A New Algorithm for the Discrete Cosine Transform of Arbitrary Number of Points

MEGHANAD D. WAGH AND H. GANESH

Abstract—**An alternate algorithm to compute the discrete cosine transform (DCT) of sequences of arbitrary number of points is proposed. The algorithm consists of partitioning the DCT kernel into submatrices which by proper row and column shuffling and negations can be made equivalent to the group tables (or parts of them) of appropriate Abelian groups. The computations pertaining to the submatrices can be carried out using multidimensional cyclic convolutions. Algorithms are also developed to perform the computations associated with the submatrices that are parts of larger group tables. The new algorithms are more versatile and generally better in terms of the computational complexity in comparison with the existing algorithms.**

Index Terms—**Computational complexity, cyclic convolution, discrete cosine transform.**

I. Introduction

THE discrete cosine transform (DCT) defined by Ahmed *et al.* [1] in 1974 has recently found a number of applications in the area of digital image processing [2]–[8]. The DCT approximates the optimal Karhunen-Loeve transform better than most other orthogonal transforms including the discrete Fourier transform (DFT) [1], [8], [9]. However, the utility of a transform is governed not only by its optimality but also by its computational simplicity and therefore there have been several attempts in recent years to find efficient algorithms for the DCT [10]–[13].

The DCT of an N-point sequence $y(i)$, $i = 0, 1, \cdots, N-1$ is defined as [1]

$$Y(0) = \frac{\sqrt{2}}{N} \sum_{i=0}^{N-1} y(i),$$

$$Y(j) = \sum_{i=0}^{N-1} M(j, i)\, y(i), \qquad j = 1, 2, \cdots, N-1, \quad (1)$$

where $M(j, i) = (2/N) \cos(j(2i + 1)\pi/2N)$.

We restrict ourselves to real data sequences. Furthermore, since a scaling of every transform component by a constant factor does not alter its applicability, we ignore the factor $(2/N)$ in $M(j, i)$ and compute $Y(0)$ as $1/\sqrt{2}\,(y(0) + y(1) + \cdots + y(N-1))$.

Among all the methods available to compute (1), those proposed in [11]–[13] appear to be best for various sequence lengths. When N is a power of 2, the algorithm [11] computes DCT directly in $N \log_2 N - 3N/2 + 4$ real multiplications and $3N/2\,(\log_2 N - 1) + 2$ real additions. The procedures of [12] and [13], on the other hand, compute DCT through a DFT of real data. It is assumed here that two real L-point DFT's are computed through one complex L-point DFT and L complex additions [14], and that the complex DFT is implemented through the WFTA [15]. Each real L-point DFT then requires $M_W(L)$ real multiplications and $A_W(L) + L$ real additions where $M_W(L)$ and $A_W(L)$ denote the number of multiplications and additions, respectively, in Winograd's algorithm [15] of length L. The DCT algorithm of [12] applicable for even N, calls for a real N point DFT followed by $(N/2 - 1)$ complex multiplications, thus requiring a total of $M_W(N) + 2N - 4$ real multiplications and $A_W(N) + 2N - 2$ real additions. The algorithm of [13], applicable for arbitrary N, requires the evaluation of the real part of a real $2N$-point DFT followed by N real multiplications. It thus involves $M_W(2N) + N$ real multiplications and $A_W(2N) + N$ real additions.

In this paper, a new DCT algorithm based on the short-length cyclic convolutions is proposed. This algorithm is generally more efficient than the algorithms mentioned earlier and is applicable to sequences with arbitrary number of points.

Section II of this paper presents the required group theoretic fundamentals and briefly reviews the earlier work in this direction. The DCT algorithms are constructed in Section IV using the theoretical background developed in Section III. The construction is illustrated throughout with an algorithm of a 10-point DCT. Finally, the computational complexity of the algorithm developed is analyzed and compared with that of the conventional algorithms [11]–[13] in Section V.

II. Group Tables and Transform Computation

In this paper we will be concerned with Abelian groups of positive integers relatively prime to N and less than N under the operation of multiplication modulo N for integer N's. The group and the group operation will be denoted by $A(N)$ and $\oplus$. The inverse of g will be denoted by $\ominus g$, ng will mean $g \oplus g \oplus \cdots n$ times, and $h \ominus g = h \oplus (\ominus g)$, $g, h \in G$.

Example 1: The integers {1, 3, 7, 9, 11, 13, 17, 19, 21, 23, 27, 29, 31, 33, 37, 39} from a group $A(40)$ under the operation of multiplication modulo 40. In this group, $17 \oplus 37 = 629$ mod $40 = 29$, $9 \oplus 11 = 99$ mod $40 = 19$, etc. Similarly, $\ominus 13 = 37$ because $37 \oplus 13 = 1$, $\ominus 21 = 21$ because $21 \oplus 21 = 1$, etc.//

The structure of $A(N)$ is fully determined by the value of N. In general, one has the following results (see [16, Theorems III.2.m and III.2.p]).

Manuscript received May 8, 1978; revised June 21, 1979.

M. D. Wagh was with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, India. He is now with the Department of Electrical Engineering, Concordia University, Montreal, P.Q., Canada.

H. Ganesh was with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, India. He is now with the Department of Electrical Engineering, Calicut Regional Engineering College, Calicut, India.

0018-9340/80/0400-0269$00.75

1) $A(r_1 \cdot r_2) \simeq A(r_1) \times A(r_2)$ when gcd $(r_1, r_2) = 1$, where $\simeq$ denotes isomorphism and $\times$, the direct product of groups.

2) $A(p^n) \simeq C_{(p-1)p^{n-1}}$ when p is an odd prime and C_R denotes a cyclic group of order R.

3) $A(2^n) \simeq C_2 \times C_{2^{n-2}}$ if $n \geq 3$, $A(2) = \{1\}$ and $A(2^2) = C_2$.

Example 1 (Cont'd.): $A(40) \simeq A(8) \times A(5) \simeq C_2 \times C_2 \times C_4$. //

Furthermore, if $A(N) \simeq C_{n_1} \times C_{n_2} \times \cdots \times C_{n_r}$ it is obvious that one can find subgroups in $A(N)$ that are isomorphic to $C_{n_1}, C_{n_2}, \cdots, C_{n_r}$ such that the group $A(N)$ is an internal direct product of these subgroups. Any element of $A(N)$ can then be expressed uniquely as a sum of the elements of these subgroups. The subgroups then are called the splitting subgroups.

Example 1 (Cont'd.): $A(40) = \{1, 19\} \times \{1, 21\} \times \{1, 3, 9, 27\}$ where it is easy to verify that the three subgroups are cyclic groups of orders 2, 2, and 4,

Note: $19 \oplus 19 = 1, 21 \oplus 21 = 1, 3 \oplus 3 = 9, 9 \oplus 3 = 27, 27 \oplus 3 = 1$ and any $h \in A(40)$ can be obtained as $h = g_1 \oplus g_2 \oplus g_3 = g_1g_2g_3$ mod 40 where $g_1 \in \{1, 19\}, g_2 \in \{1, 21\}$ and $g_3 \in \{1, 3, 9, 27\}$. For example, $17 = 19 \oplus 1 \oplus 3, 31 = 19 \oplus 21 \oplus 9, 21 = 1 \oplus 21 \oplus 1$, etc. //

One can index a sequence $U(i)$ of length $N = n_1 \cdot n_2 \cdots n_r$ with reference to any group $G = C_{n_1} \times C_{n_2} \times \cdots \times C_{n_r}$. Let a_t denote the generator of C_{n_t}, i.e., $\langle a_t \rangle = C_{n_t}$, $1 \leq t \leq r$. Then a sequence component may be interchangeably referred to as $u(i), i \in \{0, 1, \cdots, N-1\}$ or $u(g)$ $g \in G$ where for some $i_1, i_2, \cdots, i_r$, $(0 \leq i_t \leq n_t - 1)$, $g = i_1a_1 \oplus i_2a_2 \oplus \cdots \oplus i_ra_r$ and i has the unique representation $i = i_1n_2n_3 \cdots n_r + i_2n_3n_4 \cdots n_r + \cdots + i_{r-1}n_r + i_r$.

For example, a sequence $u(i)$ of length 18 will be indexed with reference to the group $C_3 \times C_3 \times C_2 = \langle a \rangle \times \langle b \rangle \times \langle c \rangle$ as, $u(0), u(c), u(b), u(b \oplus c), u(2b), u(2b \oplus c), u(a), u(a \oplus c), u(a \oplus b), u(a \oplus b \oplus c)$ $u(a \oplus 2b), \cdots, u(2a \oplus 2b \oplus c)$.

We now define a generalized convolution with respect to an Abelian group G [17]. Index the sequences u and v by the elements of G. Then their convolved sequence w is given by

$$w(h) = \sum_{g \in G} u(g)v(h \ominus g).$$

It has been shown earlier [18] that this convolution with respect to $G \simeq C_{n_1} \times C_{n_2} \times \cdots \times C_{n_r}$ can be computed through $W = U * V$ where U, V, and W are r-dimensional arrays defined as

$$U(i_1, i_2, \cdots, i_r) = u(i_1a_1 \oplus i_2a_2 \oplus \cdots \oplus i_ra_r),$$

$$V(i_1, i_2, \cdots, i_r) = v(i_1a_1 \oplus i_2a_2 \oplus \cdots \oplus i_ra_r),$$

$$W(i_1, i_2, \cdots, i_r) = w(i_1a_1 \oplus i_2a_2 \oplus \cdots \oplus i_ra_r), \quad (3)$$

$0 \leq i_t \leq n_t - 1$, $1 \leq t \leq r$ and $*$ denotes an r-dimensional cyclic convolution. Note that each of U, V, and W has $|G|$ points. Using Agarwal-Cooley algorithms [19] this convolution thus can be evaluated in $M_1 \cdot M_2 \cdots M_r$ multiplications where M_t is the number of multiplications required for a cyclic convolution of length n_t. Obviously, when G is a cyclic group, the generalized convolution is identical to the usual cyclic convolution. Convolution with respect to G can sometimes be used to multiply a matrix and column vector as the following theorem of [18] shows.

Theorem 1: Consider the computation of $Y(j) = \sum_{i \in A} M(j, i)y(i), j \in B$. If there exists a group G and the functions $\psi_1: G \to B$, $\psi_2: G \to A$; $\delta_1, \delta_2: G \to \{1, -1\}$ and $f: G \to \mathcal{C}$, the set of complex numbers, such that

i) ψ_1 and ψ_2 are one-one and onto,

ii) $M(\psi_1 h, \psi_2 g) = \delta_1(h)\delta_2(g)f(h \oplus g)$, $\forall\, h, g \in G$,

then $Y(\psi_1 h) = \delta_1(h)w(h)$, $h \in G$ where w is the convolution with respect to G of the sequences u and v defined as $u(g) = f(g)$, $v(g) = \delta_2(\ominus g)y(\psi_2(\ominus g))$, $g \in G$.

Proof: Refer to [18]. //

III. Generalized Convolution of Sequences with Redundancies

Sometimes it might not be possible to find G, ψ_1, ψ_2, etc., satisfying the conditions of Theorem 1. Fortunately, in DCT computation, whenever this happens, index sets A and B can be extended to find a suitable group and the functions. Theorem 1 can then be used to compute the DCT as a convolution over a larger group G. This is proved in Theorem 2 of this section. The extensions of sequences however, introduce redundancies which may be exploited to reduce the computational complexity as shown in Theorems 3 and 4.

Consider a group G containing an order 2 element α. The set of coset representatives of $\{1, \alpha\}$ in G will be denoted by G_α.

Example 2: $A(50) = \{1, 3, 9, 27, 31, 43, 29, 37, 11, 33, 49, 47, 41, 23, 19, 7, 21, 13, 39, 17\}$ and contains $\alpha = 49$. Then $G_\alpha = \{1, 3, 9, 27, 31, 43, 29, 37, 11, 33\}$. (Note that G_α is *not* a group.) //

Theorem 2 (Extension of Theorem 1): If a group G and functions $\psi_1: G_\alpha \to B$, $\psi_2: G_\alpha \to A$; $\delta_1, \delta_2: G_\alpha \to \{1, -1\}$ and $f: G \to \mathcal{C}$ exist such that,

i) ψ_1, ψ_2 are one-one and onto,

ii) $f(g \oplus \alpha) = f(g)$, $g \in G$, and

iii) $M(\psi_1 h, \psi_2 g) = \delta_1(h)\delta_2(g)f(h \oplus g)$, $\forall\, h, g \in G_\alpha$, then, $Y(\psi_1 h) = \delta_1(h)w(h)$, $h \in G_\alpha$, where w is the convolution with respect to G of the sequences u and v defined as

$$u(g) = f(g)/2,\ g \in G \text{ and } v(g) = \delta_2(g')y(\psi_2 g'),\ g \in G$$

where

$$g' = \ominus g \text{ if } \ominus g \in G_\alpha,$$

$$= \alpha \ominus g \text{ if } \ominus g \notin G_\alpha.$$

Proof: Extend the functions ψ_1, ψ_2, δ_1, and δ_2 to G as, $\psi_1(g \oplus \alpha) = \psi_1 g$, $\psi_2(g \oplus \alpha) = \psi_2 g$, $\delta_1(g \oplus \alpha) = \delta_1(g)$ and $\delta_2(g \oplus \alpha) = \delta_2(g)$, $g \in G_\alpha$. Then index sets B and A get extended to $B \cup B'$ and $A \cup A'$, respectively, where

$$B = \{\psi_1 g | g \in G_\alpha\},\ B' = \{\psi_1 g | g \in G, g \notin G_\alpha\}$$

$$A = \{\psi_2 g | g \in G_\alpha\},\ A' = \{\psi_2 g | g \in G, g \notin G_\alpha\}.$$

These extended index sets and functions satisfy the requirements of Theorem 1 giving the stated result. //

Often, one comes across situations in which conditions i) and iii) of Theorem 2 are satisfied and in place of ii) one has

ii)′ $f(g \oplus \alpha) = -f(g), \quad g \in G.$

Procedure of Theorem 2 can then still be used to compute Y but for the redefinition of the sequence v as

$$v(g) = \delta_\alpha(g)\delta_2(g')y(\psi_2 g'),\ g \in G$$

where $g' = \ominus g$, $\delta_\alpha(g) = 1$, if $\ominus g \in G_\alpha$ and $g' = \alpha \ominus g$, $\delta_\alpha(g) = -1$ otherwise.

Note that now, $v(g \oplus \alpha) = -v(g)$, $g \in G$, whereas in Theorem 2, $v(g \oplus \alpha) = v(g)$, $g \in G$.

Theorem 3: Let $\alpha \in G$ be an order 2 element such that $u(g) = u(g \oplus \alpha)$, $v(g) = v(g \oplus \alpha)$, $\forall g \in G$. Then their convolution w can be computed as a multidimensional cyclic convolution of patterns with only $|G|/2$ points each.

Proof: One can express G as $G \simeq C_{2^n} \times C_{n_2} \times C_{n_3} \times \cdots \times C_{n_r}$ where $\alpha \in C_{2^n}$. Then in (3), one has $U(i_1, i_2, \cdots i_r) = U(i_1 + 2^{n-1}, i_2, \cdots, i_r)$ $V(i_1, i_2, \cdots, i_r) = V(i_1 + 2^{n-1}, i_2, \cdots, i_r)$

$$0 \le i_1 \le 2^{n-1} - 1,\ 0 \le i_t \le n_t - 1, \qquad 2 \le t \le r.$$

Consequently,

$$W(i_1, i_2, \cdots, i_r) = W(i_1 + 2^{n-1}, i_2, \cdots, i_r) \quad \text{and } W' = 2 \cdot U' * V'$$

where U', V', and W' are identical to U, V, and W, respectively, except that their first index i_1 varies only from 0 to $2^{n-1} - 1$. Thus W can be computed as an r-dimensional cyclic convolution of two patterns with $2^{n-1} \times n_2 \times n_3 \times \cdots \times n_r$ points. //

Example 3: Convolution of $u = (0\,5\,7\,2\,3\,1\,0\,5\,7\,2\,3\,1)$ and $v = (2\,2\,3\,4\,6\,0\,2\,2\,3\,4\,6\,0)$ with respect to $C_4 \times C_3$. Then from (2) and (3)

$$U = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 5 & 3 & 5 & 3 \\ 7 & 1 & 7 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 2 & 4 & 2 & 4 \\ 2 & 6 & 2 & 6 \\ 3 & 0 & 3 & 0 \end{bmatrix}.$$

Therefore,

$$U' = \begin{bmatrix} 0 & 2 \\ 5 & 3 \\ 7 & 1 \end{bmatrix}, \quad V' = \begin{bmatrix} 2 & 4 \\ 2 & 6 \\ 3 & 0 \end{bmatrix}$$

and

$$W' = 2U' * V' = 2 \cdot \begin{bmatrix} 43 & 57 \\ 55 & 33 \\ 46 & 72 \end{bmatrix} = \begin{bmatrix} 86 & 114 \\ 110 & 66 \\ 92 & 144 \end{bmatrix}.$$

Finally,

$$W = \begin{bmatrix} 86 & 114 & 86 & 114 \\ 110 & 66 & 100 & 66 \\ 92 & 144 & 92 & 144 \end{bmatrix}$$

and

$$w = \{86\quad 110\quad 92\quad 114\quad 66\quad 144\quad 86\quad 110\quad 92\quad 114\quad 66\quad 144\}.$$

Note that the computation of the convolution was effected through $U' * V'$ where each U' and V' has half the number of points of those of U and V. //

Theorem 4: Let $\alpha \in G$ be an order 2 element such that $u(g) = -u(g \oplus \alpha)$, $v(g) = -v(g \oplus \alpha)$, $\forall g \in G$. Then the number of operations required to evaluate their convolution with respect to G can be considerably reduced.

Proof: Proceeding as in the earlier proof, one now has

$$U(i_1, i_2, \cdots, i_r) = -U(i_1 + 2^{n-1}, i_2, \cdots, i_r)$$
$$V(i_1, i_2, \cdots, i_r) = -V(i_1 + 2^{n-1}, i_2, \cdots, i_r)$$
$$0 \le i_1 \le 2^{n-1} - 1,\ 0 \le i_t \le n_t - 1, \qquad 2 \le t \le r. \quad (4)$$

Thus, $W(i_1, i_2, \cdots, i_r) = -W(i_1 + 2^{n-1}, i_2, \cdots, i_r)$ and one needs to compute W only for $0 \le i_1 \le 2^{n-1} - 1$; $0 \le i_t \le n_t - 1$, $2 \le t \le r$. The use of fast algorithms [19] to compute the r-dimensional cyclic convolution of U and V calls for a linear transformation of these patterns. It can be verified that for U, V satisfying (4), many of the components of the transformed sequences are zero and operations due to them are saved. //

Based on the algorithms of [19], one can easily construct modified algorithms as given in Appendix A to convolve sequences satisfying the conditions of Theorem 4 for $G = C_{2^n}$, $n = 1, 2, 3$. When $G = C_{2^n} \times C_{n_2} \times C_{n_3} \times \ldots \times C_{n_r}$, the modified algorithm of convolution with respect to G can be obtained by combining the modified algorithm for C_{2^n} with the cyclic algorithms of lengths $n_2, n_3, \ldots, n_r$. Table I compares the number of operations required to implement the convolution of sequences satisfying Theorems 3 and 4 with those of [19].

Example 4: Convolution of $u = (0\,5\,7\,2\,3\,1\,0\,-5\,-7\,-2\,-3\,-1)$ and $V = (2\,2\,3\,4\,6\,0\,-2\,-2\,-3\,-4\,-6\,0)$ with respect to $C_4 \times C_3$. As in Example 3, one now has

$$U = \begin{bmatrix} 0 & 2 & 0 & -2 \\ 5 & 3 & -5 & -3 \\ 7 & 1 & -7 & -1 \end{bmatrix} \text{ and } V = \begin{bmatrix} 2 & 4 & -2 & -4 \\ 2 & 6 & -2 & -6 \\ 3 & 0 & -3 & 0 \end{bmatrix}.$$

To compute $U * V$ as per [19], write U and V as $U = \{u_0, u_1, -u_0, -u_1\}$, $V = \{v_0, v_1, -v_0, -v_1\}$ where u_0, u_1, v_0, v_1 stand for the columns. Applying the modified Algorithm 2 of Appendix A, one has

$$c_0 = \begin{bmatrix} 0 \\ 10 \\ 14 \end{bmatrix}, \quad c_1 = \begin{bmatrix} -4 \\ 4 \\ 12 \end{bmatrix}, \quad c_2 = \begin{bmatrix} 4 \\ 16 \\ 16 \end{bmatrix}$$

$$d_0 = \begin{bmatrix} 6 \\ 8 \\ 3 \end{bmatrix}, \quad d_1 = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}, \quad \text{and} \quad d_2 = \begin{bmatrix} 4 \\ 6 \\ 0 \end{bmatrix}.$$

The product $m_i = c_i d_i$ is now to be interpreted as a length-3 cyclic convolution of c_i and d_i as per [19]. One then has

$$m_0 = \begin{bmatrix} 142 \\ 102 \\ 164 \end{bmatrix}, \quad m_1 = \begin{bmatrix} 28 \\ 36 \\ 20 \end{bmatrix}, \quad \text{and} \quad m_2 = \begin{bmatrix} 112 \\ 88 \\ 160 \end{bmatrix}$$

Finally, going back to Algorithm 2 of Appendix A,

$$w_0 = -w_2 = \begin{bmatrix} 30 \\ 14 \\ 4 \end{bmatrix}, \quad w_1 = -w_3 = \begin{bmatrix} 114 \\ 66 \\ 144 \end{bmatrix}.$$

Thus

$$W = \begin{bmatrix} 30 & 114 & -30 & -114 \\ 14 & 66 & -14 & -66 \\ 4 & 144 & -4 & -144 \end{bmatrix}$$

or $w = \{30\ 4\ 14\ 114\ 144\ 66\ -30\ -4\ -14\ -114\ -144\ -66\}$. //

IV. Development of the Algorithms

The computation of all those DCT components for which $\gcd(j, 2N) = t$ is done together. First consider the case $t | N$, i.e., t dividing N. Utilizing the properties of the cosine function, (1) can be written as

$$Y(j) = \sum_{\substack{s|(N/t) \\ s \text{ odd}}} X_s(j),$$

where

$$X_s(j) = \Sigma_{i \in A} M(j, i) x_t(i), \qquad A = \{0 \le i < N/2t \,|\, \gcd(2i+1, N/t) = s\}, \tag{5}$$

$$x_t(i) = z_t(i) - z_t(N/t - 1 - i), \qquad i = 0, 1, \cdots, \lfloor N/2t \rfloor - 1, \tag{6}$$

$$z_t(i) = \sum_{d=0}^{t-1} (-1)^d y(i + dN/t), \qquad i = 0, 1, \cdots, N/t - 1. \tag{7}$$

$\lfloor \cdot \rfloor$ denotes the integer part. Unless the context demands x_t will be referred to merely as x. Further, because $X_s(j_1) = X_s(j_2)$ if $(j_1 + j_2)/(2N/s)$ or $(j_1 - j_2)/(2N/s)$ is an even integer and $X_s(j_1) = X_s(j_2)$ if it is an odd integer, one may partition the set of j's under consideration into subsets such that j_1, j_2 will belong to the same subset iff $(2N/s) | (j_1 + j_2)$ or $(j_1 - j_2)$. Equation (5) then needs to be evaluated only for $j \in B$ where B denotes the representatives of these subsets. It will be shown later that A and B have the same number of elements.

Example 5: Computation of the DCT of length $N = 10$ of $y = (0\ 2\ 1\ 1\ 3\ -1\ 0\ 0\ 2\ -1)$. Possible values of t dividing N are 1, 2 and 5.

When $t = 1$, $Y(1)$, $Y(3)$, $Y(7)$, and $Y(9)$ are being computed. One has, then, $Y(j) = X_1(j) + X_5(j)$ where

$$X_1(j) = \sum_{i \in \{0,1,3,4\}} M(j, i) x_1(i), \qquad j \in \{1, 3, 7, 9\}, \tag{8}$$

$$X_5(j) = \sum_{i \in \{2\}} M(j, i) x_1(i), \qquad j \in \{1\}, \tag{9}$$

and $x_1 = (1\ 0\ 1\ 1\ 4)$. Note that for $(t, s) = (1, 5)$, since A is a one element set so should be B. The set of j's is $\{1, 3, 7, 9\}$. One can use the value of $2N/s = 4$ to show that all the elements of this set are in the same subset as defined earlier. Thus one may take $B = \{1\}$ in (9) and, $X_5(3) = -X_5(1)$ as $(3 + 1)/(2N/s)$ is odd and similarly $X_5(7), = X_5(9) = X_5(1)$ as $(7 + 1)/(2N/s)$ and $(9 - 1)/(2N/s)$ are even. When $t = 2$, $Y(2)$, and $Y(6)$ are computed as $Y(j) = X_1(j)$ where

$$X_1(j) = \sum_{i \in \{0,1\}} M(j, i) x_2(i), j \in \{2, 6\} \tag{10}$$

and $x_2 = (-3, 3)$.

When $t = 5$, $Y(5)$ is being computed as $Y(j) = X_1(j)$ where

$$X_1(j) = \sum_{i \in \{0\}} M(j, i) x_5(i), \qquad j \in \{5\} \tag{11}$$

and $x_5 = (5)$. //

It will now be shown that a certain choice of $G, \psi_1, \psi_2, \delta_1, \delta_2$, and f converts (5) for $j \in B$ into a convolution with respect to G. Let $L = N/st$ and $H = A(4L)$. Then $\alpha = 2L - 1$ and $\beta = 2L + 1 \in H$ and are of order 2. The choice of G is dictated by the nature of $\langle \alpha \rangle$ and $\langle \beta \rangle$.

First, at least one of $\langle \alpha \rangle$ or $\langle \beta \rangle$ should be a splitting subgroup of H because if neither of them is splitting subgroups, then both α and β should have square-roots modulo $4L$. But $\alpha\beta = -1 \pmod{4L}$. Thus -1 should have a square-root modulo $4L$ which is impossible. So we have the following two cases.

Case 1: If both $\langle \alpha \rangle$ and $\langle \beta \rangle$ are splitting subgroups of H, i.e., $H \simeq \langle \alpha \rangle \times \langle \beta \rangle \times G$, one can establish a one-one correspondence between the elements of G and B as under. For every $g \in G$, there exist integers n_1 and d_1 such that

$$g = (-1)^{n_1} \frac{j}{t} + d_1 \cdot 2L \tag{12}$$

TABLE I
A Comparison of the Computational Complexity of the Usual Cyclic Convolution Algorithms [19] and the Modified Algorithms Valid for Sequences Satisfying Conditions of Theorems 3 and 4

Sequence length	Usual Algorithms[19]		Modified Algorithms For Theorem 3		Modified Algorithms For Theorem 4	
	Mults.	Adds	Mults.	Adds.	Mults.	Adds.
2	2	4	1	0	1	0
4	5	15	2	4	3	3
8	14	46	5	15	9	15

for some $j \in B$. If (12) is satisfied, j is defined as $\psi_1 g$. The one-one and onto nature of ψ_1 is proved in Appendix B. Equation (12) also defines δ_1 as $\delta_1(g) = (-1)^{d_1}$. Similarly, one may find integers n_2 and d_2 satisfying

$$g = (-1)^{n_2}(2i + 1)/s + d_2 \cdot 2L \tag{13}$$

where $g \in G, i \in A$. This establishes a one-one correspondence between the elements of G and A. The proof for this is omitted as it runs parallel to that of (12). We then define

$$\psi_2 g = i \text{ and } \delta_2(g) = (-1)^{d_2}.$$

Furthermore, by choosing f as

$$f(g) = \cos (g\pi/2L), \tag{14}$$

it can be verified that the requirements of Theorem 1 are satisfied and (5) may be computed by defining sequences u and v as $u(g) = \cos (g\pi/2L)$ and $v(g) = \delta_2(\ominus g)x_t(\psi_2(\ominus g))$, $g \in G$ and convolving them with respect to G to get the sequence w related to X_s as $X_s(\psi_1 h) = \delta_1(h)w(h)$, $h \in G$.

Example 5 (Cont'd.): For $(t, s) = 1$ [computation of (8)], $L = 10$. Structure of $H = A(40)$, from Example 1 is $A(40) = \{1, 19\} \times \{1, 21\} \times \{1, 3, 9, 27\} = \langle\alpha\rangle \times \langle\beta\rangle \times G$. Equations (12) and (13) are satisfied for the following values of n_1, d_1, and n_2, d_2. [Equation (13) is computed for $\ominus g$ rather than for g because one needs $\delta_2(\ominus g)$ and $\psi_2(\ominus g)$.]

$g \in G$	n_1	d_1	$\psi_1 g \in B$	$\ominus g$	n_2	d_2	$\psi_2(\ominus g) \in A$
1	0	0	1	1	0	0	0
3	0	0	3	27	0	1	3
9	0	0	9	9	0	0	4
27	0	1	7	3	0	0	1

Finally a convolution with respect to G (which is a cyclic convolution of length 4 as $G = C_4$) of U and V gives W, where

$$U = (\cos(\pi/20), \cos(3\pi/20), \cos(9\pi/20), \cos(27\pi/20)),$$

$$V = (x_1(0), -x_1(3), x_1(4), x_1(1)) = (1, -1, 4, 0) \text{ and}$$

$$W = U * V = (X_1(1), X_1(3), X_1(9), -X_1(7)) = (2.068, -1.913, 3.216, 2.954).$$

For $(t, s) = (1, 5)$ [computation of (9)], $L = 2$, $H = A(8) = C_2 \times C_2 = \{1, 3\} \times \{1, 5\} = \langle\alpha\rangle \times \langle\beta\rangle$. Thus $G = \{1\}$ and $X_5(1)$ is a convolution with respect to G (i.e., a simple multiplication) of $U = (\cos (\pi/4))$ and $(x_1(2)) = (1)$. Thus $X_5(1) = 0.707$. Similarly, for $(t, s) = (5, 1)$ [computation of (11)], $G = \{1\}$ and $X_1(t) = (\cos (\pi/4)) \cdot x_5(0) = 3.535$. //

Case 2: If only one of $\langle\alpha\rangle$ or $\langle\beta\rangle$ is a splitting subgroup of H, then $H \simeq \langle\alpha\rangle \times G$ or $H \simeq \langle\beta\rangle \times G$. We assume here that $H \simeq \langle\beta\rangle \times G$ but the case of $H \simeq \langle\alpha\rangle \times G$ may be worked out similarly.

Note that now $\alpha \in G$. It may be verified that defining ψ_1, ψ_2, δ_1, δ_2, and f as in Case 1, the requirements of Theorem 2 are satisfied except that

$$\begin{aligned} f(g \oplus \alpha) &= \cos((\alpha \oplus g)\pi/2L) \\ &= \cos(\alpha g\pi/2L) \quad \text{as } \oplus \text{ denotes mult.mod } 4L \\ &= \cos((2L - 1)g\pi/2L) \\ &= -\cos(g\pi/2L) = -f(g), \qquad g \in G. \end{aligned}$$

Therefore, following the remark after Theorem 2, one gets the algorithm, in this case, as $X_s(\psi_1 h) = \delta_1(h)w(h)$, $h \in G_\alpha$ where w is the convolution with respect to G of sequences u and v defined as

$$\begin{aligned} u(g) &= -u(g \oplus \alpha) = \cos(g\pi/2L)/2, \\ v(g) &= -v(g \oplus \alpha) = \delta_2(g')\delta_\alpha(g)x_t(\psi_2 g'), \; g \in G_\alpha \end{aligned}$$

where

$$\begin{aligned} g' &= \ominus g \qquad &&\text{if } \ominus g \in G_\alpha \\ &= \alpha \ominus g &&\text{otherwise.} \end{aligned}$$

Note that $u * v$ can be evaluated efficiently as in Theorem 4.

Example 5 (Cont'd.): For $(t, s) = (2, 1)$ (computation of (10)) $L = 5$, $\alpha = 9$, and $\beta = 11$. $A(4L) = \{1, 11\} \times \{1, 3, 9, 7\} = \langle\beta\rangle \times G$. Thus $G_\alpha = \{1, 3\}$. Equations (12) and (13) are satisfied by following values of n_1, d_1, n_2, and d_2.) (Equation (13) is computed for g' rather than for g because one needs $\delta_2(g')$ and $\psi_2 g'$.)

$g \in G_\alpha$	n_1	d_1	$\psi_1 g \in B$	$\ominus g$	g'	n_2	d_2	$\psi_2 g' \in A$	$\delta_\alpha(g)$
1	0	0	2	1	1	0	0	0	1
3	0	0	6	7	3	0	0	1	−1

Finally, a convolution with respect to G (which is a cyclic convolution of length 4 as $G = C_4$) of U and V gives W where

$$U = (\cos (\pi/10)/2, \cos (3\pi/10)/2, -\cos (\pi/10)/2, -\cos (3\pi/10)/2)$$

$$V = (x_2(0), -x_2(1), -x_2(0), x_2(1)) = (-3, -3, 3, 3).$$

Thus $U * V = W = (-1.089, -4.617, 1.089, 4.617)$ or $(-1.089, -4.617) = (X_1(2), X_1(6))$.

Note that since U and V satisfy conditions of Theorem 4, one can use modified Algorithm 2 of Appendix A to implement this convolution. //

For $t \nmid N$, (1) can be reduced to

$$Y(j) = (-1)^{j/t} z_t(\lfloor N/t \rfloor) + \sum_{\substack{s | (2N/t) \\ s \text{ odd}}} X_s(j) \tag{15}$$

where

$$X_s(j) = \sum_{i \in A} M(j, i)\, x_t(i). \tag{16}$$

$$x_t(i) = z_t(i) + z_t(2N/t - 1 - i), \qquad i = 0, 1, \cdots, \lfloor N/t \rfloor - 1 \tag{17}$$

$$z_t(i) = \sum_{d=0}^{t/2-1} y(i + d\, 2N/t), \qquad i = 0, 1, \cdots, 2N/t - 1$$

$$A = \{0 \le i < N/t \,|\, \gcd(2i + 1, 2N/t) = s\}. \tag{18}$$

Furthermore, because of the arguments similar to those in the case of $t|N$, (16) need be computed only for $j \in B$, where B is a set defined as earlier.

Example 5 (Cont'd.): The only value of $t \not| N$ but $t|2N$ is 4 and for this $Y(4)$ and $Y(8)$ are being computed. One then has

$$Y(4) = -z_4(2) + X_1(4)$$
$$Y(8) = z_4(2) + X_1(8)$$

where

$$X_1(j) = \sum_{i \in \{0,1\}} M(j, i)\, x_4(i), \qquad j \in \{4, 8\} \tag{19}$$

$$z_4 = (-1, 2, 1, 3, 2), \qquad x_4 = (1, 5). \qquad //$$

Let $L = 2N/st$ and $H = A(2L)$. Then $\alpha = 2L - 1 \in H$ is of order 2. The choice of G is dictated by the nature of $\langle \alpha \rangle$ and the following two cases arise:

Case 3: If $\langle \alpha \rangle$ is a splitting subgroup of H, $H \simeq \langle \alpha \rangle \times G$. One can then establish a one-one onto relation between the group elements g and the elements j of B as

$$g = (-1)^{n_1} \frac{j}{t} + d_1 L. \tag{20}$$

Then $\psi_1 g$ and $\delta_1(g)$ are defined as j and $(-1)^{d_1}$, respectively. ψ_2 can be obtained as in (13) and δ_2 may be taken as $\delta_2(g) = 1$, $\forall\, g \in G$. Finally, choosing f as

$$f(g) = \cos(g\pi/L) \tag{21}$$

requirements of Theorem 1 may be shown to be satisfied. Consequently, the required algorithm for (16) will read as

$$X_s(\psi_1 h) = \delta_1(h)\, w(h), \qquad h \in G$$

where w is the convolution with respect to G of the sequences u and v defined as

$$u(g) = \cos(g\pi/L),$$
$$v(g) = x_t(\psi_2(\ominus g)), \qquad \forall\, g \in G.$$

Case 4: When $\langle \alpha \rangle$ is not a splitting subgroup of H, let $G = H$. Now $\alpha \in G$. By defining ψ_1, ψ_2, δ_1, δ_2, and f as in Case 3, it can be easily verified that the requirements of Theorem 2 are fulfilled. In particular, now $f(g \oplus \alpha) = f(g)$, $g \in G$. (Compare with Case 2.) The algorithm in this case thus reads as

$$u(g) = u(g \oplus \alpha) = \cos(g\pi/L)/2$$
$$v(g) = v(g \oplus \alpha) = x_t(\psi_2 g'), \qquad g \in G_\alpha$$

where

$$g' = \ominus g \qquad \text{if } \ominus g \in G_\alpha$$
$$= \alpha \ominus g \qquad \text{otherwise.}$$

Theorem 3 may be utilized to evaluate this convolution efficiently.

Example 5 (Cont'd.): For $(t, s) = (4, 1)$ [computation of (19)], one has $L = 5$, $H = A(2L) = C_4$. Thus $G = C_4 = \langle 3 \rangle = \{1, 3, 9, 7\}$. Note that $\alpha = 9 \in G$. Corresponding to (20) and (13) we have [(13) is evaluated for g' rather than for g as one needs $\psi_2 g'$]

$g \in G_\alpha$	n_1	d_1	$\psi_1 g \in B$	$\ominus g$	g'	n_2	d_2	$\psi_2 g' \in A$
1	0	0	4	1	1	0	0	0
3	1	1	8	7	3	0	0	1

Thus a convolution with respect to C_4 of W and V gives $X_1(j)$, where

$$U = (\cos(\pi/5)/2, \cos(3\pi/5)/2, \cos(\pi/5)/2, \cos(3\pi/5)/2)$$
$$V = (x_4(0), x_4(1), x_4(0), x_4(1))$$
$$= (1 \quad 5 \quad 1 \quad 5)$$
$$W = U * V = (X_1(4), -X_1(8), X_1(4), -X_1(8))$$

from Theorem 3, $U * V$ can be evaluated as a length-2 cyclic convolution of $(\cos(\pi/5), \cos(3\pi/5))$ and $(1\ 5)$ to give $(X_1(4), -X_1(8)) = (-0.736, 3.736)$. //

V. Computational Complexity

The algorithm developed has three computational stages.

i) For every t, $(t < N, t|2N)$, compute x_t sequences through (6), (7), (17), and (18).

ii) Use the x_t's so computed in various convolutional algorithms.

iii) Recombine the results of the convolutions as per (5) and (15) to get the transform.

We now analyze the computational complexity of each of these stages.

i) No multiplications are involved at this stage. The number of additions involved is substantially reduced by adopting the following scheme (proofs omitted as they can be easily constructed).

To Compute x_t's for $t \not| N$

Let p be the largest integer satisfying $2^p < N$, $2^p | N$. Then one needs to compute x_{2^n}, $n = 0, 1, \ldots, p$. Define $r_0(i) = y(i)$, $i = 0, 1, \ldots, N-1$,

$$r_n(i) = r_{n-1}(i) + r_{n-1}(N/2^{n-1} - 1 - i), \qquad i = 0, 1, \ldots, N/2^n - 1, \quad 1 \le n \le p.$$

Then

$$x_{2^n}(i) = r_n(i) - r_n(N/2^n - 1 - i), \qquad i = 0, 1, \ldots, \lfloor N/2^{n+1} \rfloor - 1, \quad 0 \le n \le p.$$

These involve $2N(1 - 1/2^p) + \lfloor N/2^{p+1} \rfloor$ additions. To compute x_{t_1} from x_{t_2} where $t_1, t_2 | N$ and $t_1 = kt_2$ for an odd integer k, one may use

$$x_{t_1}(i) = \sum_{q=0}^{\lfloor k/2 \rfloor} (-1)^q x_{t_2}(i + Nq/t_1) - \sum_{q=\lfloor k/2 \rfloor + 1}^{k-1} (-1)^q x_{t_2}(N/t_2 - 1 - i - Nq/t_1)$$

requiring only $(k-1)\lfloor N/2t_1 \rfloor$ additions.

To Compute x_t's for $t \not| N$

Such t's exist iff N is not a power of 2. Then

TABLE II
A COMPARISON OF THE COMPUTATIONAL COMPLEXITY OF THE NEW DCT ALGORITHM WITH THE BEST OF THE CONVENTIONAL DCT ALGORITHM [11]-[13] AND THE WFTA [15]

Sequence length	New DCT Multiplications.	New DCT Additions	Conventional DCT Multiplications	Conventional DCT Additions	WFTA Multiplications	WFTA Additions
3	3	4	7	24	2	9
4	5	9	6	8	0	12
5	6	15	15	54	5	22
7	9	34	23	100	8	43
8	14	32	16	26	2	34
9	13	46	29	126	10	54
13	21	110	53	300	20	137
21	37	195	73	384	26	171
29	96	381	219	1018	95	480
30	53	269	90	250	34	222
31	81	418	191	1080	80	509
33	85	423	157	876	62	405
60	130	728	184	562	68	504
65	151	1048	315	2190	125	1030
99	325	1983	559	3630	230	1716
129	613	2573	1045	5748	458	2745
258	1229	6943	1428	6004	916	5748
511	3097	17955	6071	35236	2780	17107
631	8361	40626	17351	86296	8360	42517
1262	16723	83774	19240	87556	16720	86296

$$x_{2^{p+1}}(i) = r_p(i) + r_p(N/2^p - 1 - i), \quad i = 0, 1, \ldots, \lfloor N/2^{p+1} \rfloor - 1.$$

This calls for $\lfloor N/2^{p+1} \rfloor$ additions.

Similarly, to get x_{t_1} from x_{t_2} where $t_1, t_2 \nmid N$ and $t_1 = kt_2$, one may use

$$x_{t_1}(i) = \sum_{q=0}^{\lfloor k/2 \rfloor} x_{t_2}(i + 2qN/t_1) + \sum_{q=\lfloor k/2 \rfloor + 1}^{k-1} x_{t_2}(2N/t_2 - 1 - i - 2qN/t_1)$$
$$i = 0, 1, \ldots, \lfloor N/t_1 \rfloor - 1,$$

requiring only $(k-1)\lfloor N/t_1 \rfloor$ additions.

Note from (15) that when $t \nmid N$, $z_t(\lfloor N/t \rfloor)$ also is required which could be computed as

$$z_{2^{p+1}}(\lfloor N/2^{p+1} \rfloor) = r_p(\lfloor N/2^{p+1} \rfloor)$$

requiring no extra operations and for $t_1 = kt_2$,

$$z_{t_1}(\lfloor N/t_1 \rfloor) = \sum_{q=0}^{\lfloor k/2 \rfloor - 1} x_{t_2}(\lfloor N/t_1 \rfloor + 2qN/t_1) + z_{t_2}(\lfloor N/t_2 \rfloor)$$

requiring only $\lfloor k/2 \rfloor$ additions.

Finally, to compute $Y(0)$, one needs to sum up all the y components. Let m be the smallest odd prime dividing N. Then

$$y(0) + y(1) + \ldots + y(N-1) = \sum_{i=0}^{\lfloor m/2 \rfloor - 1} x_{2N/m}(i) + y(\lfloor N/2 \rfloor),$$

requiring only $\lfloor m/2 \rfloor$ additions. When N is a power of 2, $y(0) + y(1) + \ldots + y(N-1) = r_p(0) + r_p(1)$ requiring only one addition.

ii) To determine the computational complexity of stage ii), note that each (t, s) pair leads to a multidimensional cyclic convolution decided by the group which in turn is determined by N/ts. The complexity of this stage is therefore obtained by summing of the complexities of these convolutions over all possible (t, s) pairs.

iii) This stage also does not involve any multiplications. Let S_t denote the total number of s values possible for any t and J_t, the number of j's satisfying

$$\{1 \leq j \leq N-1 \mid \gcd(j, 2N) = t\}. \tag{22}$$

Then the total additions required in this stage is

$$\sum_{\substack{t \mid N \\ t < N}} (S_t - 1)J_t + \sum_{\substack{t \mid 2N \\ t \nmid N, t < N}} S_t J_t$$

where the two summations correspond to (5) and (15).

Example 6: For $N = 10$, $p = 1$. x_1 and x_2 can be obtained in $(2 \cdot 10)(1 - 1/2) + \lfloor 10/4 \rfloor = 12$ additions. x_5 can be obtained from x_1 in $(5-1)\lfloor 10/10 \rfloor = 4$ additions. x_4 can be obtained in $\lfloor 10/4 \rfloor = 2$ additions. $y(0) + \ldots + y(9)$ is obtained from x_4 in $\lfloor 5/2 \rfloor = 2$ additions. Thus stage i) requires 20 additions. Possible (t, s) pairs are $(1, 1)$, $(1, 5)$, $(2, 1)$, $(5, 1)$, and $(4, 1)$. For $(t, s) = (1, 1)$, $H = A(4N/ts) = C_2 \times C_2 \times C_4$. Thus $G = C_4$. Convolution with respect to C_4 requires 5 multiplications and 15 additions. Similarly for $(t, s) = (1, 5)$ or $(5, 1)$, one needs a convolution with respect to $G = \{1\}$ requiring only a single multiplication. For $(t, s) = (2, 1)$, $G = C_4$ and $\alpha \in G$. Therefore Theorem 4 is applicable for this convolution. From Table I, this requires 3 multiplications and 3 additions. For $(t, s) = (4, 1)$, $G = C_4$. Theorem 3 is now applicable and from Table I the convolution requires 2 multiplications and 4 additions. Stage ii) thus needs 12 multiplications and 22 additions. From the given (t, s) pairs, $S_1 = 2$, $S_2 = 1$, $S_5 = 1$, $S_4 = 1$. Also the number of j's satisfying (22) gives $J_1 = 4$, $J_2 = 2$, $J_5 = 1$ and $J_4 = 2$. Thus stage iii) requires 6 additions. Therefore the DCT of length 10 can be evaluated using the algorithm of this paper in 12 multiplications and 48 additions. As against this, the algorithm of [12] requires 26 multiplications and 62 additions and that of [13] requires 30 multiplications and 118 additions. //

The complexity of the new DCT algorithm is compared in Table II with that of the best of [11]-[13]. It also lists the complexity of the WFTA [15].

CONCLUSIONS

This paper suggests a new approach to the fast computation of the DCT of any length. The methodology is based upon partitioning the DCT kernel into submatrices using the properties of the cosine function. The submatrices can be made equivalent to the group tables of appropriate Abelian groups by row and column shufflings and negations determined systematically through a group-theoretic procedure. The computations pertaining to each of the submatrices are performed as multidimensional cyclic convolutions. The submatrices which cannot be made equivalent to group tables can be viewed upon as parts of larger group tables. Computations related to these can be carried out using the modified algorithms.

Construction of an algorithm for a particular N involves determining the shufflings and negations of sequence components based upon group-theoretic reasoning. However, for most of the applications, the DCT computation has to be performed repeatedly for many data sets for the same N. It can therefore be assumed that the required algorithm is set up prior to the beginning of the actual computations. The time involved and the complexity encountered in setting up the algorithm is thus of no consequence in the DCT computation.

Table II, which lists the complexities of the algorithms, shows the superiority of the new DCT algorithm compared with those of [11]–[13]. Particularly for prime lengths, the new DCT algorithm is comparable (in terms of number of computations) to WFTA. It can also be seen that since the new algorithm makes use of cyclic convolutional algorithms and since they are efficient only for short lengths, the efficiency of the new DCT algorithm is dependent upon the possibility of factoring $\phi(4N)$, the Euler ϕ function, into small, relatively prime factors.

APPENDIX A

MODIFIED CYCLIC CONVOLUTIONAL ALGORITHMS FOR SEQUENCES SATISFYING THE REQUIREMENTS OF THEOREM 4

Algorithm 1

Two-point algorithm, multiplications 1, additions 0:

$$c_0 = 2U(0) \qquad d_0 = V(0)$$
$$c_0 = c_0 d_0$$
$$W(0) = -W(1) = m_0.$$

Algorithm 2

Four-point algorithm, multiplications 3, additions 3:

$$c_0 = 2U(0) \qquad d_0 = V(0) + V(1)$$
$$c_1 = 2(U(0) - U(1)) \qquad d_1 = V(0)$$
$$c_2 = 2(U(0) + U(1)) \qquad d_2 = V(1)$$
$$m_k = c_k d_k,\ k = 0, 1, 2.$$
$$W(0) = -W(2) = m_0 - m_2$$
$$W(1) = -W(3) = m_0 - m_1.$$

Algorithm 3

Eight-point algorithm, multiplications 9, additions 15:

$$c_0 = 2(U(0) - U(3))$$
$$c_1 = 2U(0)$$
$$c_2 = 2(U(0) + U(1))$$
$$c_3 = 2((U(2) + U(0)) + (U(1) - U(3)))$$
$$c_4 = 2(U(2) + U(0))$$
$$c_5 = 2((U(2) + U(0)) + (U(1) + U(3)))$$
$$c_6 = 2((U(2) - U(0)) + (U(1) + U(3)))$$
$$c_7 = 2(U(2) - U(0))$$
$$c_8 = 2((U(2) - U(0)) - (U(1) - U(3)))$$
$$d_0 = V(1) + V(3)$$
$$d_1 = (V(0) + V(2)) - (V(1) + V(3))$$
$$d_2 = V(0) + V(2)$$
$$d_3 = V(3)$$
$$d_4 = V(2) - V(3)$$
$$d_5 = V(2)$$
$$d_6 = V(1)$$
$$d_7 = V(0) - V(1)$$
$$d_8 = V(0)$$
$$m_k = c_k d_k,\ k = 0, 1, \cdots, 8.$$
$$W(0) = -W(4) = \{(m_0 + m_1) - (m_3 + m_4)\}$$
$$W(1) = -W(5) = \{(m_2 - m_1) + (m_4 - m_5)\}$$
$$W(2) = -W(6) = \{(m_0 + m_1) + (m_6 + m_7)\}$$
$$W(3) = -W(7) = \{(m_2 - m_1) + (m_8 - m_7)\}.$$

APPENDIX B

PROPERTIES OF THE MAPPING (12)

Existence

Since $\gcd(g, 2L) = 1$, by using Dirichlet's theorem on primes in arithmetical progression, one may find [20] integer k such that $q = g^{+}k2L$ is a prime. Then $\gcd(q, 2Ls) = 1$. Furthermore, by properly choosing k', $-Ls < q + k'2Ls < Ls$ and $\gcd(q + k'2Ls, 2Ls) = 1$. Thus $j = |q + k'2Ls|t$ belongs to the set of the indices under consideration. In particular, it will belong to a subset of this set (defined by the condition that j_1, j_2 belong to the same subset iff $(2N/s)|(j_1 + j_2)$ or $(j_1 - j_2)$). Let $\psi_1 g \in B$ be the representative of this subset. Then $\psi_1 g = \pm j + k''2N/s$ for some integer k''. By substituting for j one gets the relation (12).

One-One

Let $g_1 = (-1)^{n_1}(\psi_1 g_1/t) + d_1 2L$ and $g_2 = (-1)^{n_2}(\psi_1 g_2/t) + d_2 2L$, for some n_1, d_1, and n_2, d_2. If $\psi_1 g_1 = \psi_1 g_2$ then, $(g_1 - d_1 2L)(-1)^{n_1} = (g_2 - d_2 2L)(-1)^{n_2}$ giving $2L|(g_1 + g_2)$

or $(g_1 - g_2)$. Thus, there are four possibilities:

$$g_1 + g_2 = 2L \text{ or } g_1 = -g_2 + 2L = g_2 \oplus \alpha,$$

$$g_1 + g_2 = 4L \text{ or } g_1 = -g_2 + 4L = g_2 \oplus \alpha \oplus \beta,$$

$$g_1 - g_2 = 2L \text{ or } g_1 = g_2 + 2L = g_2 \oplus \beta,$$

$$g_1 - g_2 = 0 \text{ or } g_1 = g_2$$

where $\oplus$ denotes the group operation of multiplication modulo $4L$. The first three cases are not possible because $\alpha, \beta \notin G$ but both $g_1, g_2 \in G$.

Onto

Any $b \in B$ gives $\gcd(b/t, 2L) = 1$ or $\gcd(b/t \bmod 2L, 2L) = 1$.

Let $b/t \bmod 2L = (b/t) + k\, 2L$

Thus $(b/t) \bmod 2L = (b/t) + k\, 2L = h \in H$. There are four cases each of which satisfy (12) with the following value of n and d.

$$h \in G \qquad n = 0 \quad d = 0$$

$$h \oplus \alpha = -(b/t) + (1 - k)2L \in G \qquad n = 1 \quad d = 1 - k$$

$$h \oplus \beta = (b/t) + (1 + k)2L \in G \qquad n = 0 \quad d = 1 + k$$

$$h \oplus \alpha \oplus \beta = -(b/t) + (2 - k)2L \in G \qquad n = 1 \quad d = 2 - k.$$

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.

[2] A. Habibi, "Hybrid coding of pictorial data," *IEEE Trans. Commun.*, vol. COM-22, pp. 614–624, May 1974.

[3] R. W. Means, H. T. Whitehouse, and J. M. Speiser, "Television encoding using a hybrid discrete cosine transform and a differential pulse code modulation in real time," in *Proc. IEEE Nat. Telecomm. Conf.*, San Diego, CA, Dec. 1974.

[4] W. Chen and S. C. Fralick, "Image enhancement using cosine transform filtering," in *Proc. Symp. Current Math. Prob. Image Sci.*, Monterey, CA, pp. 186–192, Nov. 1976.

[5] W. Chen and C. H. Smith, "Adaptive coding of monochrome and colour images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1285–1292, Nov. 1977.

[6] T. Natarajan and N. Ahmed, "On interframe transform coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 1323–1329, Nov. 1977.

[7] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe cosine transform image coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 1329–1339, Nov. 1977.

[8] N. Ahmed, P. J. Milne, and S. G. Harris, "Electrocardiographic data compression via orthogonal transforms," *IEEE Trans. Biomed. Eng.*, vol. BME-22, pp. 484–487, Nov. 1975.

[9] M. Hamidi and J. Pearl, "Comparison of the cosine and Fourier transforms of Markov-1 signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 428–429, Oct. 1976.

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

[11] W. 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.

[12] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934–946, June 1978.

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

[14] A. Peled and B. Liu, *Digital Signal Processing*. New York: Wiley, 1976.

[15] S. Winograd, "On computing the discrete Fourier transform," *Math. Comput.*, vol. 32, pp. 175–199, Jan. 1978.

[16] E. Schenkman, *Group Theory*. New York: Van Nostrand Reinhold, 1965.

[17] S. V. Kanetkar and M. D. Wagh, "Group character tables in discrete transform theory," *J. Comput. Syst. Sci.*, vol. 19, Dec. 1979.

[18] M. D. Wagh and H. Ganesh, "Generalised convolution and the computation of discrete transforms," in review.

[19] R. C. Agarwal and J. W. Cooley, "New algorithms for digital convolution," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 392–410, Oct. 1977.

[20] T. M. Apostol, *Introduction to Analytic Number Theory*. New York: Springer-Verlag, 1976.

Meghanad D. Wagh was born in Bombay, India, on September 23, 1948. He received the B.Tech. and Ph.D. degrees from the Indian Institute of Technology, Bombay, in 1971 and 1977, respectively.

From 1971 to 1976 he was a Research Assistant and during 1977–1978 a Research Associate with the Department of Electrical Engineering, Indian Institute of Technology, Bombay. Currently he is with the Department of Electrical Engineering, Concordia University, Montreal, P. Q., Canada. His research interests include the application of group theoretic techniques to digital signal processing and computational algorithms.

H. Ganesh received the B.Sc. degree in electrical engineering from the University of Kerala, India, and the M.S. degree in electrical engineering from the South Dakota State University, Brookings. From 1976 to 1979 he was with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, working for his Ph.D. degree.

Currently, he is with the Faculty of Electrical Engineering, Calicut Regional Engineering College, Calicut, India.