

# 고정 소수점 연산시 오차의 전파를 줄이는 고속 이산 여현 변환 알고리즘

정회원 정 연 식\*, 이 임 건\*\*, 최 영 호\*, 박 규 태\*

## A Fast DCT Algorithm with Reduced Propagation Error in the Fixed-Point Computation

Yeonsik Jeong\*, Imgeun Lee\*\*, Youngho Choi\*, Kyu Tae Park\* *Regular Members*

### 요 약

이산 여현 변환(Discrete Cosine Transform: DCT)은 음성 및 영상 신호의 압축에 광범위하게 응용되고 있다. 본 논문에서는  $2^m$ -포인트의 일반적인 경우로 확장이 가능한 새로운 고속 DCT 알고리즘과 구조를 제안한다. 제안한 알고리즘은 커널의 대칭성을 이용하여  $N$ -포인트의 DCT를  $N/2$ -포인트의 DCT로 나누어 처리하며 이를 재귀적으로 적용해 나간다. 제안한 알고리즘은 적은 덧셈 및 곱셈 연산을 통해 변환을 수행하며, 변환을 위해 통과해야 하는 곱셈 연산단의 수가 적고 대부분의 곱셈 연산이 흐름도상의 후반부에서 일괄적으로 수행되므로 고정 소수점 연산 시에 발생할 수 있는 오차의 전파를 줄일 수 있다.

### ABSTRACT

Discrete Cosine Transform(DCT) has wide applications in speech and image coding. In this paper, we propose a novel fast DCT scheme with the property of reduced multiplication stages and the smaller number of additions and multiplications. This exploits the symmetry property of the DCT kernel to decompose the  $N$ -point DCT to  $N/2$ -point, and can be generally applied recursively to  $2^m$ -point.

The proposed algorithm has a structure that most of multiplications tend to be performed at final stage, and this reduces propagation of truncation error which could occur in the fixed-point computation. Also the minimization of the multiplication stages further decreases the error.

### I. 서 론

영상 처리 시스템은 영상 신호의 특성상 많은 양의 데이터를 처리하고 전송해야 하므로 신호의 압축과 복원을 바탕으로 하는 부호화에 관한 기술이 필수적이다. 영상 신호의 압축에 많이 이용되고 있는 변환 부호화는 상관 관계 제거 효과와 좁은 주파수 대역으로

\* 연세대학교 전자공학과

\*\* 대우전자 디지털 TV 연구소 선임연구원

論文番號 : 98092-0303

接受日字 : 1998年 3月 3日

에너지를 재배치하는 특성을 이용하여 주파수 대역별로 재배치된 정보에 대해 각 대역별로 가중치를 달리 하여 양자화함으로써 원하는 압축 효과를 얻을 수 있다. 이와 같은 관점에서 최적의 변환은 KLT로 알려져 있으나 이 변환은 계산량이 너무 많고 입력 신호에 따라 기저가 바뀌게 되어 실제 시스템에 적용하기에는 부적절하다. 따라서 KLT는 다른 변환들의 성능 비교를 위한 이론적 기준을 제공한다[1].

영상 처리에서는 DCT가 KLT를 대신하여 많이 이용된다. DCT는 영상 신호와 같이 자기 상관 계수가 큰 AR1 모델에 대해서는 KLT와 거의 유사한 성능을 나타낸다. 그러나 DCT의 구현을 위해서는 많은 덧셈과 곱셈 연산이 필요하게 되므로 DCT의 실시간 처리를 위해서는 고속 알고리즘이 필수적이다[2].

현재까지 발표된 고속 DCT 알고리즘은 크게 3가지로 분류되는데 FFT나 WHT를 이용한 간접 계산 방식과 행렬의 분해나 재귀적 방법을 이용한 직접 계산 방식 그리고 최적 연산법등이 있다[3][4][5]. 일반적으로 직접 계산 방식이 간접 계산 방식에 비해 적은 수의 연산을 필요로 하고 시스템으로 구현하기도 용이하므로 많이 이용된다.

본 논문에서는 DCT 커널의 대칭성을 이용하여, 적은 수의 덧셈 및 곱셈 연산을 통해 구현되는 새로운 고속 DCT 알고리즘을 제안한다. 제안한 알고리즘은 8-포인트 DCT에 대해 28번의 덧셈 연산과 12번의 곱셈 연산을 수행하며, 또한 모든 DCT 계수들이 2번의 곱셈 연산단을 통과하여 얻어지므로 기존의 알고리즘[6]에 비해 고정 소수점 연산시 발생하는 오차를 줄일 수 있는 구조를 가지고 있다.

본 논문의 구성은 다음과 같다. 2장에서 제안한 알고리즘의 수학적적인 유도를 제시하고, 3장에서 제안한 알고리즘의 여러 특성을 살펴본다. 4장에서는 고정 소수점 연산을 수행하였을 경우 발생할 수 있는 오차에 대해 분석하고 실험을 통하여 제안한 알고리즘의 성능을 평가한 후, 5장에서 결론을 맺는다.

## II. 알고리즘의 유도

본 장에서는  $N$ -포인트 DCT에 대한 순환 알고리즘을 제안한다. 입력 신호의 시퀀스를  $x(k)$ ,  $k = 0, 1, \dots, N-1$ , DCT 변환된 시퀀스를  $X(n)$ ,  $n = 0, 1, \dots, N-1$ 이라 할 때, 변환 쌍은 다음과 같이 표현되며

$$x(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} e(n)X(n) \cos \frac{\pi(2k+1)n}{2N}, \quad (1)$$

$$k = 0, 1, \dots, N-1,$$

$$X(n) = \sqrt{\frac{2}{N}} e(n) \sum_{k=0}^{N-1} x(k) \cos \frac{\pi(2k+1)n}{2N}, \quad (2)$$

$$n = 0, 1, \dots, N-1,$$

이 때  $e(0) = 1/\sqrt{2}$ 이고,  $n = 1, 2, \dots, N-1$ 일 때  $e(n) = 1$ 이다. 수식 전개에 편리를 위해  $\hat{X}(n)$ 을 다음과 같이 정의할 때

$$\hat{X}(n) = \sqrt{\frac{N}{2}} \cdot \frac{X(n)}{e(n)},$$

식 (2)는 다음과 같이 재 정의될 수 있다.

$$\hat{X}(n) = \sum_{k=0}^{N-1} x(k) \cos \frac{\pi(2k+1)n}{2N}, \quad (3)$$

$$n = 0, 1, \dots, N-1.$$

본 알고리즘의 요점은 식 (3)을 짝수  $n$ 과 홀수  $n$ 에 대해 분리하는 것이다.  $n = 2l$ 에 대해 식 (3)은 다음과 같이 되며

$$\hat{X}(2l) = \sum_{k=0}^{N/2-1} [x(k) + x(N-1-k)] \cos \frac{\pi(2k+1)2l}{2N}, \quad (4)$$

$$l = 0, 1, \dots, N/2-1.$$

이는  $N/2$ -포인트의 DCT로부터 얻어질 수 있다.  $n = 2l+1$ 에 대해 식 (3)은 다음과 같이 표현된다.

$$\hat{X}(2l+1) = \sum_{k=0}^{N/2-1} [x(k) - x(N-1-k)] \cos \frac{\pi(2k+1)(2l+1)}{2N},$$

$$l = 0, 1, \dots, N/2-1. \quad (5)$$

식 (5)를  $N/2$ -포인트 DCT의 형태로 변형하기 위해 다음의 삼각 등식을 이용한다.

$$2 \cos \frac{\pi(2l+1)}{2N} \cdot \cos \frac{\pi(2k+1)(2l+1)}{2N} \quad (6)$$

$$= \cos \frac{\pi(2l+1)2(k+1)}{2N} + \cos \frac{\pi(2l+1)2k}{2N}.$$

$y(k)$ 를 다음과 같이 정의할 때

$$y(k) = x(k) - x(N-1-k),$$

$$y(-1) = 0,$$

식 (5)는 다음과 같이 된다.

$$\hat{X}(2l+1) = \sum_{k=0}^{N/2-1} y(k) \cdot \cos \frac{\pi(2k+1)(2l+1)}{2N}$$

$$= \left(2 \cos \frac{\pi(2l+1)}{2N}\right)^{-1} \sum_{k=0}^{N/2-1} y(k)$$

$$\left[ \cos \frac{\pi(2l+1)2(k+1)}{2N} + \cos \frac{\pi(2l+1)2k}{2N} \right] \quad (7)$$

$$= \left(2 \cos \frac{\pi(2l+1)}{2N}\right)^{-1} \left[ \sum_{k=0}^{N/2-1} y(k) \cos \frac{\pi(2l+1)2(k+1)}{2N} \right.$$

$$\left. + \sum_{k=0}^{N/2-1} y(k) \cos \frac{\pi(2l+1)2k}{2N} \right].$$

$$= \left(2 \cos \frac{\pi(2l+1)}{2N}\right)^{-1} \left[ \sum_{k=0}^{N/2-1} (y(k-1) + y(k)) \cdot \cos \frac{\pi(2l+1)2k}{2N} \right]. \quad (8)$$

식 (8)의 재귀적 특성을 얻기 위해 식 (8)의  $y(k)$ 를 짝수  $k$ 와 홀수  $k$ 에 대해 분해하면 다음과 같다.

$$\hat{X}(2l+1) = \left(2 \cos \frac{\pi(2l+1)}{2N}\right)^{-1} \left[ \sum_{m=0}^{N/4-1} (y(2m-1) \right.$$

$$+ y(2m)) \cos \frac{\pi(2l+1)2m}{N}$$

$$+ \sum_{m=0}^{N/4-1} (y(2m) + y(2m+1))$$

$$\left. \cos \frac{\pi(2l+1)(2m+1)}{N} \right]. \quad (9)$$

식 (9)의 첫 번째 요소는 식 (8)의 반-포인트 DCT의 형태이고 두 번째 요소는 식 (5)의 반포인트의 DCT 형태임을 알 수 있다. 즉 식 (9)의 각 요소는 재귀적으로 분해될 수 있으며 이는  $2^m$ -포인트의 일반적인 경우로도 확장이 가능하다.

제안한 알고리즘의 일반적인 특성을 살펴보기 위해 16-포인트 DCT의 흐름도와 사용된 상수를 그림 1과 표 1에 나타내었다. 그림 1로부터 흐름도의 절반 윗부분, 즉  $X(n)$ 의 짝수 지수 부분은 8-포인트의 DCT

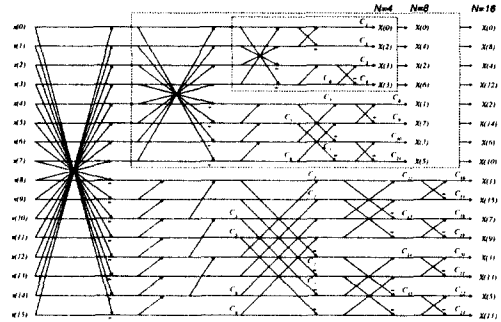


그림 1. 제안한 알고리즘에 의한 16-포인트 DCT의 흐름도.  
Fig. 1 Flowgraph of 16-point DCT by the proposed algorithm.

표 1. 16-포인트 DCT의 상수 ( $C_m^n = \cos \frac{\pi n}{m}$ ).

Table 1. Constants for 16-point DCT.

상수	일반식	상수	일반식
$C_0$	$1/C_4^4$	$C_{12}$	$1/2C_{16}^4$
$C_1$	$1/\sqrt{N}$	$C_{13}$	$1/2C_{16}^7$
$C_2$	$\sqrt{2/N} \cdot C_4^4$	$C_{14}$	$1/2C_{16}^3$
$C_3$	$\sqrt{2/N} \cdot C_4^4/2C_8^4$	$C_{15}$	$1/2C_{16}^6$
$C_4$	$\sqrt{2/N} \cdot C_4^4/2C_8^3$	$C_{16}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^2$
$C_5$	$C_4^4/C_8^4$	$C_{17}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^{15}$
$C_6$	$1/C_8^4$	$C_{18}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^7$
$C_7$	$C_8^4/C_8^4$	$C_{19}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^9$
$C_8$	$\sqrt{2/N} \cdot C_8^4/2C_{16}^4$	$C_{20}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^3$
$C_9$	$\sqrt{2/N} \cdot C_8^4/2C_{16}^7$	$C_{21}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^{13}$
$C_{10}$	$\sqrt{2/N} \cdot C_8^4/2C_{16}^3$	$C_{22}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^6$
$C_{11}$	$\sqrt{2/N} \cdot C_8^4/2C_{16}^6$	$C_{23}$	$\sqrt{2/N} \cdot C_4^4/2C_{32}^{11}$

를 포함하고 있으며 또한 8-포인트 DCT의 절반 윗부분 역시 4-포인트의 DCT를 포함하고 있음을 알 수 있다. 제안한 알고리즘은 [6]의 알고리즘에 비해 큰  $m$ 에 대해서도 흐름도상의 알고리즘의 유추가 가능하며 사용된 상수에 대한 일반식의 유추도 가능하다.

IDCT(Inverse DCT) 역시 비슷한 방법으로 분해되어 구현될 수 있지만, DCT가 직교 변환이므로 새로이 구하기보다는 DCT 행렬을 전치시켜 구현하는 방법이 일반적이다. 또한 이 과정은 흐름도 내의 화살표 방향을 역전시켜 새로운 흐름도를 얻는 과정과 같다.

### Ⅲ. 알고리즘의 특성

제안한 알고리즘은 DCT 커널의 대칭성을 이용하여 구현되며  $2^m$ -포인트의 일반적인 경우로도 확장이 가능하다.  $N$ -포인트 DCT  $X(n)$ 의 짝수 지수 부분은  $N/2$ -포인트 DCT로부터 얻어지며, 홀수 지수 부분은 식 (9)를  $\log_2 N - 2$ 번 분해함으로써 유도될 수 있다.

제안한 알고리즘의 덧셈 연산 수의 식은 다음과 같이 표현될 수 있으며

$$A(2) = 2, \quad A(4) = 9, \quad A(8) = 28, \quad A(16) = 78,$$

$$A(N) = N \cdot \log_2 N + \frac{5}{2} N - (\log_2 N)^2 - 3 \log_2 N + 2, \quad N \geq 32. \quad (10)$$

기존의 알고리즘[6][7][8]보다 적은 덧셈 연산 수를 필요로 한다. 이는 식 (10)의 우세항이  $N \cdot \log_2 N$ 인데 반하여 기존의 알고리즘은  $(3/2)N \cdot \log_2 N$ 으로 표현되기 때문이다. 또한 곱셈 연산 수를 나타내는 식은 다음과 같다.

$$M(N) = \frac{N}{2} \log_2 N. \quad (11)$$

위의 식 (11)은 기존의 대표적인 알고리즘에서 필요로 하는 곱셈 연산의 수를 나타내는 식과 같다. 8-포인트 DCT의 경우 제안한 알고리즘은 [6][7][8]보다 적은 수를 가지는 28번의 덧셈 연산과 [6][7][8]과 같은 12번의 곱셈 연산을 통해 구현된다. 표 2에 제안한 방법과 기존의 알고리즘과의 연산 수의 비교를 나타내었다.

표 2. 덧셈 및 곱셈 연산 수의 비교.

Table 2. Comparison of the number of additions and multiplications.

N	제안한 알고리즘		[6][7][8]	
	덧셈	곱셈	덧셈	곱셈
4	9	4	9	4
8	28	12	29	12
16	78	32	81	32
32	202	80	209	80
64	492	192	513	192

또한 제안한 알고리즘은 대부분의 곱셈 연산을 마지막 연산단에서 일괄적으로 수행하므로 고정 소수점 연산 시에 발생할 수 있는 오차의 전파를 줄일 수 있다(그림 1 참조). 이러한 고정 소수점 연산시 발생하는 오차는 하나의 DCT 변환 계수를 얻기 위해 통과해야 되는 곱셈 연산단의 수와도 밀접한 관계가 있다. 즉 DCT 변환된 계수를 얻기 위해 통과해야 하는 곱셈 단수가 적을수록 오차의 전파가 줄어들다[9]. DCT 변환된 계수를 얻기 위해 통과해야 하는 곱셈 단수의 최대값(Maximum Number of Multiplication Stages: MMS)을 나타내는 식은 다음과 같다.

$$MMS(N) = \begin{cases} \log_2 N - 1 & N \geq 8, \\ \log_2 N & N < 8. \end{cases} \quad (12)$$

8-포인트 DCT의 경우, 제안한 알고리즘에서는 모든 DCT 계수들이 2단계의 곱셈단을 통해 얻어지므로 MMS(8)은 2가 된다. 반면에 [6]에서는  $MMS(N) = \log_2 N$ 으로 표현되므로 제안한 알고리즘보다 1단계 많은 곱셈단을 통해 변환 계수들이 얻어지게 되며 이는 4장에서 언급될 정수 연산시 발생하는 오차에 영향을 미친다(그림 1 참조).

그림 1에 주어진 DCT의 흐름도로부터 입력 신호의 시퀀스는 0부터  $N-1$ 까지의 자연 순서로 되어 있음을 알 수 있다. 또한 출력 신호의 시퀀스는 다음과 같은 과정을 통해 정렬될 수 있다. 우선  $N=2$ 에 대해  $(X(0), X(1))$ 의 시퀀스를 얻은 후  $N=2^m$ 에 대해  $(X(a_0), X(a_1), \dots, X(a_{2^m-1}))$ 과 같은  $N/2$ -포인트 출력 신호의 시퀀스를 얻는다. 그리고 각각의  $j=1, 2, \dots, 2^{m-1}-1$ 에 대해  $X(a_j)$ 를  $(X(a_j), X(N-a_j))$ 로 분해하며  $X(a_0)$ 는  $(X(a_0), X(N/2))$ 로 분해한다.  $N=4$ 일 경우 출력 신호의 시퀀스는  $(X(0), X(2), X(1), X(3))$ 이 되고  $N=8$ 일 경우에는  $(X(0), X(4), X(2), X(6), X(1), X(7), X(3), X(5))$ 가 된다(그림 2 참조).

### Ⅳ. 오차 분석

신호 처리에 이용되는 DCT를 실제 구현할 경우 속도와 계산상의 편의를 위해 고정 소수점 연산을 사

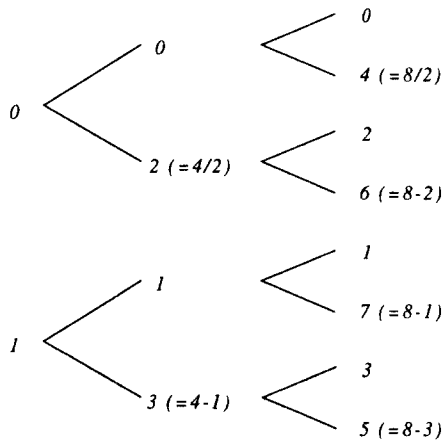


그림 2. 출력 신호 시퀀스의 순서.  
Fig. 2 Index of the output sequence.

용하게 된다. 영상이나 음성 신호의 압축을 위한 하드웨어에 이용되는 DCT는 연속적인 입력을 처리하기 위해 고속 연산이 필수적이므로 이들 하드웨어는 정수 연산을 기본으로 설계되어 있다. 이는 DCT칩을 개발하기 위한 ASIC 설계에서도 동일하게 적용된다. DCT 컴포넌트에서 입력 데이터는 고정 소수점 데이터이지만 커널의 값이 부동 소수점 데이터이므로 일정 비트를 할당하여 정수형의 데이터로 표현하여야 한다. 실수 데이터에 많은 비트를 할당하면 할수록 원 데이터의 값을 정확히 표현할 수 있으나, 시스템에서 사용할 수 있는 비트의 수가 한정되어 있으므로 반올림 오차가 발생하게 된다. 또한 이 반올림 오차는 곱셈 연산이 수행될 때마다 발생하여 다음 곱셈단으로 누적, 전파된다. 본 장에서는 제안한 알고리즘이 정수 연산시 발생하는 오차를 효과적으로 줄일 수 있음을 보이기 위해 실험을 통해 기존의 대표적인 알고리즘인 [6]과 성능 비교를 수행하였다.

정수 연산을 수행하는 DCT 구현에서는 다음과 같은 2가지의 오차가 발생하게 된다.

- 1) 실수 커널의 값을 정수로 표현할 때 발생하는 표현 오차
  - 2) 연속적인 곱셈에 의해 누적, 전파되는 누적 오차
- 그림 3에 제안한 알고리즘에서 사용되는 계수값과 [6]에서 사용되는 계수값들의 분포를 나타내었다. 제안한 알고리즘의 계수값들이 보다 균일한 분포를 보이

므로 한정된 비트 내에서 많은 비트를 할당하여 계수들을 표현할 수 있다. 또한 흐름도에서 볼 수 있듯이 제안한 알고리즘에 의한 DCT 시퀀스가 적은 곱셈단을 통과하여 얻어지므로 누적 오차를 줄일 수 있다.

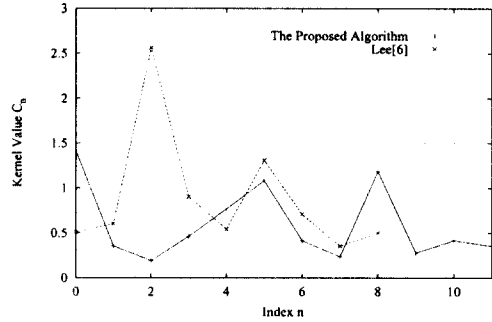


그림 3. 8-포인트 DCT에서 사용되는 커널값들의 분포.  
Fig. 3 Distribution of kernel values used in 8-point DCT.

시스템 내의 곱셈기의 입력 데이터 비트를  $B$ , 주어진  $B$ 에 대해 실수 커널  $C_n$ 을 최대의 정확도로 표현하기 위해 필요로 되는 최대 할당 비트를  $k$ 라 할 때,  $k$ 는 다음과 같이 계산될 수 있다.

$$k = \arg \max_k (\max_n (C_n) \cdot 2^k \leq 2^B). \quad (13)$$

즉 주어진 비트 내에서 실계수  $C_n$ 을 최대의 정확도로 표현하기 위해서는  $k \leq B$ 를 만족하는  $k$  중에서  $C_n \cdot 2^k$ 의 값이 최대인 것을 찾으면 된다. 위 식으로부터 얻어진  $k$ 를 이용하여 모든 실수 커널을 정수로 바꾸게 되며 이 정수값은 제한된  $B$ 비트 내에서 실계수값들을 최대의 정확도를 갖는 정수로 표현한 것이다. 최대 할당 비트  $k$ 를 크게 할수록 정수 표현 과정에서 나타나는 표현 오차를 줄일 수 있으며 식 (13)으로부터  $k$ 는  $C_n$ 의 최대값에 따라 결정됨을 알 수 있다. 따라서  $C_n$ 이 작으면서 균일한 분포를 가질 때 표현 오차를 줄일 수 있다. 제안한 알고리즘에서는  $2^m$ -포인트 DCT에 대해  $C_n$ 의 최대값이  $2^{m-3} \leq \max_n (C_n) < 2^{m-2}$ 의 범위 안에 존재하므로  $k$ 가  $B - (m - 2)$ 로

결정된다. 이는 [6]의 알고리즘의 최대 할당 비트  $B-(m-1)$ 보다 1이 더 크므로, 결과적으로 한 비트를 더 할당할 수 있게 되어 표현 오차를 그만큼 더 줄일 수 있다.

한정된 비트로 실수를 표현했을 때 나타나는 표현 오차는 곱셈단을 연속적으로 통과하면서 누적되므로 곱셈기 및 곱셈단의 수가 적은 구조에서는 누적 오차를 줄일 수 있다. 즉 곱셈기의 출력은 다음 곱셈단의 입력으로 들어가야 하므로 다음 단의 곱셈기 입력 비트에 맞추어 비트를 조절해야 한다. 따라서 출력을 잘라내는 과정에서 발생하는 오차는 곱셈단의 수가 줄어들면 감소될 수 있다. 실수 커널  $C_n$ 을  $k$ 비트로 표현했을 때의 정수값  $I_n(k)$ 는 다음과 같이 표현되며

$$I_n(k) = \lfloor C_n \cdot 2^k + 0.5 \rfloor, \quad (14)$$

여기서  $\lfloor x \rfloor$ 는  $x$ 를 넘지 않는 최대의 정수를 의미한다. 따라서  $C_n$ 의 정수형  $\tilde{C}_n$ 은 다음과 같이 표현될 수 있다.

$$\tilde{C}_n = I_n(k)/2^k. \quad (15)$$

고정 소수점 연산시 발생하는 총 오차의 계산은 식 (2)를 이용한 DCT 시퀀스와  $\tilde{C}_n$ 을 이용하여 계산된 DCT 시퀀스와의 차를 이용하여 수행될 수 있다.  $i$ 번째 랜덤 입력 벡터에 대해 식 (2)에 의한 표준 시퀀스를  $X_i$ ,  $\tilde{C}_n$ 을 이용하여 계산된 DCT 시퀀스를  $\tilde{X}_i$ 라 할 때, 두 시퀀스 사이의 평균 제곱 오차 (MSE)는 다음의 식으로 계산될 수 있으며

$$MSE = \frac{1}{BN} \sum_{i=0}^{BN-1} \frac{(X_i - \tilde{X}_i)^T \cdot (X_i - \tilde{X}_i)}{N}, \quad (16)$$

BN은 입력 랜덤 벡터의 개수이다. 제안한 알고리즘의 성능을 평가하기 위해, 8-포인트와 16-포인트 랜덤 10,000 벡터에 대해  $B$ 를 변경시켜 가며 MSE를 계산하였으며 이를 [6]의 알고리즘의 결과와 비교하여 그림 4와 그림 5에 나타내었다.

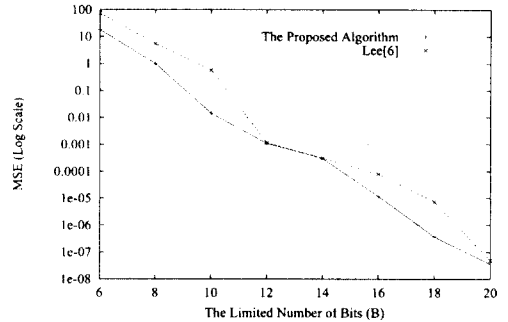


그림 4. 8-포인트 DCT에서 정수 연산시 발생하는 오차 비교  
Fig. 4 Comparison of error occurring in the integer computation for 8-point DCT.

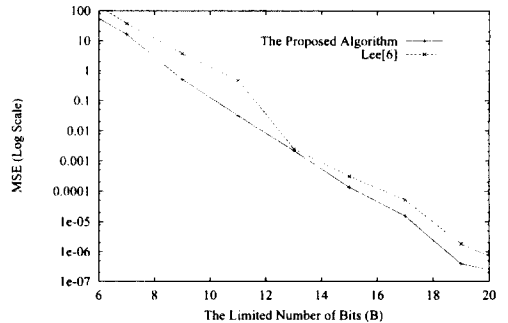


그림 5. 16-포인트 DCT에서 정수 연산시 발생하는 오차 비교  
Fig. 5 Comparison of error occurring in the integer computation for 16-point DCT.

그림에서 볼 수 있듯이 주어진 모든  $B$ 에 대해 제안한 알고리즘의 성능이 우수함을 알 수 있다. 이는 8-포인트 DCT의 경우, 주어진  $B$ 에 대해 제안한 알고리즘의  $k$ 는  $B-1$ 이며 [6]에서는  $B-2$ 가 되기 때문이며 또한 제안한 알고리즘이 MMS가 적은 구조로 구현되었기 때문이다. 즉 표현 오차와 누적 오차에서의 이득이 실험 결과에 반영되었음을 알 수 있다.

## V. 결 론

본 논문에서는 새로운 DCT 재귀 알고리즘의 구조와 특성을 제안하였다. 제안한 알고리즘은 적은 수의

덧셈 및 곱셈 연산을 통해 DCT를 구현하며 2<sup>m</sup>-포인트 DCT로의 일반적인 확장이 가능하다. 또한 대부분의 곱셈 연산을 마지막 단계에서 일괄적으로 수행하고 적은 곱셈 연산단을 통해 변환이 수행되므로 고정 소수점 연산 시에 발생할 수 있는 오차의 전파를 줄일 수 있다.

### 참 고 문 헌

1. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition*, Academic Press, 1992.
2. K. R. Rao and P. Yip, *Discrete Cosine Transform - Algorithms, Advantages, Applications*, Academic Press, 1990.
3. S. C. Chan and K. L. Ho, "A new two-dimensional fast cosine transform algorithm," *IEEE Trans. Signal Processing*, vol. 39, no. 2, pp. 481-485, Feb. 1991.
4. J. Makhoul, "A fast cosine transform in one and two dimensions," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 27-34, Feb. 1980.
5. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan. 1974.
6. B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1243-1245, Dec. 1984.
7. H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no.10, Oct. 1987.
8. P. Lee and F. Huang, "Restructured recursive DCT and DST algorithms," *IEEE Trans. Signal Processing*, vol. 42, no. 7, pp. 1600-1609, July 1994.
9. Y. Jeong, I. Lee, T. Yun, G. Park, and K. T. Park, "A fast algorithm suitable for DCT implementation with integer multiplication," *IEEE TENCON-Digital Signal Processing Applications*, pp. 784-787, 1996.



정 연 식(Yeonsik Jeong) 정회원  
1995년 2월:연세대학교 전자공학과 졸업(공학사)  
1997년 2월:연세대학교 대학원 전자공학과 졸업(공학석사)  
1997년 3월~현재:연세대학교 대학원 전자공학과 박사과정

※주관심분야:영상 처리, 필터 뱅크 시스템, 멀티미디어 통신 시스템



이 임 건(Imgeun Lee) 정회원  
1991년 2월:연세대학교 전자공학과 졸업(공학사)  
1993년 2월:연세대학교 대학원 전자공학과 졸업(공학석사)  
1998년 2월:연세대학교 대학원 전자공학과 졸업(공학박사)

1998년 2월~현재:대우전자 디지털 TV 연구소 선임 연구원

※주관심분야:신호 처리, 위성 방송, 필터 뱅크 시스템

최 영 호(Youngho Choi) 정회원  
한국통신학회 논문지 제 23권 제 8호 참조

박 규 태(Kyu Tae Park) 정회원  
현재:연세대학교 전자공학과 교수  
한국통신학회 논문지 제 22권 제 7호 참조