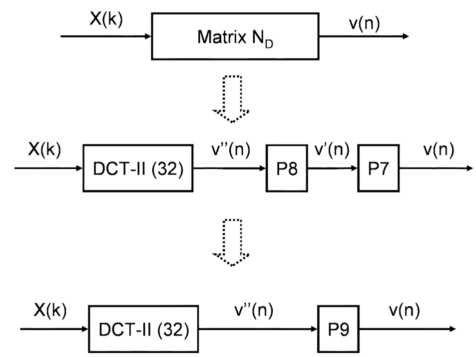

Fig. 4. Decomposition of the matrix operation in downsampled SQMF.

Combining (12) and (14) yields the direct relationship

$$v(n)=\begin{cases}v''(32-n), & n=0,1,\ldots,31\\ v''(n-32), & n=32,33,\ldots,95\\ 0, & n=96\\ -v''(160-n), & n=97,98,\ldots,127.\end{cases}\qquad(15)$$

The above derivation is illustrated in Fig. 3, and (15) stands for the permutation $P6$. Finally, this result can be equivalently written in sparse matrix factorizations. Equation (2) is rewritten as $v=X\ N$. Let C_{64}^{II} be the 64-point DCT-II. Consequently, matrix N can be equivalently written as follows:

$$N=C_{64}^{II}\begin{bmatrix}A_{32\times32} & I_{32} & 0 & 0\\ B_{32\times32} & 0 & I_{32} & D_{32\times32}\end{bmatrix}$$

where

$$A_{32\times32}=\begin{bmatrix}0 & 0 & \cdots & 0\\ 0 & & & \\ \vdots & & \bar{I}_{31} & \\ 0 & & & \end{bmatrix}$$

$$B_{32\times32}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \\ 0 & 0 & \cdots & 0\end{bmatrix}$$

and

$$D_{32\times32}=\begin{bmatrix}0 & 0 & \cdots & 0\\ 0 & & & \\ \vdots & & -\bar{I}_{31} & \\ 0 & & & \end{bmatrix}.$$

C. Downsampled SQMF

The derivation for downsampled SQMF is similar to that for SQMF, except that the length is decreased to the half. Therefore, the 32-point DCT-II is substituted for the 64-point DCT-II.

TABLE I
REDUCTION OF COMPUTATIONAL COMPLEXITY IN THE PROPOSED QMF

Filterbank	Operation	Original	Proposed			Proposed/
			Fast DCT	Permutation	Total	Original
AQMF	Mult	2048	80	0	80	3.9%
	Add	2016	209	31	240	11.9%
SQMF (64-channel)	Mult	8192	192	0	192	2.3%
	Add	8064	513	31	544	6.7%
Total	Mult	10240	272	0	272	2.7%
	Add	10080	722	62	784	7.8%

Let (16) be defined as the 32-point DCT-II [3]

$$v''(n) = \sum_{k=0}^{31} X(k) \cos\left[\frac{\pi}{64}(n)(2k+1)\right]$$
$$\text{for} \quad k = 0, 1, \ldots, 31; \quad n = 0, 1, \ldots, 31. \quad (16)$$

Therefore, the relationship between $v(n)$ in (3) and $v''(n)$ in (16) is as follows:

$$v(n) = \begin{cases} v''(16-n), & n = 0, 1, \ldots, 15 \\ v''(n-16), & n = 16, 18, \ldots, 47 \\ 0, & n = 48 \\ -v''(80-n), & n = 49, 50, \ldots, 63. \end{cases} \quad (17)$$

The above derivation is illustrated in Fig. 4, and (17) stands for the permutation $P9$. Finally, this result can be equivalently written in sparse matrix factorizations. Equation (3) is rewritten as $v = X\ N_D$. Let C_{32}^{II} be the 32-point DCT-II. Consequently, matrix N_D can be equivalently written as follows:

$$N_D = C_{32}^{II} \begin{bmatrix} A_{16\times16} & I_{16} & 0 & 0 \\ B_{16\times16} & 0 & I_{16} & D_{16\times16} \end{bmatrix}$$

where

$$A_{16\times16} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \bar{I}_{15} & \\ 0 & & & \end{bmatrix}$$

$$B_{16\times16} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

and

$$D_{16\times16} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & -\bar{I}_{15} & \\ 0 & & & \end{bmatrix}.$$

IV. PERFORMANCE

Here, we estimate the performance of the proposed methods by computational complexity. There are many fast algorithms for N-point DCT-II and DCT-III, where $\mathrm{N} = 2^{\mathrm{M}}$ and $\mathrm{M} > 0$ [4], [7]. MN/2 multiplications and $3\mathrm{MN}/2 - \mathrm{N} + 1$ additions are required for both DCT types using these fast algorithms [4]. In this way, 80 multiplications and 209 additions are required for a 32-point DCT-III or DCT-II, and 192 multiplications and 513 additions are required for a 64-point DCT-II. Next, the operations can be estimated by adding little complexity of the permutations. In AQMF, the input permutation (10) requires only zero multiplication and 31 additions; in SQMF, the output permutation (15) requires only zero multiplication and 31 additions (minuses). Hence, the total multiplications and additions for AQMF are 80 and 240; those for SQMF are 192 and 544. All the multiplications and additions required in QMF (AQMF+ SQMF) are 272 and 784. Compared with the original unoptimized computation-intensive operations, the multiplications and additions of the proposed methods are only 2.7% and 7.8%. Similarly, replacing SQMF with downsampled SQMF can estimate the complexity in the downsampled case. Table I summaries the reduction of computational complexity of the proposed QMF, and Table II summaries the reduction with downsampled SQMF. Moreover, the proposed methods can be further employed in combination with the IMDCT in the core AAC, which is also decomposed into a conventional DCT type and a permutation [7]. Hence, implementation of unified DCT for these three kinds of filterbanks in the HE AAC decoder can be easily accomplished.

TABLE II
REDUCTION OF COMPUTATIONAL COMPLEXITY IN THE PROPOSED QMF WITH DOWNSAMPLED SQMF

Filterbank	Operation	Original	Proposed			Proposed/
			Fast DCT	Permutation	Total	Original
AQMF	Mult	2048	80	0	80	3.9%
	Add	2016	209	31	240	11.9%
SQMF (32-channel)	Mult	2048	80	0	80	3.9%
	Add	1984	209	15	224	11.3%
Total	Mult	4096	160	0	160	3.9%
	Add	4000	418	46	464	11.6%

V. CONCLUSION

In this letter, we derive fast QMF in the low power SBR tools for MPEG HE AAC. Because both QMF are variants with respect to the subband filtering of the previous MPEG audio coding, fast decomposition is derived by modifying existing fast algorithms. The matrix operation in AQMF can be decomposed into a simple input permutation and a 32-point DCT-III. The matrix operation in SQMF can be decomposed into a 64-point DCT-II with a simple output permutation, and that in downsampled SQMF can be decomposed into a 32-point DCT-II with a simple output permutation. With the help of the fast algorithm for the standard type DCT, the computational complexity can be reduced up to 2.7% and 7.8% with respect to the original multiplications and additions. Therefore, it is easy to implement the various filterbanks in the HE AAC decoder with unified DCT computations.

REFERENCES

[1] *Information Technology—Coding of Audio-Visual Objects—Part 3: Audio, Amendment 1: Bandwidth Extension*, ISO/IEC 14496-3:2001/Amd. 1:2003, Nov. 2003.

[2] O. Shimada *et al.*, "A low power SBR algorithm for the MPEG-4 audio standard and its DSP implementation," in *Proc. AES 116th Conv.*, May 2004.

[3] K. R. Rao and P. C. Yip, *The Transform and Data Compression Handbook*. Boca Raton, FL: CRC, 2001.

[4] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 757–760, Mar. 1997.

[5] *Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to 1.5 Mbit/s, Part 3: Audio, International Standard IS 11172-3*, ISO/IEC JTC1/SC29 WG11, 1992.

[6] K. Konstantinides, "Fast subband filtering in MPEG audio coding," *IEEE Signal Process. Lett.*, vol. 1, no. 2, pp. 26–28, Feb. 1994.

[7] C. M. Liu and W. C. Lee, "A unified fast algorithm for cosine modulated filter banks in current audio coding standards," *J. Audio Eng. Soc.*, vol. 47, no. 12, pp. 1061–1075, Dec. 1999.