Direct N-Point DCT Computation From Three Adjacent N/3-Point DCT Coefficients

Soo-Chang Pei, *Fellow, IEEE*, and Meng-Ping Kao

Abstract—An efficient method for computing a length- N Discrete Cosine Transform (DCT) from three consecutive length- $N/3$ DCTs is proposed. This method differs from previous ones in that it reduces considerable 31–38% arithmetic operations and uses only length- $N/3$ DCTs instead of length- N DCTs. We also find its great applications in fractional scaling of a block DCT-based image by the factor of $1/2^\alpha 3^\beta$. This would be very useful in high-definition television (HDTV) standard, whose display size is usually 16:9. The comparison with conventional methods is provided in this paper.

Index Terms—Direct DCT domain computation, fast DCT, radix-3 DCT.

I. INTRODUCTION

SINCE the Discrete Cosine Transform (DCT) was first introduced in 1974, it has been widely used in various fields among digital signal processing (DSP). For example, it is the foundation of many prevailing image and video compression techniques, such as JPEGs, MPEGs, and H.26x. Moreover, most of the images and videos nowadays are stored in the DCT domain through the above techniques. The following question, however, arises: How can we directly manipulate or process such a compressed media stored in the DCT domain?

To avoid unnecessary computations in decompression and re-compression, a number of algorithms have been proposed to complete the operations directly in the DCT domain. Among the various operations, scaling of an image, i.e., interpolation or decimation, is probably the most common one we might encounter in relevant applications. For example, Park [4] has proposed an algorithm using the symmetric convolution property of DCT. Zhao [5] utilizes the relationship between linear transforms in spatial domain and those in DCT domain to develop an alternative approach.

However, as long as an ideal scaling operation is concerned, it is necessary to construct a long DCT sequence from several short DCT sequences or to decompose a long DCT sequence into several short DCT sequences in the compressed domain. For example, when the decimation operation is concerned, it is

Fig. 1. Schematic representation of the traditional approach.

straightforward to reserve the low-frequency components and discard the high-frequency ones from the long DCT sequence, which is comprised by several short, original DCTs. On the other hand, when the interpolation operation is concerned, we should first zero-pad a short DCT to its high-frequency band and then split the resulting long DCT to several short DCTs to complete the interpolation. Readers could refer to [4] for more details.

Since Skodras [1] has proposed a corresponding method for the case of two sequences, we now focus on the case of three consecutive sequences in this paper. Combining with Skodras's method, many DCT domain operations, such as fractional scaling by the factor of $1/2^\alpha 3^\beta$, are now realizable in a simple and efficient way.

II. CONVENTIONAL APPROACH

Let us readdress the problem as follows. Assume a length N sequence x_n is composed of three consecutive sequences with length $N/3$, i.e., $a_n = x_n$, $b_n = x_{n+(N/3)}$, and $c_n = x_{n+(2N/3)}$ for $n = 0, 1, \dots, N/3 - 1$, and A_k , B_k , and C_k are their length $N/3$ DCT coefficients, respectively. Given A_k , B_k , and C_k , how can we find an efficient way to compute the length N DCT coefficients X_k ?

The conventional way to solve this problem is depicted in Fig. 1. According to this scheme, three $N/3$ -point IDCTs are required, followed by a N -point DCT. Even if the fast N -point DCT structure using three $N/3$ -point DCTs [2] is applied to simplify the N -point DCT, however, the total computational complexity in Fig. 1 is still limited by at least six $N/3$ -point DCTs.

In order to modify the conventional approach, we propose a new structure based on DCT decomposition to avoid the N -point DCT computation. In this way, we use only four $N/3$ -point DCTs to find X_k , which is much more efficient than before.

Manuscript received March 24, 2004; revised June 14, 2004. This work was supported by the National Science Council, R.O.C., under Contracts NSC93-2219-E-002-004 and NSC93-2752-E-002-006-PAE. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Markus Pueschel.

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617 R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw; gyalp@yahoo.com.tw).

Digital Object Identifier 10.1109/LSP.2004.840868

Now, we go straight to investigating the key concept of this new structure. First of all, the normalized forward DCT-II of length- N sequence x_n is given as follows:

$$X_k = \sqrt{\frac{2}{N}} \varepsilon_k \sum_{n=0}^{N-1} x_n \cos \frac{(2n+1)k\pi}{2N}, \quad k = 0, 1, \dots, N-1 \quad (1)$$

and the corresponding inverse DCT is

$$x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \varepsilon_k X_k \cos \frac{(2n+1)k\pi}{2N}, \quad n = 0, 1, \dots, N-1 \quad (2)$$

where

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k = 0 \\ 1, & \text{for } k \neq 0 \end{cases}. \quad (3)$$

Note that $\varepsilon_{3k} = \varepsilon_k$ and $\varepsilon_{3k+1} = \varepsilon_{3k+2} = 1$. We now decompose X_k into three equal-length parts X_{3k} , $X_{3k+1} + X_{3k-1}$, and $X_{3k+2} + X_{3k-2}$, respectively, according to [2]. That is

$$\begin{aligned} X_{3k} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \varepsilon_k \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{a_n + b'_n + c_n\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ X_{3k+1} + X_{3k-1} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{(2a_n - b'_n - c_n) \cos \theta_n \\ &\quad + \sqrt{3}(b'_n - c_n) \sin \theta_n\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ X_{3k+2} + X_{3k-2} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{(2a_n - b'_n - c_n) \cos 2\theta_n \\ &\quad + \sqrt{3}(c_n - b'_n) \sin 2\theta_n\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{2 \cos \theta_n [(2a_n - b'_n - c_n) \cos \theta_n \\ &\quad + \sqrt{3}(c_n - b'_n) \sin \theta_n] \\ &\quad - (2a_n - b'_n - c_n)\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \end{aligned} \quad (4) \quad (5) \quad (6)$$

where

$$\theta_n = \frac{(2n+1)\pi}{2N}. \quad (7)$$

Note that in (5) and (6), when $k = 0$, X_{-1} and X_{-2} are defined in the same manner as that in (1). Moreover,

$b'_n = b_{(N/3)-1-n} = x_{(2N/3)-1-n}$ for $n = 0, 1, \dots, (N/3) - 1$, and $k = 0, 1, \dots, (N/3) - 1$. As a consequence

$$\begin{aligned} B'_k &= \text{DCT}_{\frac{N}{3}} \{b'_n\} \\ &= \text{DCT}_{\frac{N}{3}} \left\{ b_{\left(\frac{N}{3}\right)-1-n} \right\} \\ &= (-1)^k B_k, \quad \text{for } k = 0, 1, \dots, \left(\frac{N}{3}\right) - 1. \end{aligned} \quad (8)$$

The decomposition method addressed above is categorized as “decimation in frequency.” Its computational complexity is given by the following equations in both recursive and nonrecursive forms:

$$\begin{aligned} M_N^{\text{DCT}} &= 3M_{\frac{N}{3}}^{\text{DCT}} + N - 2 \\ &= \left(\frac{N}{9}\right) M_9^{\text{DCT}} + N \log_3 N - \left(\frac{19}{9}\right) N + 1 \\ A_N^{\text{DCT}} &= 3A_{\frac{N}{3}}^{\text{DCT}} + 3N - 5 \\ &= \left(\frac{N}{9}\right) A_9^{\text{DCT}} + 3N \log_3 N - \left(\frac{113}{18}\right) N + \frac{5}{2} \\ &\quad \text{for } N = 3^m, \quad m \geq 2. \end{aligned} \quad (9) \quad (10)$$

There are, of course, other methods of decomposition, such as “decimation in time” or “mix decimation.” For example, Bi [6] has proposed an algorithm using the mix decimation method in which it has a minimum nine-point DCT complexity with $M_9^{\text{DCT}} = 8$ and $A_9^{\text{DCT}} = 34$, where some operations on trivial factors are precluded. For details, please refer to [6]. In general, Bi’s method has the computational complexities as follows:

$$\begin{aligned} M_N^{\text{DCT}} &= 3M_{\frac{N}{3}}^{\text{DCT}} + \left(\frac{4N}{3}\right) - 3 \\ &= \left(\frac{N}{9}\right) M_9^{\text{DCT}} + \left(\frac{4}{3}\right) N \log_3 N \\ &\quad - \left(\frac{17}{6}\right) N + \frac{3}{2} \\ A_N^{\text{DCT}} &= 3A_{\frac{N}{3}}^{\text{DCT}} + \left(\frac{8N}{3}\right) - 4 \\ &= \left(\frac{N}{9}\right) A_9^{\text{DCT}} + \left(\frac{8}{3}\right) N \log_3 N \\ &\quad - \left(\frac{50}{9}\right) N + 2 \quad \text{for } N = 3^m, \quad m \geq 2. \end{aligned} \quad (11) \quad (12)$$

We see from (9) and (11) that Chan’s method is more superior to Bi’s in that its multiplications are about $(N/3) \log_3 N$ less when the same nine-point DCT module are utilized for $N \geq 9$. As a consequence, for practical implementation, we could combine Bi’s optimized nine-point DCT module [6] together with Chan’s N -point DCT decomposition for $N \geq 9$.

III. PROPOSED APPROACH

Since the DCT coefficients A_k , B_k , and C_k are precomputed and saved in the compressed domain, the main difference between the proposed method and the radix-3 DCT algorithm [2] is to merge three length $N/3$ DCT coefficients A_k , B_k , and C_k into a length N DCT X_k very efficiently.

A. X_{3k} Coefficients Calculation

From (4)

$$\begin{aligned} X_{3k} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \varepsilon_k \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{a_n + b_n' + c_n\} \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ &= \sqrt{\frac{1}{3}} \{A_k + B_k' + C_k\}, \quad k = 0, 1, \dots, \frac{N}{3} - 1. \end{aligned} \quad (13)$$

B. $X_{3k+1} + X_{3k-1}$ Coefficients Calculation

From (5)

$$\begin{aligned} X_{3k+1} + X_{3k-1} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{(2a_n - b_n' - c_n) \cos \theta_n \\ &\quad + \sqrt{3}(b_n' - c_n) \sin \theta_n\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ &= \sqrt{\frac{1}{3}} \frac{1}{\varepsilon_k} \text{DCT}_{\frac{N}{3}} \left\{ (2a_n - d_n) \cos \theta_n + \sqrt{3}e_n \sin \theta_n \right\} \\ &= \sqrt{\frac{1}{3}} \frac{1}{\varepsilon_k} \text{DCT}_{\frac{N}{3}} \left\{ \text{IDCT}_{\frac{N}{3}} \{2A_k - D_k\} \cos \theta_n \right. \\ &\quad \left. + \text{IDCT}_{\frac{N}{3}} \{E_k\} \sqrt{3} \sin \theta_n \right\} \end{aligned} \quad (14)$$

where

$$d_n = b_n' + c_n, \quad e_n = b_n' - c_n. \quad (15)$$

As a consequence

$$D_k = (-1)^k B_k + C_k, \quad E_k = (-1)^k B_k - C_k. \quad (16)$$

C. $X_{3k+2} + X_{3k-2}$ Coefficients Calculation

From (6)

$$\begin{aligned} X_{3k+2} + X_{3k-2} &= \sqrt{\frac{1}{3}} \sqrt{\frac{2}{\frac{N}{3}}} \sum_{n=0}^{\left(\frac{N}{3}\right)-1} \{2 \cos \theta_n [(2a_n - b_n' - c_n) \cos \theta_n \\ &\quad + \sqrt{3}(c_n - b_n') \sin \theta_n] \\ &\quad - (2a_n - b_n' - c_n)\} \\ &\quad \times \cos \frac{(2n+1)k\pi}{2\left(\frac{N}{3}\right)} \\ &= \sqrt{\frac{1}{3}} \frac{1}{\varepsilon_k} \left\{ \text{DCT}_{\frac{N}{3}} \left\{ 2 \cos \theta_n \left[\text{IDCT}_{\frac{N}{3}} \{2A_k - D_k\} \cos \theta_n \right. \right. \right. \\ &\quad \left. \left. - \text{IDCT}_{\frac{N}{3}} \{E_k\} \sqrt{3} \sin \theta_n \right\} \right. \\ &\quad \left. - (2A_k - D_k) \right\}. \end{aligned} \quad (17)$$

Fig. 2. Flow graph of the proposed algorithm.

Fig. 3. Flow graph of X_{3k+1} and X_{3k+2} .

According to the derivation above, we can construct the entire structure in Fig. 2.

D. X_{3k+1} and X_{3k+2} Coefficients Calculation

Since we have obtained $X_{3k+1} + X_{3k-1}$ and $X_{3k+2} + X_{3k-2}$, the next question is how to calculate X_{3k+1} and X_{3k+2} from them. According to (1), X_k is an even function of k because both ε_k and $\cos((2n+1)k\pi/2N)$ are even functions of k . Therefore, for $k = 0$

$$X_{3k+1} + X_{3k-1}|_{k=0} = X_1 + X_{-1} = 2X_1 \quad (18)$$

$$X_{3k+2} + X_{3k-2}|_{k=0} = X_2 + X_{-2} = 2X_2. \quad (19)$$

Using sequences $X_{3k+1} + X_{3k-1}$ and $X_{3k+2} + X_{3k-2}$ together with the initial conditions X_1 and X_2 , we can derive X_{3k+1} and X_{3k+2} in an iterative approach. The flow graph is illustrated in Fig. 3, where $N/3$ is chosen to be 8, for example.

IV. COMPUTATIONAL COMPLEXITY

From the proposed structures in Figs. 2 and 3, the computation of X_{3k} , $X_{3k+1} + X_{3k-1}$, and $X_{3k+2} + X_{3k-2}$ requires seven length- $N/3$ vector additions, three length- $N/3$ vector multiplications, and 4 length- $N/3$ DCTs, assuming that the

DCT and the indiscrete DCT (IDCT) are of the same complexity. Furthermore, when $n = [(N/3) - 1]/2$, $\theta_n = \pi/6$, $\cos \theta_n = \sqrt{3}/2$, $\sin \theta_n = 1/2$, and (14) and (17) become

$$\begin{aligned} & X_{3k+1} + X_{3k-1} \\ &= \sqrt{\frac{1}{3}} \frac{1}{\varepsilon_k} \text{DCT}_{\frac{N}{3}} \left\{ \frac{\sqrt{3}}{2} \left[\text{IDCT}_{\frac{N}{3}} \{2A_k - D_k\} \right. \right. \\ &\quad \left. \left. + \text{IDCT}_{\frac{N}{3}} \{E_k\} \right] \right\} \\ & X_{3k+2} + X_{3k-2} \\ &= \sqrt{\frac{1}{3}} \frac{1}{\varepsilon_k} \left\{ \text{DCT}_{\frac{N}{3}} \left\{ \frac{3}{2} \left[\text{IDCT}_{\frac{N}{3}} \{2A_k - D_k\} \right. \right. \right. \\ &\quad \left. \left. - \text{IDCT}_{\frac{N}{3}} \{E_k\} \right] \right\} \right. \\ &\quad \left. - (2A_k - D_k) \right\} \end{aligned}$$

Therefore, we can save two more multiplications. On the other hand, the computation of X_{3k+1} and X_{3k+2} in Fig. 3 requires 2 length- $(N/3 - 1)$ vector additions since multiplication by $1/2$ can be implemented using a bit-wise shift operator.

For the purpose of simplicity, we choose $N = 2^m \times 3$ in the following comparisons such that the fast DCT could be used to implement length- $N/3$ DCTs, where m is a positive integer. According to [3], the arithmetic complexity of a length- $N/3 = 2^m$ fast DCT is

$$M_{\frac{N}{3}}^{\text{DCT}} = M_{2^m}^{\text{DCT}} = \left(\frac{m}{2}\right) \times \left(\frac{N}{3}\right) \quad (20)$$

$$A_{\frac{N}{3}}^{\text{DCT}} = A_{2^m}^{\text{DCT}} = \left(\frac{3m}{2}\right) \times \left(\frac{N}{3}\right) - \left(\frac{N}{3}\right) + 1. \quad (21)$$

The total numbers of multiplications M_N^P and additions A_N^P of the proposed structure in Fig. 2, then, become

$$\begin{aligned} M_N^P &= 3 \times \left(\frac{N}{3}\right) - 2 + 4 \times M_{\frac{N}{3}}^{\text{DCT}} \\ &= N \left(\frac{2m}{3} + 1\right) - 2 \end{aligned} \quad (22)$$

$$\begin{aligned} A_N^P &= 7 \times \left(\frac{N}{3}\right) + 2 \times \left(\frac{N}{3} - 1\right) + 4 \times A_{\frac{N}{3}}^{\text{DCT}} \\ &= N \left(2m + \frac{5}{3}\right) + 2. \end{aligned} \quad (23)$$

As for the conventional approach in Fig. 1, the complexity is equal to three length- $N/3$ DCTs and one length- N DCT. The total numbers of multiplications M_N^T and additions A_N^T of the conventional structure, then, become [2]

$$\begin{aligned} M_N^T &= 3 \times M_{\frac{N}{3}}^{\text{DCT}} + M_N^{\text{DCT}} = 6 \times M_{\frac{N}{3}}^{\text{DCT}} + N - 2 \\ &= N(m + 1) - 2 \end{aligned} \quad (24)$$

$$\begin{aligned} A_N^T &= 3 \times A_{\frac{N}{3}}^{\text{DCT}} + A_N^{\text{DCT}} = 6 \times A_{\frac{N}{3}}^{\text{DCT}} + 3N - 5 \\ &= N(3m + 1) + 1. \end{aligned} \quad (25)$$

We summarize the computational complexity of these two approaches in Table I. Note that the computational saving percentage increases with N . Specifically, when $N \rightarrow \infty$, the asymptotic saving percentages approach 50% for both multiplications and additions. Moreover, for $N = 8 \times 3$, which is the standard block dimension for image and video coding, about 31–38% operations can be saved.

TABLE I
COMPUTATIONAL COMPLEXITIES OF CONVENTIONAL AND PROPOSED METHODS
DESCRIBED IN (22) TO (25)

N	Conventional [2],[3]			Proposed			Saved		
	Mul	Add	Mul+Add	Mul	Add	Mul+Add	Mul	Add	Mul+Add
$3 \times 4 = 12$	34	85	119	26	70	96	31%	21%	24%
$3 \times 8 = 24$	94	241	335	70	186	256	34%	30%	31%
$3 \times 16 = 48$	238	625	863	174	466	640	37%	34%	35%
$3 \times 32 = 96$	574	1537	2111	413	1122	1535	39%	37%	38%

For the application of block-DCT-based image down-sampling, since the higher frequency DCT coefficients will virtually be discarded, there is no need to compute them in Figs. 2 and 3. Let us take the case of down-sampling by 3, for example. We need only to compute the first $1/3$ terms. Therefore, two vector additions, i.e., the left-top and right-bottom additions in Fig. 2, can be replaced by length- $\lceil N/9 \rceil$ vector additions. The two length- $N/3$ forward DCTs could be replaced by length- $N/3$ pruned DCTs (PDCT) of first $\lceil N/9 \rceil$ terms. Moreover, the additions in Fig. 3 are also reduced to two length- $(\lceil N/9 \rceil - 1)$ vector additions. The complexity of PDCT can be computed according to [8]. In this way, a huge amount of computations could be saved.

V. CONCLUSION

We propose an efficient approach to compute a length- N DCT given three consecutive length- $N/3$ DCTs. The operations could be saved from 31% up to 38% when the block size is chosen from 8 to 32, and the larger the block size, the more operations we can save. Another advantage is that only length- $N/3$ DCTs and IDCTs are required, instead of length- N DCTs. Combining with Skodras's method, scaling a DCT-based image by the factor of $1/2^{\alpha\beta}$ is easily realizable by cascading α for Skodras's structures and β for the authors' structures. This provides more flexibility and efficiency in the application of fractional scaling, especially in the rising standard of HDTV, whose screen size is usually set to be 16:9.

REFERENCES

- [1] A. N. Skodras, "Direct transform to transform computation," *IEEE Signal Process. Lett.*, vol. 6, no. 8, pp. 202–204, Aug. 1999.
- [2] Y.-H. Chan and W.-C. Siu, "Fast radix-3/6 algorithms for the realization of the discrete cosine transform," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, May 3–6, 1992, pp. 153–156.
- [3] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 757–760, Mar. 1997.
- [4] H. W. Park, Y. S. Park, and S.-K. Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1016–1034, Sep. 2003.
- [5] Y. Zhao, M. S. Kankanhall, and T.-S. Chua, "Fractional scaling of image and video in DCT domain," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Barcelona, Spain, Sep. 14–17, 2003, pp. 185–188.
- [6] G. Bi and L. W. Yu, "DCT algorithms for composite sequence lengths," *IEEE Trans. Signal Process.*, vol. 46, no. 3, pp. 554–562, Mar. 1998.
- [7] Y.-H. Chan and W.-C. Siu, "Mixed-radix discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 41, no. 11, pp. 3157–3161, Nov. 1993.
- [8] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1833–1837, Jul. 1994.