

VLSI 구현을 위한 정수화 DCT 개발

* 正會員 郭 勳 星* 正會員 李 宗 河**

Development of Integer DCT for VLSI Implementation

Hoon Sung Kwak*, Jong Ha Lee** *Regular Members*

要 約

본 논문에서는 VLSI 구현을 위하여 IC의 구조를 간단하게 하고 정수 연산을 수행하는 정수화 DCT에 대한 고속 알고리즘을 제안하였다.

정수화 DCT의 고속 알고리즘은 Chen의 행렬 분해 방식을 사용하여 구현하였다. 이 고속 알고리즘은 직접적인 정수화 DCT 계산방식에 비하여 덧셈과 곱셈수의 연산수가 크게 감소하였으며, 덧셈수는 DCT의 고속 알고리즘의 경우와 같으며, 곱셈수는 N가 8일 때는 DCT의 고속 알고리즘의 경우와 같지만 N가 8보다 클 경우 곱셈수가 현저하게 감소한다. 뿐만아니라 유한길이 연산으로 인한 DCT의 성능 저하를 극복할 수 있다.

ABSTRACT

This paper presents a fast algorithm of integer discrete cosine transform(IDCT) allowing VLSI implementation by integer arithmetic.

The proposed fast algorithm has been developed using Chen's matrix decomposition in DCT, and requires less number of arithmetic operations compared to the IDCT. In the presented algorithm, the number of addition number is the same as the one of Chen's algorithm in DCT, and the number of multiplication is the same as that in DCT at $N=8$ but drastically decreasing when N is above 8. In addition, the drawbacks of DCT such as performance degradation at the finite length arithmetic could be overcome by the IDCT.

※이 논문은 1992년도 산학협동재단의 학술연구비 지원에 의하여 연구되었음

* 全北大學校 컴퓨터工學科
 ** 全州工學專門大學 電子科
 論文番號 : 93-192

I. 서 론

디지털 영상 데이터의 많은 중복성을 줄이기 위해

서는 변환 방식이 많이 사용되고 있다⁽¹⁾⁽²⁾. 여러 변환 방식중에서 DCT는 상관성이 많은 영상 데이터에 적용하는 경우 최적변환 방식인 KLT와 유사한 성능을 얻을 수 있어 그동안 발표된 여러 고속 알고리즘^{(3) (6)}과 집적회로의 제조 기술 발달로 인하여 실시간 DCT의 구현이 가능하게 되었다⁽⁷⁾.

DCT의 구현은 사용한 IC에 따라서 TTL IC를 사용한 구현⁽⁸⁾, VLSI 제조 기술을 이용한 DCT 전용 소자에 의한 구현⁽⁷⁾, 그리고 범용의 VLSI인 DSP소자를 이용한 구현⁽⁹⁾들이 있는데 이러한 소자를 이용할 경우 DCT의 행렬 요소가 실수이기 때문에 유한길이연산(finite-length arithmetic)으로 발생하는 영상 재구성 오차를 무시할 수 있도록 DCT 계수를 표현하는 데는 적어도 8 비트(bit)가 필요하다⁽¹⁰⁾. 그러나 변환 행렬 요소가 정수이면 절삭 오차(truncation error)를 갖지 않고 적은 비트 수로 DCT 계수를 표현할 수 있으며, 정수 연산을 수행하고 구현시 IC의 구조를 간단하게 할 수 있다.

기존의 정수화 변환인 WHT(Walsh-Hadamard transform)⁽¹²⁾는 비교적 간단하게 하드웨어로 구현시킬 수 있으며 변환시 덧셈 연산만을 수행하므로 처리속도가 빠른 장점이 있으나 변환계수의 상관도가 낮아 성능이 좋지않다. 그리고 정수화 변환인 CMT(C-matrix transform)⁽¹¹⁾는 DCT에 WHT를 적용한 변환 행렬을 도입하고, 이 행렬의 블록 대각 구조(block diagonal structure)를 이용하여 이를 정수화한 것으로서 정수 연산을 수행하고 연산과정에서 나타내는 곱셈수와 덧셈수를 줄임으로써 연산속도는 개선되었으나 DCT에 비해 성능이 떨어진다는 단점을 갖고 있다. 또한 DCT의 실수 연산을 간단하게 하면서 변환과정에서 발생하는 오차로 인한 성능 저하를 줄이는 연구도 있어 왔으나⁽¹²⁾ 근본적으로 절삭오차를 해결하지는 못하였다. 그래서 DCT에 근사한 성능을 가지면서 정수 연산만을 수행하는 정수형 변환에 대한 연구가 계속되어 왔다. Cham 등은 dyadic symmetry 원리를 이용하여 DCT에 근사한 성능을 가지는 ICT(integer cosine transform)을 제안하였으며^{(13), (14)}, Kwak 등은 DCT의 구조 특성을 이용하여 DCT 보다 좋은 특성을 나타내는 새로운 정수화 변환 IDCT(integer DCT)을 개발하였다⁽¹⁵⁾.

본 논문에서는 VLSI 제조 기술을 이용한 DCT 전용 소자의 구현시 정수 연산을 수행하고 IC의 구조를 간단하게 하는 정수화 DCT(IDCT)에 대한 고속처리 알고리즘을 제시하고 이들의 연산 횟수를 다른 변환

들과 비교한다.

II. 정수화 DCT

DCT 행렬에 대하여 행렬 분해를 실시하면 각 분해 행렬들은 직교성을 가지며, 각 분해 행렬의 첫번째 열 벡터(row vector)의 요소 값들이 크기 순서로 배열되며, 그 요소 값들이 각 분해 행렬 내에서 중복되어 구성되는 특성을 갖는다. 이 특성을 이용하여 정수화 DCT가 구현된다.

$N \times 1$ 데이터 벡터 $[f]$ 에 대한 정수화 DCT(integer DCT; IDCT)와 그 역변환(inverse IDCT; IIDCT)은 각각 식(1), (2)와 같이 정의된다⁽¹⁵⁾.

$$[F_1] = [K_N][\Phi_N][f] \quad (1)$$

$$[f] = [\Phi_N]^T [K_N][F_1] \quad (2)$$

여기에서 $[\Phi_N]$ 은 정수값을 가지며 DCT의 직교성과 크기 순서를 이용하여 구성된 정수화 DCT 행렬이다. $[K_N]$ 는 정수화 DCT 행렬 $[\Phi_N]$ 의 열을 scale하여 정규 직교 행렬이 되도록 하는 대각 행렬 즉, $[K_N] = \text{diag}\{K_N\}$ 인 크기 인자(scale factors)이다.

정수화 DCT $[\Phi_N]$ 은 식(3)과 같이 DCT 행렬에 대응되는 분해 행렬로 나타낼 수 있다.

$$\begin{aligned} [\Phi_N] &= [P_N][\bar{\Phi}_N][B_N] \\ &= [P_N][\text{diag}\{\text{diag}\{\dots\{\text{diag}\{\text{diag}\{[D_{N/2}^{n-1}], \\ &\quad [S_{N/2}^{n-1}]\}\}[B_{N/2}^{n-2}], [S_{N/2}^{n-2}]\}\}[B_{N/2}^{n-3}]\} \\ &\quad \dots [S_{N/4}]\}[B_{N/2}], [Q_{N/2}]\}\}[B_N] \end{aligned} \quad (3)$$

여기서 $n = \log_2 N$ 이며, $[\bar{\Phi}_N]$ 은 $[\Phi_N]$ 을 비트 역순으로 재 배열한 행렬을 의미한다.

정수화 DCT 행렬의 각 분해 행렬($[D_{N/2}^{n-1}], [S_{N/2}^{n-1}], [S_{N/2}^{n-2}], \dots, [S_{N/4}], [Q_{N/2}]$)들은 DCT 행렬의 분해 행렬($[A_{N/2}^{n-1}], [O_{N/2}^{n-1}], [O_{N/2}^{n-2}], \dots, [O_{N/4}], [R_{N/2}]$)에 각각 대응되는 행렬이며, 분해 행렬 요소들은 DCT 행렬의 분해 행렬 요소들의 크기 순서를 갖는 정수값 들로 구성된다.

그러므로 각 분해 행렬 요소들의 정수값을 구하는 방법은 먼저, 분해 행렬 $[D_{N/2}^{n-1}]$ 의 행렬 요소에 대응하는 DCT 행렬의 분해 행렬인 $[A_{N/2}^{n-1}]$ 의 행렬 요소들이 같은 값으로 구성되어 있으므로 정수값 중에서 고속 알고리즘 구현에 유리한 값 1을 택하여 구성하였다. 그리고 나머지 분해 행렬($[S_{N/2}^{n-1}], [S_{N/2}^{n-2}], \dots, [S_{N/4}], [Q_{N/2}]$)의 행렬 요소들은 DCT 행렬의 분해 행렬인 $[O_{N/2}^{n-1}], [O_{N/2}^{n-2}], \dots, [O_{N/4}], [R_{N/2}]$ 의 행렬 요소들이 같은 값으로 구성되어 있으므로 정수값 중에서 고속 알고리즘 구현에 유리한 값 1을 택하여 구성하였다. 그리고 나머지 분해 행렬($[S_{N/2}^{n-1}], [S_{N/2}^{n-2}], \dots, [S_{N/4}], [Q_{N/2}]$)의 행렬 요소들은 DCT 행렬의 분해 행렬인 $[O_{N/2}^{n-1}], [O_{N/2}^{n-2}], \dots, [O_{N/4}], [R_{N/2}]$ 의 행렬 요소들이 같은 값으로 구성되어 있으므로 정수값 중에서 고속 알고리즘 구현에 유리한 값 1을 택하여 구성하였다. 그리고 나머지 분해 행렬($[S_{N/2}^{n-1}], [S_{N/2}^{n-2}], \dots, [S_{N/4}], [Q_{N/2}]$)의 행렬 요소들은 DCT 행렬의 분해 행렬인 $[O_{N/2}^{n-1}], [O_{N/2}^{n-2}], \dots, [O_{N/4}], [R_{N/2}]$ 의 행렬 요소들이 같은 값으로 구성되어 있으므로 정수값 중에서 고속 알고리즘 구현에 유리한 값 1을 택하여 구성하였다.

2^{n-2} , ..., $[S_{N/4}]$, $[Q_{N/2}]$)들에 대한 분해 행렬 요소들은 DCT 행렬의 직교성과 크기 순서를 갖는 변수(variable)들로 구성하고 정수값을 구한다.

직교성과 크기 순서를 만족하는 정수값의 집합은 실제 여러개 존재할 수 있으므로 최적의 정수값을 구하기 위하여 이들 정수값의 집합들로 정수화 DCT 행렬을 구성하여 Hilbert-Schmidt norm⁽¹⁶⁾이 가장 최소가 되는 정수값을 선정한다.

최적의 정수값을 구하기 위하여 Hilbert-Schmidt norm을 사용하는 것은 norm이 작을수록 변환의 성능이 최적 변환인 KLT에 더 근접한 성능을 갖기 때문이다.

Hilbert-Schmidt norm은 입력 신호가 1차 Markov 신호라 가정하고 공분산 행렬을 $[T]$, 정수화 DCT를 $[\Phi]$, $T' = \Phi T \Phi^{-1}$, $T_U' = \text{diag}(\Phi T \Phi^{-1}) = \text{diag}(T_{11}', T_{22}', \dots, T_{mm}')$, $T_U = \Phi^{-1} T_U' \Phi$ 이면 식(4)와 같다.

$$|T - T_U|^2 = \frac{1}{N} \left[\sum_{m,n=0}^{N-1} |(T - T_U)_{m,n}|^2 \right]$$

$$|T|^2 = -\frac{1}{N} \sum_{m=0}^{N-1} |(\Phi T \Phi^{-1})_{m,n}|^2 \quad (4)$$

정수화 DCT 행렬 $[\Phi_N]$ 의 정수값을 구하는 과정을 $N=8$ 일때를 예로하여 설명하면 다음과 같다. 먼저, 정수화 DCT 행렬 $[\Phi_8]$ 의 분해 행렬은 일반성을 갖는 정수화 DCT 행렬 $[\Phi_N]$ 의 분해 행렬이 DCT 행렬의 분해 행렬에 대응되는 크기 순서를 갖도록 변수들을 사용하여 정리하면 식(5)~(7)과 같다.

$$[D_2] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5)$$

$$[S_2] = \begin{bmatrix} a_1 & a_2 \\ a_2 & -a_1 \end{bmatrix} \quad (6)$$

$$[S_4] = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_3 & -b_1 & b_4 & b_2 \\ b_2 & -b_4 & -b_1 & -b_3 \\ b_4 & -b_3 & b_2 & -b_1 \end{bmatrix} \quad (7)$$

식(6), (7)의 분해 행렬 요소가 DCT의 직교 조건과 크기 순서를 만족하는 정수값을 갖기 위한 조건은

다음과 같다.

$$b_1 b_2 = b_1 b_3 + b_2 b_4 + b_3 b_4 \quad (8)$$

$$a_1 > a_2 \quad (9)$$

$$b_1 > b_2 > b_3 > b_4 \quad (10)$$

$$b_1 > a_1 \quad (11)$$

직교 조건과 크기 순서를 갖는 정수화 DCT 행렬 $[\Phi_8]$ 의 정수값을 구하면, 분해 행렬인 식(6), (7)의 행렬 요소가 서로 다른 값을 갖으므로 먼저, 식(8)의 직교 조건과 식(10)의 크기 순서를 만족하는 분해 행렬 요소 b_1, b_2, b_3, b_4 를 구한 다음 식(9), (11)을 만족하는 행렬 요소 a_1 과 a_2 를 구한다. 그런데 식(8)과 식(10)을 만족하는 정수값을 $H_1 = \{b_1, b_2, b_3, b_4\}$ 이라 할 때 여러 개의 H_1 이 존재하며, a_1 과 a_2 는 식(8)의 직교 조건과 독립이기 때문에 식(9), (10)을 만족하는 임의의 정수값을 취해야 한다.

최적의 정수값을 구하기 위하여 H_1 과 a_1, a_2 를 이용하여 정수화 DCT를 구성하고 Hilbert-Schmidt norm이 가장 작은 정수값을 선정하면 각각 $a_1 = 7, a_2 = 3, b_1 = 10, b_2 = 9, b_3 = 6, b_4 = 2$ 가 되며, 하드웨어 구현시 4비트로 구성할 수 있는 장점을 갖는다.

이들 정수값을 이용하여 분해 행렬 $[[D_2], [S_2], [S_4]]$ 을 구성하고 이 분해 행렬을 식(3)에 대입하고 정리하면 식(12)의 정수화 DCT 행렬 $[\Phi_8]$ 을 얻을 수 있으며, 이 행렬을 scale하여 정규 직교 행렬로 만드는 대각 행렬 $[K_8]$ 은 식(13)과 같다.

$$[\Phi_8] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 7 & 3 & -3 & -7 & -7 & -3 & 3 & 7 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 3 & -7 & 7 & -3 & -3 & 7 & -7 & 3 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix} \quad (12)$$

$$[K_8] = \text{diag}\{k_1, k_2, k_3, k_2, k_1, k_2, k_3, k_2\} \quad (13)$$

$$k_1 = 1/\sqrt{8}, \quad k_2 = 1/\sqrt{442}, \quad k_3 = 1/\sqrt{232}$$

III. 정수화 DCT의 고속 알고리즘

정수화 DCT의 실현을 위한 하드웨어 구조는 행렬의 벡터 곱을 직접 실현 시키는 구조와 정수화 DCT의 고속 알고리즘을 만들고, 이 고속 알고리즘을 근거로하여 표현되는 신호의 흐름도를 관찰하고 기본 연산의 구조를 찾아내어 이를 고속으로 실현 시킬 수 있는 구조로 나누어 생각할 수 있다.

예로서, 크기가 $N \times 1$ 인 1차원 정수화 DCT에 대해서 $N=8$ 인 경우는 식(14)와 같이 나타낼 수 있다.

$$\begin{bmatrix} F_1(0) \\ F_1(1) \\ F_1(2) \\ F_1(3) \\ F_1(4) \\ F_1(5) \\ F_1(6) \\ F_1(7) \end{bmatrix} = [K_8] \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ B_1 & B_2 & B_3 & B_4 & -B_4 & -B_3 & -B_2 & -B_1 \\ A_1 & A_2 & -A_2 & -A_1 & -A_1 & -A_2 & A_2 & A_1 \\ B_2 & -B_4 & -B_1 & -B_3 & B_3 & B_1 & B_4 & -B_2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ B_3 & -B_1 & B_4 & B_2 & -B_2 & -B_4 & B_1 & -B_3 \\ A_2 & -A_1 & A_1 & -A_2 & -A_2 & A_1 & -A_1 & A_2 \\ B_4 & -B_3 & B_2 & -B_1 & B_1 & -B_2 & B_3 & -B_4 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix} \tag{14}$$

여기서, $A_1=7, A_3=3, B_1=10, B_2=9, B_3=6, B_4=2$ 이다.

정수화 DCT의 고속 알고리즘을 구현하기 위하여 정수화 DCT 행렬 $[Φ_N]$ 은 Chen등이 제시한 행렬 분해 방식⁽⁴⁾을 사용하여 $N=8$ 일때 식(15)와 같이 나타낼 수 있다.

$$[Φ_8] = [P_8] \left[\begin{array}{c|c} Φ_4 & 0 \\ \hline 0 & Q_4 \end{array} \right] [B_8] \tag{15}$$

여기서 $[P_8]$ 은 변환 계수 벡터를 비트 역순에서 정순서로 바꾸어 주는 행렬이고 $[B_8]$ 은 단일 행렬 $[I_4]$ 로 이루지는 행렬이다. 그리고 $[Q_4]$ 는 그림 1과 같이 3개의 행렬로 분해되어 0을 많이 포함하는 행렬로 표시된다. 즉, 정수화 DCT에 Chen의 행렬 분해 방식을 이용할 경우 DCT의 경우와 똑 같이 크기가 $N \times N$ 인 정수화 DCT 행렬의 $[Q_{N/2}]$ 을 $2\log_2 N - 3$ 개의 sparse 행렬로 분해하여 이들의 곱으로 표시함으로써 곱셈 계산을 감소시킬 수 있다. $N=8$ 일때의 정수화 DCT 행렬의 분해 과정 및 신호 흐름도는 각각 그림 1과 그림 2에 나타내었다. 그림1, 그림2에서 정규화

$$[F_8] = [P_8] \left[\begin{array}{c|c} Φ_4 & 0 \\ \hline 0 & Q_4 \end{array} \right] [B_8][f_8]$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ A_1 & A_2 & -A_2 & -A_1 & 0 & 0 & 0 & 0 \\ A_2 & -A_1 & A_1 & -A_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & B_4 & B_3 & B_2 & B_1 \\ 0 & 0 & 0 & 0 & B_2 & B_4 & -B_1 & B_3 \\ 0 & 0 & 0 & 0 & -B_3 & -B_1 & -B_4 & B_2 \\ 0 & 0 & 0 & 0 & -B_1 & B_2 & -B_3 & B_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}$$

⇓

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & B_4 & 0 & 0 & B_1 \\ 0 & 0 & 0 & 0 & 0 & B_2 & B_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -B_3 & B_2 & 0 \\ 0 & 0 & 0 & 0 & -B_1 & 0 & 0 & B_4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_2 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -A_1 & A_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & P_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -P_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -P_2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & P_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$P_1 = (B_2 + B_3) / 2B_1, P_2 = (B_1 - B_4) / 2B_3$

그림 1. 정수화 DCT 행렬의 분해 과정
Fig. 1. Process of decomposing IDCT matrix

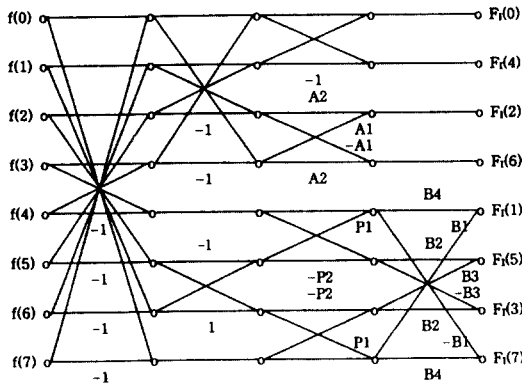


그림 2. 고속 알고리즘에 의한 정수화 DCT의 신호 흐름도 (N=8)

Fig. 2. Signal flowgraph for integer DCT by fast algorithm(N=8)

계수 $[K_8]$ 은 포함되지 않은 것이다. 이것은 DCT의 경우와 마찬가지로 정수화 DCT의 부호화 과정중 부호화기의 양자화 부분과 복호화기의 역양자화 부분에서 흡수할 수 있기 때문이다.

N 개의 입력에 대한 DCT와 정수화 DCT의 고속 알고리즘에 의해 얻은 연산 횟수는 다음과 같다. 먼저, DCT의 경우 Chen의 고속 알고리즘을 사용한 경우 덧셈 수 K_A 는 식(16)에 곱셈 수 K_{MC} 는 식(17)에 나타내었다⁴⁾.

$$K_A = \frac{3N}{2} (\log_2 N - 1) + 2 \quad (16)$$

$$K_{MC} = N \log_2 N - \frac{3N}{2} + 4 \quad (17)$$

그리고 정수화 DCT의 고속 알고리즘을 사용한 경우 덧셈 수는 DCT의 고속 알고리즘의 경우와 같으며 곱셈 수는 N=8일때는 DCT의 고속 알고리즘의 경우와 같이 16개 이지만 N=8 보다 클때는 곱셈수가 현저하게 감소하는데 N가 8보다 클때의 곱셈 수 K_{MI} 은 식(18)과 같다.

$$K_{MI} = 16 + 24 \cdot 2^M \quad (18)$$

여기서, $M = \log_2 N - 4, N \geq 16$

위의 계산식을 토대로 N=8, 16인 경우 1차원 정수화 DCT, DCT의 고속 알고리즘과 CMT에 대하여 얻은 연산 횟수를 표1에 나타내었다. 표1에 나타난

바와 같이 정수화 DCT의 덧셈 수는 DCT의 경우와 같으나 곱셈수는 줄어든다.

이와 같이 정수화 DCT에 대한 고속 알고리즘의 곱셈 수가 적어지는 것은 정수화 DCT를 구현하는 과정에서 DCT 행렬의 행렬 요소가 같은 값으로 구성된 열 벡터에 대하여 고속 알고리즘 구현에 유리한 정수값 1을 택하므로써 곱셈 수를 줄일 수 있었다. 그리고 정수화 DCT의 분해 행렬인 $Q_{N/2}$ 는 Chen의 행렬 분해 방법을 적용하여 $2 \log_2 N - 3$ 개의 sparse 행렬로 분해하여 이들의 곱으로 나타낼 때 행렬 요소가 정수값들로 구성되므로 그림1과 같이 P1, P2, ..., Pm($m = N/4$)를 도입하므로써 N=8일때는 전체 곱셈 수가 DCT의 고속 알고리즘의 경우와 같게 되나 N이 8보다 클때에는 곱셈수가 현저하게 줄어드는 반면 덧셈 수는 같게 된다. 덧셈 수가 같다는 것은 신호 흐름도에 나타난 바와 같이 연산단계(stage)가 같다는 것을 의미한다.

표 3-1. 정수화 DCT, DCT의 고속 알고리즘과 CMT에 대한 연산 횟수 비교.

Table 3-1. Comparison of number of addition and multiplications required for fast algorithm of IDCT, DCT and CMT.

변 환		N = 8		N = 16	
		덧셈 수	곱셈 수	덧셈 수	곱셈 수
IDCT	직접 계산	56	64	240	256
	고속 알고리즘	26	16	74	40
DCT	직접 계산	56	64	240	256
	고속 알고리즘	26	16	74	44
CMT		34	16	106	56

IV. 결 론

본 논문에서는 VLSI 제조 기술을 이용한 DCT 전용 소자의 구현시 정수 연산을 수행하고 IC의 구조를 간단하게 하는 정수화 DCT에 대한 고속 알고리즘을 제안하였다.

정수화 DCT는 DCT의 직교성이 크기 순서를 이용하여 구현한 정수형 변환이므로 DCT의 특성을 가지면서 정수값들로 구성되기 때문에 DCT의 유효길이 연산으로 발생하는 절삭오차로 인한 성능 저하 및 부

동 소숫점 연산으로 인한 하드웨어의 복잡성과 연산 속도를 줄일 수 있다.

정수화 DCT의 고속 알고리즘은 Chen등이 제시한 행렬 분해 방식을 사용하여 구현하였다. 이 고속 알고리즘은 직접적인 정수화 DCT 계산방식에 비하여 덧셈과 곱셈의 연산수가 크게 감소하였으며, 이 알고리즘의 연산수를 DCT의 고속 알고리즘과 비교하였을 때, $N=8$ 일때는 연산수가 같았으나, N 이 8보다 클 경우, 덧셈수는 같은 반면 곱셈수는 현저하게 줄었다. 또한 DCT 고속 알고리즘의 경우는 신호흐름도 상에서 연산 단계마다 연산오차가 발생하는 반면, 정수화 DCT와 고속 알고리즘은 연산오차가 아주 적게 발생한다.

참 고 문 헌

1. A.K. Jain, "Image data Compression: A Review," Proc. IEEE, vol. 69, No. 3, pp.349-398, Mar. 1981.
2. R. J. Clarke, "Transform Coding of Image," Academic Press, 1985.
3. K.R. Rao, P. Yip, "Discrete Cosine Transform Algorithms, Advantages, Applications," Academic press, 1990.
4. W.H. Chen, C.H. Smith, and S.C. Fralic, "A Fast Computational algorithm for the Discrete Cosine Transform," IEEE Trans. on Communications, Vol. COM-25, No. 9, pp.1004-1009, Sep. 1977.
5. John Makhoul, "A Fast Cosine Transform in One and Two Dimensions," IEEE Trans. on ASSP, vol. ASSP-28, No.1, pp. 27-34, Feb. 1980.
6. M.A.Haque, "A Two Dimensional Fast Cosine Transform," IEEE Trans. on ASSP, vol. ASSP-33, No.6, pp. 1532-1539, Dec. 1985.
7. A. Artieri, S. Kritter, F. Jutand, N. Demassieux, "A One chip VLSI for Real Time Two-Dimensional Discrete Cosine Transform," ISCAS, pp. 701-704, 1988.
8. Ali Jalali and K.R.Rao, "A High Speed FDCT Processor for Real-Time Processing of NTSC Color TV Signal," IEEE Trans on Electromagnetic Compatibility, vol. EMC-24, No.2, pp. 278-286, May 1982.
9. TMS320C25x User's Guide, Texas Instruments, 1988.
10. M.Guglielmo, "An Analysis of Error Behavior in the Implementation of 2-D Orthogonal Transforms," IEEE Trans. on Communi., vol. COM-30, No.9, pp. 973-975, Sep. 1986.
11. H. W. Jones, D. H. Hein, and S. C. Knauer, "The Karhunen-Loeve, discrete cosine, and related transforms obtained via the Hadamard transform," Int. Telemetry Conf., Los Angeles, Nov. pp.14-16, 1978.
12. 전준현, 고종석, 김성대, 김재균, "고속 DCT 변환 방식의 정수형 연산에 관한 연구," 한국통신학회 논문지, Vol.12, No.4, Aug. 1987.
13. W.K.Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," IEE Proc. Inst. Elect. Eng. Part I, pp. 276-282, Aug. 1989.
14. W.K.Cham and Y.T.Chan, "An Order-16 Integer Cosine Transform," IEEE Trans. on Signal Processing, vol. 39, No. 5, May 1991.
15. M.S.Park, I.J.Song, J.H.LEE, H.S.Kwak, "New integer cosine transform," Asia-Pacific Conference on Communications, pp. 801-805, Aug. 1993.
16. M. Hamidi, J. Pearl, "Comparison of the cosine and Fourier transform of Markov-1 signals," IEEE Acoust., Speech, Signal Process. Trans. ASSP-24, pp. 248-429, 1976.



郭勳星(Hoon Sung Kwak) 正會員
1944年 6月 27日生
1964年~1971年 2月: 全北大學校
電氣工學科 卒業(工學士)
1971年~1978年 8月: 全北大學校
大學院 卒業(工學碩士, 博士)
1981年~1982年 8月: 美國텍사스大
學校 客員教授

1978年~現在: 全北大學校 컴퓨터공학과 教授
1992年~現在: 全北大學校 電子計算所 所長
※주관심분야: 영상처리, 패턴인식 및 인공지능



李宗河(Jong Ha Lee) 正회원
1959년 3월 19일생
1982년 2월: 전북대학교 전기공학
과(공학사)
1984년 2월: 전북대학교 대학원 전
기공학과(공학석사)
1993년 8월: 전북대학교 대학원 전
자공학과(공학박사)

1990년 3월~ 현재: 전주공업전문대학 전자과 조교수
※주관심분야: 디지털 신호 및 영상처리, 컴퓨터 비전