The Fast DCT Algorithm Based on the New Prime Factor and Common Factor Decomposition

Byeong-Ho Choi*, Jong-Uk Kim*, Ki-Bum Suh*, Jong-Wha Chong*, Gyo-Yoon Bang**

CAD & Communication circuit Lab
Dept of Electronic Eng. Hanyang Univ.*
Video Lab.
GoldStar Co., Ltd**

Abstract- In this paper, we present a new algorithm for the fast computation of the discrete cosine transform (DCT). This algorithm consists of the three dimensional prime factor-decomposed algorithm(PFA) and three dimensional common factor-decomposed algorithm (CFA). We can compute N-point DCT for the number N decomposable into three relative prime numbers using PFA and into three common numbers using CFA. We also show input and output index mapping for the three decomposition. It results in requiring fewer multiplicaions than the previous algorithms. Particularly, for the large number N, it is more powerful in reducing the number of multiplication.

I.INTRODUCTION

The discrete cosine transform(DCT) has been used for many years since its first introduction in 1974.[1] Because of its suboptimal property for data compression, the DCT has been of great interest in speech and image signal processing. Thus, various fast algorithms, reducing the number of multiplications, have been investigated by many researchers.[3]-[14]

Generally, we can divide fast algorithms into two categories depending on the number of points N : one on general composite number cases[5]-[6] and the other one on the prime factor cases[7][22]-[24]. In the case of prime factor, two dimensional prime factor DCT alorithm has been developed by LEE[7]. In this paper, we will extend prime factor algorithm(PFA) in three dimensional form, and show three dimensional common factor algorithm(CFA), and show their efficiency.

II. MATHEMATICAL DERIVATION OF THREE DEMENSIONAL PFA

In this paper, we represent the DCT algorithm having the block size N, where N is factorizable into three mutually relative prime number N_1, N_2 and N_3.
Therefore, we assume that

$$N=N_1 * N_2 * N_3 \tag{1}$$

where N_1, N_2 and N_3 are mutually prime integers.

Let $x(k)$, $k = 0,1,2, \ldots, N-1$, be the input sequence and $X(n)$, $n = 0,1,2, \ldots, N-1$, be the output sequence.
The DCT and the inverse DCT(IDCT) is defined as follows:

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

$$x(k) = \sum_{n=0}^{N-1} e(n)\, X(n) \cos[\,\pi(2k+1)n/2N\,], \quad k = 0,1,2,\ldots\ldots,N-1 \tag{3}$$

$$\text{where, } e(n) = \begin{cases} 1/\sqrt{2} & \text{if } n = 0 \\ 1 & \end{cases}$$

Since the DCT and the inverse DCT(IDCT) are the orthogonal transform, IDCT matrix can be obtained by transposing the DCT matrix.
Our purpose is to gain the $\cos[\pi(2k_1+1)n/2N_1]*\cos[\pi(2k_2+1)n/2N_2]*\cos[\pi(2k_3+1)n/2N_3]$ of three dimensinal DCT kernel from the $\cos[\pi(2k+1)n/2N]$ of one dimensional DCT kernel, that is,

$$x(k_1,k_2,k_3) = \sum_{n_3=0}^{N_3-1}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} X(n_1,n_2,n_3) * \cos[\,\pi(2k_1+1)n/2N_1\,] * \cos[\,\pi(2k_2+1)n/2N_2\,] * \cos[\,\pi(2k_3+1)n/2N_3\,] \tag{4}$$

$$\text{for } n_1,k_1 = 0,1,2,\cdots\cdots, N_1-1;\quad n_2,k_2 = 0,1,2,\cdots\cdots, N_2-1;\quad n_3,k_3 = 0,1,2,\cdots\cdots, N_3-1$$

Let $x(n), n=0,1,\cdots\cdots N-1$, be the one dimensional input sequence and $x(n_1, n_2, n_3)$, $n_1=0,1,\cdots\cdots N_1-1$, $n_2=0,1,\cdots\cdots N_2-1$, $n_3=0,1,\cdots\cdots N_3-1$, be the three dimensional input sequence.
We must define the input and output mapping to gain (4) from (3). Now we first consider the input mapping which connects $x(n)$ to $x(n_1, n_2, n_3)$.

Let Ω_1 denote the set of N integers from 0 to N-1. Similary, let Ω_1, Ω_2, and Ω_3, respectively, denote the sets of N_1 integers from 0 to N_1-1 and the sets of N_2 integers from 0 to N_2-1 and the sets of N_3 integers from 0 to N_3-1. We define four input mapping functions from $\Omega_1 x \Omega_2 x \Omega_3$ to Ω such that

$$f_a(n_1, n_2, n_3) = \begin{cases} |-2N - (n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2)| & \text{if } -3N < n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2 < -N \\ |n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2| & \text{if } -N < n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2 < N \\ |2N - (n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2)| & \text{if } N < n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2 < 3N \end{cases} \quad (5)$$

$$f_b(n_1, n_2, n_3) = \begin{cases} |-2N - (n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2)| & \text{if } -3N < n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2 < -N \\ |n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2| & \text{if } -N < n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2 < N \\ |2N - (n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2)| & \text{if } N < n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2 < 3N \end{cases}$$

$$f_c(n_1, n_2, n_3) = \begin{cases} |-2N - (n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2)| & \text{if } -3N < n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2 < -N \\ |n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2| & \text{if } -N < n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2 < N \\ |2N - (n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2)| & \text{if } N < n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2 < 3N \end{cases}$$

$$f_d(n_1, n_2, n_3) = \begin{cases} |-2N - (n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2)| & \text{if } -3N < n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2 < -N \\ |n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2| & \text{if } -N < n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2 < N \\ |2N - (n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2)| & \text{if } N < n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2 < 3N \end{cases}$$

Then, the input mapping funtion have the following properties.

$$\begin{aligned} f_a(n_1, n_2, n_3) &= f_a(N_1-n_1, N_2-n_2, N_3-n_3) \\ f_b(n_1, n_2, n_3) &= f_b(N_1-n_1, N_2-n_2, N_3-n_3) \\ f_c(n_1, n_2, n_3) &= f_c(N_1-n_1, N_2-n_2, N_3-n_3) \\ f_d(n_1, n_2, n_3) &= f_d(N_1-n_1, N_2-n_2, N_3-n_3) \end{aligned} \quad (6)$$

Therefore, these mapping funtions are not one to one mapping for all n_1 in Ω_1, for all n_2 in Ω_2, for all n_3 in Ω_3. But, we can obtain the 4Ω integers through the collection of $\Omega_a, \Omega_b, \Omega_c$ and Ω_d, such that

$$\begin{aligned} \Omega_a &= \{ n \mid n = f_a(n_1, n_2, n_3), \; n_1 \in \Omega_1, n_2 \in \Omega_2, n3 \in \Omega_3 \} \\ \Omega_b &= \{ n \mid n = f_b(n_1, n_2, n_3), \; n_1 \in \Omega_1, n_2 \in \Omega_2, n3 \in \Omega_3 \} \\ \Omega_c &= \{ n \mid n = f_c(n_1, n_2, n_3), \; n_1 \in \Omega_1, n_2 \in \Omega_2, n3 \in \Omega_3 \} \\ \Omega_d &= \{ n \mid n = f_d(n_1, n_2, n_3), \; n_1 \in \Omega_1, n_2 \in \Omega_2, n3 \in \Omega_3 \} \end{aligned} \quad (7)$$

This implies that a summation over N indexes in Ω can split into four terms-a summation over the N indexes in Ω_a, a summation over the N indexes in Ω_b, a summation over the N indexes in Ω_c, a summation over the N indexes in Ω_d. Therefore, according to each mapping, we can rewrite (4) as the summation of four DCTs:

$$\begin{aligned} x(k) = {} & 1/4 \sum_{n=\Omega_a} X(n) \cos[\pi(2k+1)n/2N] \\ & + 1/4 \sum_{n=\Omega_b} X(n) \cos[\pi(2k+1)n/2N] \\ & + 1/4 \sum_{n=\Omega_c} X(n) \cos[\pi(2k+1)n/2N] \\ & + 1/4 \sum_{n=\Omega_d} X(n) \cos[\pi(2k+1)n/2N] \end{aligned} \quad (8)$$

We denote, for all (n_1, n_2, n_3) in Ω_a x Ω_b x Ω_c x Ω_d,

$$\begin{aligned} X_a(n_1, n_2, n_3) &= s_a(n) X(n) \big|_{n=f_a(n_1,n_2,n_3)} \\ X_b(n_1, n_2, n_3) &= s_b(n) X(n) \big|_{n=f_b(n_1,n_2,n_3)} \\ X_c(n_1, n_2, n_3) &= s_c(n) X(n) \big|_{n=f_c(n_1,n_2,n_3)} \\ X_d(n_1, n_2, n_3) &= s_d(n) X(n) \big|_{n=f_d(n_1,n_2,n_3)} \end{aligned} \quad (9)$$

for

$$s_a(n) = \begin{cases} 1 & \text{if } -N < n_1N_2N_3 + n_2N_3N_1 + n_3N_1N_2 < N \\ -1 & \text{otherwise} \end{cases}$$

$$s_b(n) = \begin{cases} 1 & \text{if } -N < n_1N_2N_3 + n_2N_3N_1 - n_3N_1N_2 < N \\ -1 & \text{otherwise} \end{cases}$$

$$s_c(n) = \begin{cases} 1 & \text{if } -N < n_1N_2N_3 - n_2N_3N_1 + n_3N_1N_2 < N \\ -1 & \text{otherwise} \end{cases}$$

$$s_d(n) = \begin{cases} 1 & \text{if } -N < n_1N_2N_3 - n_2N_3N_1 - n_3N_1N_2 < N \\ -1 & \text{otherwise} \end{cases}$$

The negative sign appears in according to the following property

$$\cos[\pi(2k+1)(|2N-n|)/2N] = -\cos[\pi(2k+1)n/2N] \quad (10)$$

We can rewrite (8) as

$$\begin{aligned} x(k) = 1/4 \sum_{n_3-1}^{N_3-1} \sum_{n_2-1}^{N_2-1} \sum_{n_1-1}^{N_1-1} \Big[& X_a(n_1,n_2,n_3) * \cos[\pi(2k+1)(n_1N_2N_3+n_2N_3N_1+n_3N_1N_2)/2N] \\ & + X_b(n_1,n_2,n_3) * \cos[\pi(2k+1)(n_1N_2N_3-n_2N_3N_1+n_3N_1N_2)/2N] \\ & + X_c(n_1,n_2,n_3) * \cos[\pi(2k+1)(n_1N_2N_3+n_2N_3N_1-n_3N_1N_2)/2N] \\ & + X_d(n_1,n_2,n_3) * \cos[\pi(2k+1)(n_1N_2N_3-n_2N_3N_1-n_3N_1N_2)/2N] \Big] \end{aligned} \quad (11)$$

Since the DCT for the irrelevant mapping is zero, (11) can be rewritten as

$$\begin{aligned} x(k) = 1/4 \sum_{n_3-1}^{N_3-1} \sum_{n_2-1}^{N_2-1} \sum_{n_1-1}^{N_1-1} X(n_1,n_2,n_3) \, & * \{ \cos[\pi(2k+1)(n_1N_2N_3+n_2N_3N_1+n_3N_1N_2)/2N] \\ & + \cos[\pi(2k+1)(n_1N_2N_3-n_2N_3N_1+n_3N_1N_2)/2N] \\ & + \cos[\pi(2k+1)(n_1N_2N_3+n_2N_3N_1-n_3N_1N_2)/2N] \\ & + \cos[\pi(2k+1)(n_1N_2N_3-n_2N_3N_1-n_3N_1N_2)/2N] \} \end{aligned} \quad (12)$$

where

$$X(n_1,n_2,n_3) = \begin{cases} X_a(n_1,n_2,n_3) = X_b(n_1,n_2,n_3) = X_c(n_1,n_2,n_3) = X_d(n_1,n_2,n_3) & \text{if } n_1 = n_2 = n_3 = 0 \\ X_a(n_1,n_2,n_3) + X_b(n_1,n_2,n_3) + X_c(n_1,n_2,n_3) + X_d(n_1,n_2,n_3) & \text{otherwise} \end{cases} \quad (13)$$

Since

$$\cos(a+b+c) + \cos(a-b+c) + \cos(a+b-c) + \cos(a-b+c) = 4\cos(a)\cos(b)\cos(c) \quad (14)$$

Equation (12) can be reduced to the following

$$x(k) = \sum_{n_3=0}^{N_3-1} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} X(n_1, n_2, n_3) * \cos[\pi(2k+1)n_1/2N_1] * \cos[\pi(2k+1)n_2/2N_2] * \cos[\pi(2k+1)n_3/2N_3] \quad (15)$$

for $n_1 = 0,1,2,\cdots\cdots, N_1-1$
$n_2 = 0,1,2,\cdots\cdots, N_2-1$
$n_3 = 0,1,2,\cdots\cdots, N_3-1$
$k = 0,1,2,\cdots\cdots N-1$

Now, we consider the output mapping which connects $X(k): k = 0,1,2,\cdots\cdots, N-1$, to $x(k_1,k_2,k_3): k_1 = 0,1,2,\cdots\cdots, N_1-1$, $k_2 = 0,1,2,\cdots\cdots, N_2-1$, $k_3 = 0,1,2,\cdots\cdots, N_3-1$. Let k_a be a k modulo $2N_1$, k_b be a k modulo $2N_2$, k_c be a k modulo $2N_3$. We define g to be a mapping from Ω_1 to $\Omega_1 \times \Omega_2 \times \Omega_3$ such that $(k_1,k_2,k_3) = g(k)$ with

$$k_1 = \begin{cases} k_a & \text{if } k_a < N_1 \\ 2N_1 - 1 - k_a & \text{otherwise} \end{cases} \quad (16)$$

$$k_2 = \begin{cases} k_b & \text{if } k_b < N_2 \\ 2N_2 - 1 - k_b & \text{otherwise} \end{cases}$$

$$k_3 = \begin{cases} k_c & \text{if } k_c < N_3 \\ 2N_3 - 1 - k_c & \text{otherwise} \end{cases}$$

for each (k_1, k_2, k_3) in $\Omega_1 \times \Omega_2 \times \Omega_3$ and a k in Ω.

Thus it can be shown that g is a one to one mapping, so there exists the inverse mapping g from $\Omega_1 \times \Omega_2 \times \Omega_3$ to Ω.

$$X(k_1, k_2, k_3) = X(k)\Big|_{k=g^{-1}(k_1\ k_2\ k_3)} \quad (17)$$

If we apply this to (15), we finally obtain the expression in (4),since

$$\cos[\pi(2k+1)n_1/2N_1] = \cos[\pi(2k_1+1)n_1/2N_1] \quad (18)$$
$$\cos[\pi(2k+1)n_2/2N_2] = \cos[\pi(2k_2+1)n_2/2N_2]$$
$$\cos[\pi(2k+1)n_3/2N_3] = \cos[\pi(2k_3+1)n_3/2N_3]$$

We use the 30-point DCT to show the three dimensional-DCT of PFA. Then, from the equation (5), we obtain the input mapping in table I. And from the equation (17), we also obtain the output mapping in table II.

In table I, we know that the collection of four input mapping is 4Ω, that is, $4\Omega=\{0,0,0,0,1,1,1,1, \ldots\ldots, N-1,N-1,N-1,N-1\}$. This verifies that we can obtain the 4Ω through the collection of Ω_a, Ω_b, Ω_c and Ω_d.

Therefore,we must extract the mapping point in equal proportion from each set Ω_a, Ω_b, Ω_c and Ωd, which result in one to one mapping.

Input N_1, N_2, N_3: 2, 3, 5

	n3 = 0			n3 = 1			n3 = 2			n3 = 3			n3 = 4		
n1\n2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
0	0	10	20	6	16	26	12	22	28	18	28	22	24	26	16
1	15	25	25	21	29	19	27	23	13	27	17	7	21	11	1
0	0	10	20	6	16	26	12	22	28	18	28	22	24	26	16
1	15	5	5	9	1	11	3	7	17	3	13	23	9	19	29
0	0	10	20	6	4	14	12	2	8	18	8	2	24	14	4
1	15	25	25	9	19	29	3	13	23	3	7	17	9	1	11
0	0	10	20	6	4	14	12	2	8	18	8	2	24	14	4
1	15	5	5	21	11	1	27	17	7	27	23	13	21	29	19

Table.I Input mapping table for PFA(N=30)

Table.II is the output mapping obtained from the equation (17). These output mapping results in the rearrangement of output sequences. The number of multiplications for N-point DCT can be calculated by using the following formula.

The No. of multiplications for N-point DCT
= (multiplication for N_1-point DCT)$*N_2N_3$
+ (multiplication for N_2-point DCT)$*N_1N_3$
+ (multiplication for N_3-point DCT)$*N_2N_3$

For each N_1-point DCT, N_2-point DCT, N_3-point DCT, we can combine the another algorithm or PFA. Therefore, the complexity depends on those algorithms.

(0,0,0)	——>	x(0)
(0,0,1)	——>	x(11)
(0,0,2)	——>	x(12)
(0,0,3)	——>	x(23)
(0,0,4)	——>	x(24)
(0,1,0)	——>	x(19)
(0,1,1)	——>	x(28)
(0,1,2)	——>	x(7)
(0,1,3)	——>	x(16)
(0,1,4)	——>	x(4)
(0,2,0)	——>	x(20)
(0,2,1)	——>	x(8)
(0,2,2)	——>	x(27)
(0,2,3)	——>	x(3)
(0,2,4)	——>	x(15)
(1,0,0)	——>	x(29)
(1,0,1)	——>	x(18)
(1,0,2)	——>	x(17)
(1,0,3)	——>	x(6)
(1,0,4)	——>	x(5)
(1,1,0)	——>	x(10)
(1,1,1)	——>	x(1)
(1,1,2)	——>	x(22)
(1,1,3)	——>	x(13)
(1,1,4)	——>	x(25)
(1,2,0)	——>	x(9)
(1,2,1)	——>	x(21)
(1,2,2)	——>	x(2)
(1,2,3)	——>	x(26)
(1,2,4)	——>	x(14)

Table.II Output mapping for PFA(N=30)

III. THREE DIMENSIONAL CFA

To be enable the computation of DCT which have the block size N, such that, 2*2*2=8, 2*2*4=16, 2*4*4=32, ..., we derive the mapping algorithm with common factor decomposition. CFA is the algorithm using those common factor mapping.

The mathematical derivation of CFA is the same as PFA. Therefore, in this section, we shows the common factor input and output mappings.

The main goal of CFA is in deriving $\cos[\pi(2k_1+1)n/2N_1] * \cos[\pi(2k_2+1)n/2N_2]*\cos[\pi(2k_3+1)n/2N_2N_3]$ from $\cos[\pi(2k+1)n/2N]$.

This follows that

$$x(k_1,k_2,k_3) = \sum_{n_3=0}^{N_3-1}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} X(n_1,n_2,n_3) * \cos[\pi(2k_1+1)n_1/2N] * \cos[\pi(2k_2+1)n_2/2N_2] * \cos[\pi(2k_3+1)n_3/2N_2N_3] \quad (20)$$

for $n_1,k_1 = 0,1,2,\cdots\cdots, N_1-1$

$n_2,k_2 = 0,1,2,\cdots\cdots, N_2-1$

$n_3,k_3 = 0,1,2,\cdots\cdots, N_3-1$

<Input mapping>

(21)

$f_a(n_1,n_2,n_3) = (n_2N_3N_1 + n_3N_1 + n_1) \bmod N$ if $-3N < n_2N_3N_1 + n_3N_1 + n_1 < -2N$

$|-2N - (n_2N_3N_1 + n_3N_1 + n_1)|$ if $-2N < n_2N_3N_1 + n_3N_1 + n_1 < -N$

$-(n_2N_3N_1 + n_3N_1 + n_1) \bmod N$ if $-N < n_2N_3N_1 + n_3N_1 + n_1 < 0$

$n_2N_3N_1 + n_3N_1 + n_1$ if $0 < n_2N_3N_1 + n_3N_1 + n_1 < N$

$|2N - (n_2N_3N_1 + n_3N_1 + n_1)|$ if $N < n_2N_3N_1 + n_3N_1 + n_1 < 3N$

$f_b(n_1,n_2,n_3) = (n_2N_3N_1 + n_3N_1 - n_1) \bmod N$ if $-3N < n_2N_3N_1 + n_3N_1 - n_1 < -2N$

$|-2N - (n_2N_3N_1 + n_3N_1 - n_1)|$ if $-2N < n_2N_3N_1 + n_3N_1 - n_1 < -N$

$-(n_2N_3N_1 + n_3N_1 - n_1) \bmod N$ if $-N < n_2N_3N_1 + n_3N_1 - n_1 < 0$

$n_2N_3N_1 + n_3N_1 - n_1$ if $0 < n_2N_3N_1 + n_3N_1 - n_1 < N$

$|2N - (n_2N_3N_1 + n_3N_1 - n_1)|$ if $N < n_2N_3N_1 + n_3N_1 - n_1 < 3N$

$f_c(n_1,n_2,n_3) = (n_2N_3N_1 - n_3N_1 + n_1) \bmod N$ if $-3N < n_2N_3N_1 - n_3N_1 + n_1 < -2N$

$|-2N - (n_2N_3N_1 - n_3N_1 + n_1)|$ if $-2N < n_2N_3N_1 - n_3N_1 + n_1 < -N$

$-(n_2N_3N_1 - n_3N_1 + n_1) \bmod N$ if $-N < n_2N_3N_1 - n_3N_1 + n_1 < 0$

$n_2N_3N_1 - n_3N_1 + n_1$ if $0 < n_2N_3N_1 - n_3N_1 + n_1 < N$

$|2N - (n_2N_3N_1 - n_3N_1 + n_1)|$ if $N < n_2N_3N_1 - n_3N_1 + n_1 < 3N$

$f_d(n_1,n_2,n_3) = (n_2N_3N_1 - n_3N_1 - n_1) \bmod N$ if $-3N < n_2N_3N_1 - n_3N_1 - n_1 < -2N$

$|-2N - (n_2N_3N_1 - n_3N_1 - n_1)|$ if $-2N < n_2N_3N_1 - n_3N_1 - n_1 < -N$

$-(n_2N_3N_1 - n_3N_1 - n_1) \bmod N$ if $-N < n_2N_3N_1 - n_3N_1 - n_1 < 0$

$n_2N_3N_1 - n_3N_1 - n_1$ if $0 < n_2N_3N_1 - n_3N_1 - n_1 < N$

$|2N - (n_2N_3N_1 - n_3N_1 - n_1)|$ if $N < n_2N_3N_1 - n_3N_1 - n_1 < 3N$

(22)

$$x_a(n_1,n_2,n_3) = s_a(n)\, x(n)\big|_{n=f_a(n_1,n_2,n_3)}$$
$$x_b(n_1,n_2,n_3) = s_b(n)\, x(n)\big|_{n=f_b(n_1,n_2,n_3)}$$
$$x_c(n_1,n_2,n_3) = s_c(n)\, x(n)\big|_{n=f_c(n_1,n_2,n_3)}$$
$$x_d(n_1,n_2,n_3) = s_d(n)\, x(n)\big|_{n=f_d(n_1,n_2,n_3)}$$

$s_a(n) = -1$ if $-2N < n_2N_3N_1 + n_3N_1+n_1 < -N$ or $N < n_2N_3N_1 + n_3N_1+n_1 < 3N$
1 otherwise

$s_b(n) = -1$ if $-2N < n_2N_3N_1 + n_3N_1-n_1 < -N$ or $N < n_2N_3N_1 + n_3N_{1a}-n_1 < 3N$
1 otherwise

$s_c(n) = -1$ if $-2N < n_2N_3N_1 - n_3N_1+n_1 < -N$ or $N < n_2N_3N_1 - n_3N_1+n_1 < 3N$
1 otherwise

$s_d(n) = -1$ if $-2N < n_2N_3N_1 - n_3N_1-n_1 < -N$ or $N < n_2N_3N_1 - n_3N_1-n_1 < 3N$
1 otherwise

< output mapping >

$k_a = k \bmod N_1$ (23)
$k_b = k \bmod 2N_2$
$k_c = k \bmod 3N_3$

$k_1 = k_a$

$k_2 = k_b$ if $k_b < N_2$
$\quad\; 2N_3 - 1 - k_b$ if $N_2 <= k_b < 2N_2$

$k_3 = k_c$ if $k_c < N_3$
$\quad\; |2N_3 - 1 - k_c|$ if $N_3 <= k_c < 2N_3$
$\quad\; |3N_3 - 1 - k_c|$ if $2N_3 <= k_c < 3N_3$

We use the 8-point DCT to show the three dimensional-DCT of CFA. Then, through the equation (20), we obtain the input mapping in table III,in table VI. And, through the equation (21), we also obtain the output mapping in table V.

In table III, we know that the result of input mapping is 4 Ω, that is, 4Ω={0,0,0,0,1,1,1,1,2,2,...... ,N-1,N-1, N-1,N-1}. This verifies that we can obtain the 4Ω through the collection of Ω_a, Ω_b, Ω_c and Ω_d. Then we extract resulting input table in table VI.

Input N1,N2,N3:2,2,2

	n3 = 0		n3 = 1	
n2/n1	0	1	0	1
0	0	4	1	5
1	2	6	3	7
0	0	4	1	5
1	6	2	7	3
0	0	4	7	3
1	2	6	1	5
0	0	4	7	3
1	6	2	5	1

Table.III Input mapping table for CFA(N=8)

n2/n1	0	1	0	1
0	0	4	1	5
1	6	2	7	3

Table.IV Resulting input mapping for CFA(N=8)

Input N1,N2,N3:2,2,2

	k_3 = 0		k_3 = 1	
k_1\\k_2	0	1	0	1
0	0	6	4	2
1	3	5	7	1

Table.V Output mapping for CFA(N=8)

And we show the signal flowgraph for input mapping in fig.1 and output mapping in fig.2.

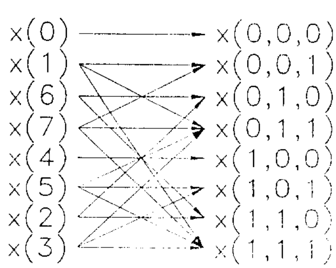

Fig.1 Signal flow graph(CFA) for input mapping

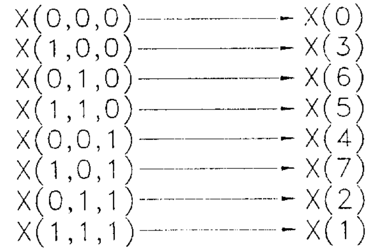

Fig.2 Signal flow graph(CFA) for output mapping (N=8)

Considering input mapping and output mapping, we can draw fig.3.

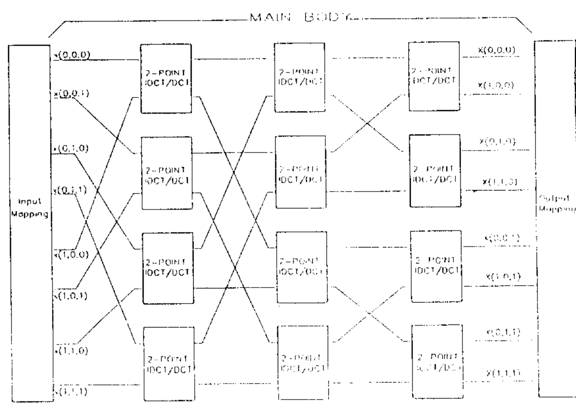

Fig.3 signal flow graph of N-point DCT for CFA(N=8)

VI. Conclusion

In this paper, we have derived the three decomposition algorithms. As a result of it, the larger block size of N is, the fewer number of multiplication for DCT needs.

Employing brute-force techniques, the number of multiplications reduce from N^2 into $N*(N_1+N_2+N_3)$. Ommitting multiplication by 0,1 and the same multiplication, the number of multiplication can be further reduced.

V. REFERENCE

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

[2] N. Suehiro and M. Hatori, " Fast algorithm for the DFT and other sinusoidal transform", IEEE Trans. Acoust, Speech, Signal processing, vol. ASSP-34, pp. 642- 644, Jun. 1986

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

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

[5] B. G. Lee, " A new algorithm to compute the discrete cosine transform", IEEE Trans. Acoust, Speech, Signal processing, Dec. 1984

[6] H. S. Hou, " A fast recursive algorithm for computing the discrete cosine transform", IEEE Trans. Acoust, Speech, Signal processing, pp 1455-1461 Oct. 1987

[7] B. G. Lee, " Input and output index mappings for a prime factor decomposed computation of discrete cosine transform", IEEE Trans. Acoust, Speech, Signal processing, vol. ASSP-37, pp. 237- 244, Feb. 1989

[8] H. S. Malvar, " Fast computation of the discrete cosine transform and discrete Hartly transform", IEEE Trans. Acoust, Speech, Signal processing, vol. ASSP-35, pp. 1484-1485, Oct. 1987

[9] V. Nagesha, " Commants on fast computation of the discrete cosine transform and discrete Hartly transform", IEEE Trans. Acoust, Speech, Signal processing, vol. ASSP-37, pp. 439- 440 Mar. 1989

[10] Z. Wang " Reconsideration of a fast computational algorithm for the discrete cosine transform", IEEE Tran. Commun., vol. COM-31, pp 121-123, Jan, 1983

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

[12] M. Vetterli and H. Nussbaumer, " Simple FFT and DCT algorithms with reduced number of operations", Signal processing, vol. 6, pp. 267-278, Aug. 1984

[13] P. Yip and K. R. Rao, " The dicimation - In frequency algorithms for a family of discrete sine and cosine transforms", Circuits, Systems and Signal process, pp. 4-19, 1988

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

[15] R. N. Bracewell, " The fast Hartly transform", Proc. IEEE vol. 72, pp 1010-1018, Aug, 1984

[16] H. S. Hou, " The fast Hartly transform algorithm", IEEE Trans. Comput, vol. C-36, pp. 147-156, Feb. 1987

[17] H. V. Sorenson, D. L. Jones, C. S. Burrus, and M. T. Heideman, " On computing the discrete Hartly transform", IEEE Tran. Acoust., Speech, and Signal process, vol. ASSP-34, pp 285-295, April, 1986

[18] B Chitprasert and K. R. Rao, " Human visual weighted progressive image transmission", IEEE Trans. Commun, vol. COM-38,1990

[19] N. I. Cho and S. U. Lee, " DCT algorithms for VLSI implementation", IEEE Trans. Acoust, Speech, Signal processing, Jan. 1990

[20] P. Duhamel, " Implementation of Split-Radix FFT algorithms for comlex, real and real symmatric data", IEEE Trans. Acoust, Speech, Signal processing, vol. ASSP-34, pp. 285-295, Apr. 1986

[21] C. S. Burus,「 Index mappings for multidimensional formulation of the DFT and convolutions,」 IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-25, pp. 239-242, June 1977.

[22] C. S. Burus and P. W. Escanbacher, 「 An in-place in-order prime factor FFT algorithm,」 IEEE Trans. Acous. Speech, Signal Processing, vol.ASSP-29, pp. 806-817, Aug. 1981.

[23] S. Chu and C. S. Burus, 「A prime factor algorithm using distributed arithmetics,」 IEEE Trans. Acous. Speech, Signal Processing, vol.ASSP-30, pp. 217-227, Apr. 1982.

[24] C. M. Rader, 「Discrete Fourier transforms when the number of data samples is prime」, Proc. IEEE, vol. 56, pp. 1107-1108, June 1968.