A New Algorithm to Compute the Discrete Cosine Transform

BYEONG GI LEE

Abstract— **A new algorithm is introduced for the 2^m-point discrete cosine transform. This algorithm reduces the number of multiplications to about half of those required by the existing efficient algorithms, and it makes the system simpler.**

I. Introduction

During the past decade, the discrete cosine transform (DCT) [1] has found applications in speech and image processing. Various fast algorithms have been introduced for reducing the number of multiplications involved in the transform [2]-[6]. In this correspondence we propose an additional algorithm which not only reduces the number of multiplications but also has a simpler structure. We refer to this algorithm as the FCT (fast cosine transform), since it is similar to the FFT (fast Fourier transform). The number of real multiplications it requires is about half that required by the existing efficient algorithms.

II. Algorithm Derivation

We denote the DCT of the data sequence $x(k)$, $k = 0, 1, \cdots, N-1$, by $X(n)$, $n = 0, 1, \cdots, N-1$. Then we have [1]

$$\sum_{n=0}^{N-1} e(n)X(n)cos[\pi(2k+1)n/2N] \qquad k = 0, 1, \cdots, N-1 \tag{1}$$

and

$$X(n) = \frac{2}{N}e(n)\sum_{k=0}^{N-1} x(k)cos[\pi(2k+1)n/2N] \qquad n = 0, 1, \cdots, N-1 \tag{2}$$

where

$$e(n) = \begin{cases} 1/\sqrt{2}, & if \quad n = 0, \\ 1, & otherwise. \end{cases} \tag{3}$$

We consider (1), which is the inverse DCT (IDCT), and define C such that

$$C_{2N}^{(2k+1)n} = cos[\pi(2k+1)n/2N]. \tag{4}$$

Then the N-point IDCT becomes

$$x(k) = \sum_{k=0}^{N-1} \hat{X}(n)C_{2N}^{(2k+1)n}, \quad k = 0, 1, \cdots, N-1 \tag{5}$$

where

$$\hat{X}(n) = e(n)X(n) \tag{6}$$

Manuscript received August 15, 1983; revised February 29, 1984.
The author is with the Granger Associates, Santa Clara, CA 95051.

Decomposing $x(k)$ into even and odd indexes of n (assuming that N is even), we can rewrite (5) as

$$x(k) = g(k) + h'(k) \tag{7a}$$

$$x(N-1-k) = g(k) - h'(k), \qquad k = 0, 1, \cdots, N/2-1 \tag{7b}$$

where

$$g(k) = \sum_{n=0}^{N/2-1} \hat{X}(2n)C_{2N}^{(2k+1)2n} \tag{8a}$$

$$h'(k) = \sum_{n=0}^{N/2-1} \hat{X}(2n+1)C_{2N}^{(2k+1)(2n+1)} \tag{8b}$$

Clearly, $g(k)$, $k = 0, 1, \cdots, N/2-1$, forms an $N/2$-point IDCT, since

$$C_{2N}^{(2k+1)2n} = C_N^{(2k+1)n} = C_{2(N/2)}^{(2k+1)n} \tag{9}$$

We rewrite $h'(k)$ in the form

$$h'(k) = \sum_{n=0}^{N/2-1} \hat{X}'(2n+1)C_{2(N/2)}^{(2k+1)n} \tag{10}$$

which is another N/2-point IDCT. Since

$$2C_{2N}^{(2k+1)}C_{2N}^{(2k+1)(2n+1)} = C_{2N}^{(2k+1)2n} + C_{2N}^{(2k+1)2(n+1)} \tag{11}$$

we have

$$2C_{2N}^{(2k+1)}h'(k) = \sum_{n=0}^{N/2-1} \hat{X}(2n+1)C_{2N}^{(2k+1)2n} + \sum_{n=0}^{N/2-1} \hat{X}(2n+1)C_{2N}^{(2k+1)2(n+1)}. \tag{12}$$

So, if we define

$$\hat{X}(2n-1)|_{n=0} = 0 \tag{13}$$

then

$$\sum_{n=0}^{N/2-1} \hat{X}(2n+1)C_{2N}^{(2k+1)2(n+1)} \qquad \sum_{n=0}^{N/2-1} \hat{X}(2n-1)C_{2N}^{(2k+1)2n} \tag{14}$$

because

$$C_{2N}^{(2k+1)2(N/2)} = C_2^{(2k+1)} = 0. \tag{15}$$

Thus (12) can be rewritten as

$$2C_{2N}^{(2k+1)}h'(k) = \sum_{n=0}^{N/2-1} (\hat{X}(2n+1) + \hat{X}(2n-1))C_{2N}^{(2k+1)2n} \tag{16}$$

which has the form of (10). Now we define

$$G(n) = \hat{X}(2n), \tag{17a}$$

$$H(n) = \hat{X}(2n+1) + \hat{X}(2n-1)$$
$$n = 0, 1, \cdots, N/2-1, \tag{17b}$$

and

$$g(k) = \sum_{n=0}^{N/2-1} G(n) C_{2(N/2)}^{(2k+1)n}, \tag{18a}$$

$$h(k) = \sum_{n=0}^{N/2-1} H(n) C_{2(N/2)}^{(2k+1)n},$$
$$k = 0, 1, \cdots, N/2-1. \tag{18b}$$

Then (7), (8), and (16)-(18) finally yield

$$x(k) = g(k) + (1/(2C_{2N}^{(2k+1)}))h(k), \tag{19a}$$

$$x(N-1-k) = g(k) - (1/(2C_{2N}^{(2k+1)}))h(k),$$
$$k = 0, 1, \cdots, N/2-1. \tag{19b}$$

Therefore, we have decomposed the N-point IDCT in (5) into the sum of two N/2-point IDCT's in (18). By repeating this process, we can decompose the IDCT further.

We can also decompose the DCT in a similar manner. Alternatively, the DCT can be obtained by "transposing" the IDCT-i.e., reversing the direction of the arrows in the flow graph of IDCT, since the DCT is an *orthogonal* transform.

III. Example

With N=8, (17)-(19) yield

$$G(n) = \hat{X}(2n), \tag{20a}$$

$$H(n) = \hat{X}(2n+1) + \hat{X}(2n-1), \ n = 0, 1, 2, 3 \tag{20b}$$

and

$$g(k) = \sum_{n=0}^{3} G(n) C_8^{(2k+1)n} \tag{21a}$$

$$h(k) = \sum_{n=0}^{3} H(n) C_8^{(2k+1)n}, \tag{21b}$$

$$x(k) = g(k) + (1/(2C_{16}^{2k+1}))h(k), \tag{22a}$$

$$x(7-k) = g(k) - (1/(2C_{16}^{2k+1}))h(k), \quad k = 0, 1, 2, 3 \tag{22b}$$

Equations (20) and (22) respectively form the first and the last stages of the flow graph in Fig.1. By repeating the above steps on (21), we obtain the FCT flow graph for an eight-point IDCT as shown in Fig.1.

IV. Concluding Remarks

It follows from Fig.1 that the flow graphs of the FCT and FFT are similar. The number of real multiplications thus appears to be $(N/2)log_2N$ for an N-point FCT with N=2^m, which is about half the number required by existing efficient algorithms. The number of additions, however, is slightly higher and given by $(3N/2)log_2N-N+1$. See Table I for a comparison with the algorithm in [4].

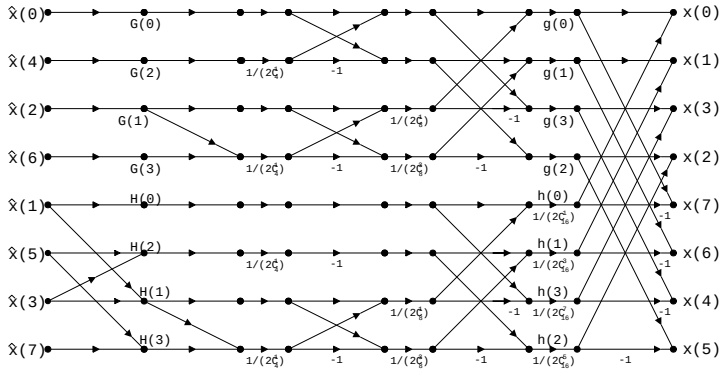

Fig. 1.

TABLE I

m	N	Number of Multiplications		Number of Additions	
		REF[4]	FCT	REF[4]	FCT
3	8	16	12	26	29
4	16	44	32	74	81
5	32	116	80	194	209
6	64	292	192	482	513
7	128	708	448	1154	1217
8	256	1668	1024	2690	2817
9	512	3844	2304	6146	6401
10	1024	8708	5120	13826	14337
11	2048	19460	11264	30722	31745
12	4096	43012	24576	67586	69633

If Fig.1 we also note that the input sequence $\hat{X}(n)$ is in bit-reversed order. The order of the output sequence $x(k)$ is generated in the following manner: starting with the set (0,1), form a set by adding the prefix "0" to each element, and then obtain the rest of the elements by complementing the existing ones. This process results in the set (00,01,11,10), and by repeating it we obtain (000, 001, 011, 010, 111, 110, 100, 101). Thus, we have the output sequence $x(0)$, $x(1)$, $x(3)$, $x(2)$, $x(7)$, $x(6)$, $x(4)$, $x(5)$ for the case N=8; see Fig.1.

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-94, Jan. 1974.

[2] M. R. Haralick, "A storage efficient way to implement the discrete cosine transform," *IEEE Trans. Comput.*, vol. C-25, pp. 764-765, July 1976.

[3] B. D. Tseng and W. C. Miller, "On computing the discrete cosine transform," *IEEE Trans. Comput.*, vol. C-27, pp. 966-968, Oct. 1978.

[4] W. H. Chen, C. H. Smith, and S. C. Fralick,"A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, Sept. 1977.

[5] M. J. Narasimha and A. M. Peterson, "On the computation of dis-

crete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934-936, June 1978.

[6] J. Makhoul, "A fast cosine transform in one and two dimensions," *IEEE Trans. Acoust., Speech, Signal Proce− ssing*, vol. ASSP-28,pp. 27-34, Feb. 1980.