DBpia

論 文

DCT 행렬 분해에 관한 연구

正會員 崔 太 永*

On Factorizing the Discrete Cosine Transform Matrix

Tae Young CHOI* *Regular Member*

要 約 행렬 분해방식에 의한 새로운 고속 DCT 연산 방법을 유도하였다. N점 DCT변환 (T_N)을 N/2점 DCT 변환 ($T_{N/2}$)과 2개의 N/4점 변환(DCT의 전치행렬 $T_{N/4}^t$ 및 $\Phi_{N/4}$)들로 얻을수 있었다.

이 방법은 곱셈작용이 대부분 신호 흐름도상의 출력단에 가깝게 있게 되어 유한길이 연산인 경우에 발생하는 반올림오차량이 기존의 Lee와 Chen방법에 비하여 매우 적다는 점이 장점이다. 그리고 곱셈작용의 위치는 다르지만 동일 연산량을 갖는 또다른 3개의 DCT 행렬분해 결과도 보였다.

ABSTRACT A new fast algorithm for computing the discrete cosine transform(DCT) is developed decomposing an N-point DCT into an N/2-point DCT and two N/4-point transforms(transpose of an N/4-point DCT, $T_{N/4}^t$ and $\Phi_{N/4}$).

It has an important characteristic that in this method, the roundoff noise power for a fixed-point arithmetic can be reduced significantly with respect to the wellknown fast algorithms of Lee and Chen, since most coefficients for multiplication are distributed at the nodes close to the output and far from the input in the signal flow graph.

In addition, it also shows three other versions of factorization of DCT matrix with the same number of operations but with the different distributions of multiplication coefficients.

I. 서 론

1974년 DCT(discrete cosine transform)[1]가 나오면서부터 데이타 압축 효과가 다른 변환방법보다 우수하여 신호처리분야에 널리 응용되고 있다[2-11]. 그만큼 DCT의 고속 연산 방법도 다양하다[11-27].

기존의 고속 DCT 연산 방법들은 FFT(fast Fourier transform)나 다른 변환 방법을 이용하는 간접 방법[12-14]과 DCT 행렬을 분해하는 직접적인 방법[15-27]으로 대별할 수 있다.

행렬 분해 방법은 행렬의 원소가 거의 영으로 된 여러 형렬들로 분해하는 방법인 만큼 분해 방법도 여러가지일 것이고 분해 방법에 따라 연산량이나 성능이 다를 것이다. 근본적으로 (N×N) 크기의 DCT행렬의 N^2개의 원소값은 N가지의 서로 다른 음수 및 양수로 되어 있으므로 행렬 분해 방법에 의해 연산량을 줄이 수 있음을 쉽게 알 수 있다. 이 행렬분해 방법에 의한 고속 DCT 연산법의 태두라고 할수 있는 Chen[15] 방법은, 기본적으로 (N×N) 크기의 DCT 행렬이 (N/2×N/2) 크기의 DCT 행렬과 또 다른 직교 행렬($\Phi_{N/2}$: 2장에 정의)의 합성으로 되어 있으므로 이 Φ 행렬의 분해방법의 한 방법을 보인 것이다. 한편 cosine 계수의 역수를 처음 도입하여 IDCT (inverse DCT) 행렬을 2개의 절반크기의 IDCT 행렬로 분해한 Lee[18]의 방법은 승산수를 크게줄인 매우 효과적인 연산방법이다. 그러나 유한길이 연산에서는 N이 큰 경우에는 cosine계수의 역

*亞洲大學校 電子工學科
Dept. of Electronic Engineering, Ajou University
論文番號 : 91-118 (接受1991. 5. 28)

www.dbpia.co.kr

수의 값이 커져서 overflow될 염려가 있다[20].

본 논문에서는 행렬 분해 방식을 이용하여 안정되고 고속인 DCT 연산법을 제시하고자 한다. 근본적으로 Chen[15]의 DCT 행렬 분해에 기초하여 Φ 행렬과 DCT 또는 DCT의 전치행렬과의 관계식 4가지를 유도함으로서 고속연산법을 얻을 수 있었다.

본 논문은 Ⅱ장에서는 DCT의 행렬화 과정을 보이고 Ⅲ장에서는 Φ 행렬과 DCT및 DCT의 전치행렬과의 관계를 보이고 Ⅳ장에서는 제안된 고속 연산법의 신호 흐름도를 보이고 연산량 및 고정 소수점 승산시의 반올림 오차의 영향에 대하여 기존의 고속 연산법과 비교하고 마지막으로 결론 순으로 구성되어있다. 그리고 부록에는 본 논문에서 사용될 행렬 및 벡타의 정의식이 있다.

Ⅱ. DCT의 행렬화

1차원 이산신호 x(n) (n=0,1,2,…,N−1)의 N점 DCT 변환 y(m) (m=0,1,2,…,N−1)과 그 역변환(IDCT)은 각각 다음의 식 (1)과 (2)로 정의된다[1].

$$y(m) = \sum_{n=0}^{N-1} \sqrt{2}\ e(m) \cos\frac{\pi m(2n+1)}{2N}\ x(n) \qquad (1)$$

$$x(n) = \frac{1}{N}\sum_{m=0}^{N-1} \sqrt{2}\ e(m) \cos\frac{\pi m(2n+1)}{2N}\ y(m) \qquad (2)$$

여기서, $e(m) = \begin{cases} 1/\sqrt{2} & , m = 0 \\ 1 & , m \neq 0 \end{cases}$ 이다.

위 식 (1)의 DCT 변환식은 짝수와 홀수 변환계수로 구분하면 다음의 식 (3)과 (4)처럼 나타낼 수 있다.

$$y(2m) = \sum_{n=0}^{M-1} \sqrt{2}\ e(m) \cos\frac{\pi m(2n+1)}{N}\ s(n) \qquad (3)$$

$$y(2m+1) = \sum_{n=0}^{M-1} \sqrt{2} \cos\frac{\pi(2m+1)(2n+1)}{2N}\ d(n) \qquad (4)$$

여기서, M=N/2(정수), (m=0,1,2,…,M−1)
s(n)=x(n)+x(N−1−n)
d(n)=x(n)−x(N−1−n)이다.

이상의 식 (1),(3),(4)의 DCT 변환 관계를 부록에 정의한 행렬 기호(앞으로 모든 행렬 및 벡타는 부록을 참조바람)를 사용하여 행렬식화 하면 다음의 식 (5),(6)으로 됨을 알 수 있다(여기서 행렬 및 벡타의 크기가 표기 안된것은 모두다 N×N 및 N×1임).

$$Y = TX \qquad (5)$$

$$PY = QBX \qquad (6)$$

여기서

$$Q = \begin{bmatrix} T_{N/2} & Z_{N/2} \\ Z_{N/2} & \Phi_{N/2} \end{bmatrix}$$ 이다.

위 식(5),(6)에서 DCT 변환행렬 T와 전치 행렬 $T^t(=NT^{-1})$는 다음의 식 (7)-a 및 (7)-b로 행

www.dbpia.co.kr

렬 분해가 됨을 알수 있다.

$$T = P^t Q B \qquad (7)\text{-a}$$

$$T^t = B^t Q^t P \qquad (7)\text{-b}$$

Chen[15]이 이 (7)-a식에 의거하여 순환방식과 행렬분해(Q의 $\Phi_{N/2}$을 $(2\log_2 N-3)$개의 행렬로 분리)에 의한 고속 DCT 연산법을 보인 바와 같이 위 (7)식의 $\Phi_{N/2}$을 어떻게 연산하느냐가 고속 DCT의 관건임은 틀림없다.

본 논문에서는 이행렬Φ의 특성을 분석하여 DCT행렬 T와의 연관성을 구하여 고속화 방법을 제시하고자 한다.

Ⅲ. Φ행렬과 DCT 행렬과의 연관성

우선 Φ행렬의 원소 $\phi(m,n)$은 부록에서 정의한 바와 같이 다음과 같다(행렬크기 N×N).

$$\phi(m,n) = \sqrt{2}\cos\frac{\pi(2m+1)(2n+1)}{4N}$$

$$m,n = 0,1,2,\cdots,N-1 \qquad (8)$$

그런데 Φ가 다음 식(9)를 만족하는 바와 같이 직교행렬임은 잘 알려져 있다[16](참고문헌[16]에서는 EDCT-2, [17]에서는 DCT-Ⅳ라고 명명했음).

$$\left.\begin{aligned} \Phi\Phi &= NI \\ \Phi^{-1} &= (1/N)\Phi \end{aligned}\right\} \qquad (9)$$

그리고 Jain[16]이 직교행렬 Φ을 유도할때 사용한 매개변수형 행렬 $J(\alpha)$와 Φ의 관계식은 다음의 식(10),(11)과 같다.

$$J(\alpha) = \begin{pmatrix} (1+\alpha) & \alpha & 0 & 0 & & & & \\ \alpha & 1 & \alpha & 0 & & & O & \\ 0 & \alpha & 1 & \alpha & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & 0 & \\ & O & & & \ddots & \alpha & 1 & \alpha \\ & & & & & 0 & \alpha & (1-\alpha) \end{pmatrix} \qquad (10)$$

$$\Phi J(\alpha)\Phi = N\Lambda \qquad (11)$$

여기서 Λ는 대각선행렬로 그원소(λ_m ; m=0, 1,2,…,N-1)는 다음과 같다.

$$\lambda_m = 1 + 2\alpha\cos\frac{\pi(2m+1)}{2N} \qquad (12)$$

그런데 위 식(10),(12)는 α=1/2인 경우에(참고문헌[25]의 B=2J(1/2)과 같음).

$$J(1/2) = \frac{1}{2} U E^{-1} E^{-1} L$$

$$\lambda_m = 2\cos^2\frac{\pi(2m+1)}{4N}$$

이 되므로 식(11)에서 다음의 관계를 얻을 수 있음을 알수 있다.

$$\Phi U E^{-1} E^{-1} L \Phi = N C_d C_d \qquad (13)$$

그런데 식(9)를 이용하여 식(13)을 변형하면 식(14)처럼 되므로 또하나의 직교행렬(A)을 구할 수 있다.

www.dbpia.co.kr

$$NI = EU^{-1}\Phi C_d C_d \Phi L^{-1} E$$
$$= (EU^{-1}\Phi C_d)(EU^{-1}\Phi C_d)^t \quad (14)$$

$$A = EU^{-1}\Phi C_d \qquad AA^t = NI \quad (15)$$

그런데 바로 이 A행렬이 DCT행렬 T임을 다음의 식(16)의 결과로 알수 있다. 즉, 식 (15)의 우측항 ΦC_d값을 계산하면,

$$\Phi C_d = UE^{-1}T \quad (16)$$

이 되므로

$$A = T$$
$$= EU^{-1}\Phi C_d \quad (17)$$

임을 알수 있다.

위 식(17)의 결과와 Φ의 대칭성과 직교 특성인 식(9)의 결과를 이용하여 Φ와 T의 관계를 다음의 4가지 다른형태의 식으로 표시할 수 있다 ($T^tT=NI$의 성질이용 ; 부록참조).

$$\Phi = UE^{-1}TC_d^{-1} \quad (18)\text{-a}$$

$$= C_d^{-1}T^tE^{-1}L : \text{(18-a의 전치행렬)} \quad (18)\text{-b}$$

$$= C_dT^tEU^{-1} : \text{(18-a의 역행렬)} \quad (18)\text{-c}$$

$$= L^{-1}ETC_d : \text{(18-c의 전치행렬)} \quad (18)\text{-d}$$

여기에서 DCT 변환(T)및 전치행렬(T^t)을 이용하여 Φ변환을 얻을수 있음을 알수 있고 Φ변환시 주된 곱셈량은 T 또는 T^t와 C_d 또는 C_d^{-1}이고 U 또는 U^{-1}, L 또는 L^{-1}은 단순한 덧셈 작용임 (N−1 회)을 알수 있고 E 또는 E^{-1}은 단 1회의 곱셈이 필요함을 알수 있다. 한편, 식(18)-b의 결과는 Lee[18]의 고속 연산법과 결과적으로 동일함을 알수 있다.

또한 이 (18)식은 동일한 Φ행렬을 나타내지만 컴퓨터 연산시 고정 소수점 승산에 의한 오차량은[20] 세각각 다를 것이다. 이 (18)식을 토대로 DCT 변환의 계산량, Signal flow graph, 반올림 오차량등을 다음장에서 취급하겠다.

Ⅳ. DCT의 흐름도

앞서 나온 식(7) 및 (18)을 이용하여 N점 ($N=2^k$, $k\geq2$) DCT 변환을 순환 및 행렬분해로 얻고자 한다. 우선 N=2인 경우의 T및 Φ 변환부터 살펴보자.

1) 2점 DCT 변환 및 Φ변환

식 (1),(5) 및 부록에서 N=2일때의 DCT 행렬 T_2와 원소 $t_2(m,n)$은 식(19)와 같으므로 T_2변환은 단순한 합과 차 연산으로서 얻을수 있음을 알 수 있다.

$$t_2(m,n) = \sqrt{2}\, e(m) \cos \frac{\pi\, m(2n+1)}{4}$$
$$m,n = 0,1$$

$$T_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad T_2^t = T_2 \quad (19)$$

그런데 N=2일때의 Φ_2변환은 식(18)처럼 4가지 형태가 가능하다. 식(18)-a,b,c,d를 N=2일때 각각 행렬로 분해하여 나타낸것이 식(20)-a,b,c,d이고 각각의 흐름도(Y=ΦX)는 그림.1의 a,b,c,d이다.

www.dbpia.co.kr

$$\Phi_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} C(1/8)^{-1} & 0 \\ 0 & C(3/8)^{-1} \end{bmatrix} \quad (20)\text{-a}$$

$$= \begin{bmatrix} C(1/8)^{-1} & 0 \\ 0 & C(3/8)^{-1} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (20)\text{-b}$$

$$= \begin{bmatrix} C(1/8) & 0 \\ 0 & C(3/8) \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \quad (20)\text{-c}$$

$$= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} C(1/8) & 0 \\ 0 & C(3/8) \end{bmatrix} \quad (20)\text{-d}$$

여기서 $C(i/j)=2\cos\frac{i}{j}\pi$ 이다.

그림.1에서 식(18)의 4가지 형태의 Φ 변환(N=2)은 모두다 곱셈 3회 덧셈 3회가 필요함은 알수있고 그림.1의 a)와 d)는 b)와 c)에 비해 곱셈계산이 입력단에 가깝다는 점이 다르고 그림.1의 a)와 b)는 c)와 d)에 비하여 곱하는 상수값이 상호 역수인 점이 서로 다르다($C(i/j)$와 $C(i/j)^{-1}$ 및 $\sqrt{2}$와 $1/\sqrt{2}$).

한편 이미 알려진 바와 같이 i/j값이 1/2에 가까워질수록(N이 클수록) $C(i/j)^{-1}$을 곱하는 경우에는 곱셈 결과가 overflow될 염려가 있다[20]. 그렇지만 $C(i/j)$와 $C(i/j)^{-1}$ 모두 dynamic range(최대값/최소값의 비)가 동일하므로 곱셈결과가 overflow되지 않는 범위내에서 취급한다면 이점은 무시할수 있을 것이다. 하지만 곱셈결과의 반올림오차는 연산단이 길어질수록(예, 그림.1의 a), d)는 b), c)보다 반올림오차가 계속하여 덧셈기에 누적됨) 오차영향이 커질것이다. 만약 곱셈결과의 반올림 오차가 서로 독립된 잡음원으로 간주한다면(분산값 σ^2), 그림.1의 a)의 경우에 a와 a′노드에서 각각 오차분산값이 σ^2, σ^2이고 b, b′에는 각각 $2\sigma^2$, $2\sigma^2$, c, c′에는 $3\sigma^2$, $2\sigma^2$이 되므로 y(0), y(1)값은 각각 $5\sigma^2$, $2\sigma^2$의 오차분산값(합계 : $7\sigma^2$)을 갖는다는 것을 쉽게 알수 있다. 마찬가지로 그림.1의 b), c), d)의 경우에는 누세오차분산값은 각각 $4\sigma^2$, $4\sigma^2$, $8\sigma^2$임을 알수 있다.

앞의 간단한 예에서 알수 있듯이 반올림오차만을 고려한다면 식(18)-b 또는 (18)-c가 유리하며 overflow까지 고려한다면 식 (18)-c가 가장 효과적이기 때문에 앞으로 Φ변환에는 식 (18)-c를 사용하겠다. 그렇지만 식(18)-b와 (18)-c는 Φ계산시에 T가 아닌 T^t 전치행렬 변환식 (7)-b을 이용하여야 한다는 점에 유의하여야 하고 (18)식의 4가지 중 어느 것을 사용하여도 연산량은 동일함을 알수 있다.

2) T_4, T_4^t 및 Φ_4

식 (7)-a,-b, 식(18)-c 및 식(19)를 이용하여 4점 DCT 변환 T_4, 전치행렬 T_4^t 및 Φ_4 흐름도를 그려보면 그림.2의 결과를 얻을 수 있다.

한편 그림 2 내에서도 사용하였지만 함수 $MAV(T_4)=(\alpha,\beta,\gamma)$은 T_4행렬 연산시 필요한 승산수(α), 가감수(β), 및 고정소수점 승산시 발생하는 반올림 오차의 분산값(γ)를 뜻한다.

그리고 그림.2에서, T_4에서 Φ_2는 식(18)-c를 사용하였으나 T_4^t에서 Φ_2^t는 Φ_2^t인 식(18)-d를 사용치 않고 마찬가지로 식(18)-c를 사용함으로서 반올림 오차의 영향을 줄일수 있음을 알수 있다.

www.dbpia.co.kr

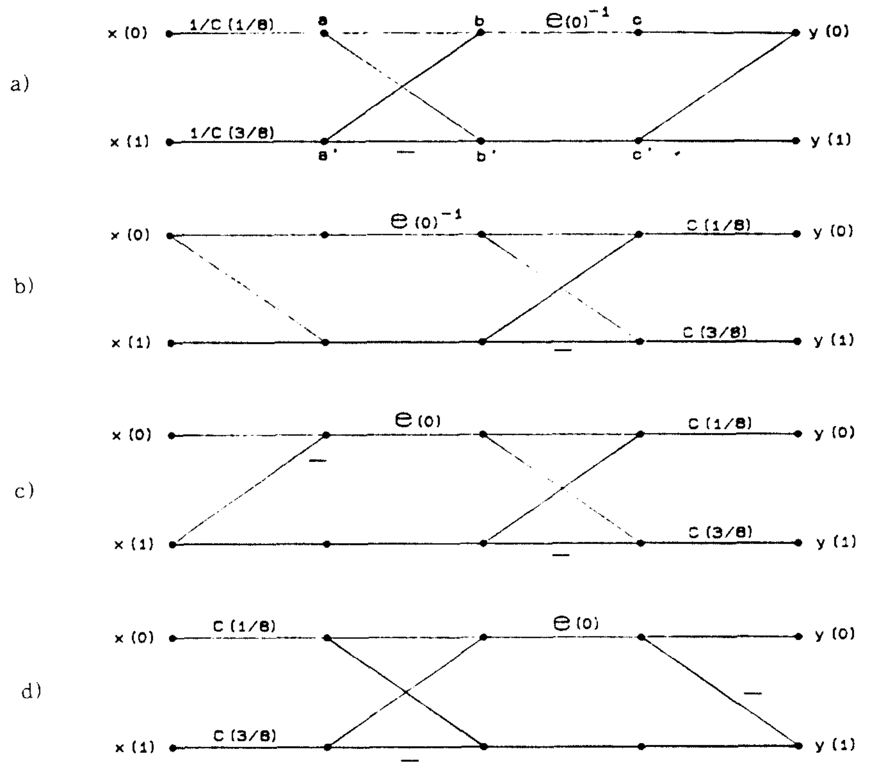


그림 1. 4가지 형태의 Φ_2변환 흐름도 : a,b,c,d는 각각 식(18) a,b,c,d의 흐름도임.

Fig. 1. Four types of signal flow graphs for 2-point Φ_2 transform : a,b,c, and d are signal flow graphs of Eq.(18) a,b,c and d, respectively.

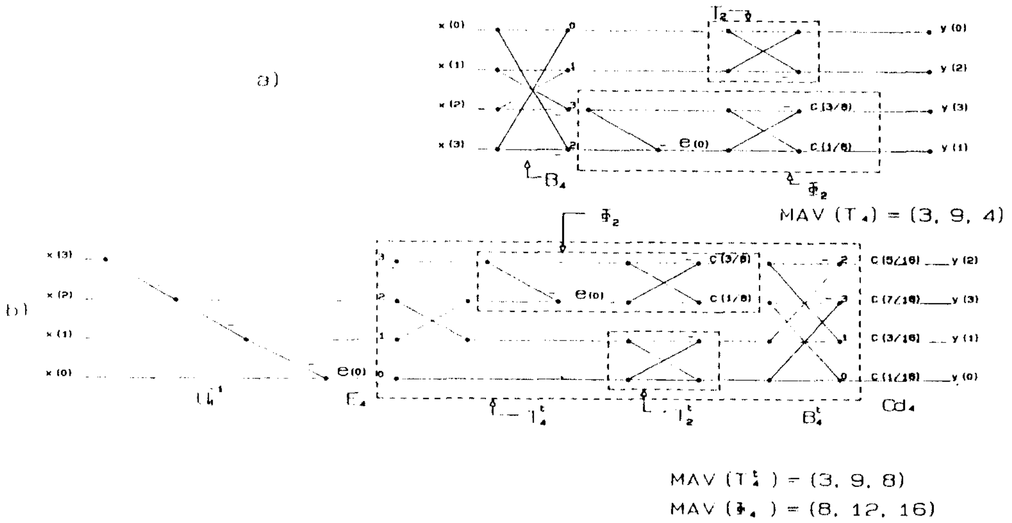


그림 2. 신호 흐름도 : a)4점 DCT(T_4), b)Φ_4.

Fig. 2. Signal flow graphs : a) T_4 b)Φ_4.

www.dbpia.co.kr

3) N점 DCT(N≥8)

식 (7)-a,b와 식 (18)-c에서 (N=8인 경우에),

$$T_8 = P_8^t \begin{bmatrix} T_4 & Z_4 \\ Z_4 & \Phi_4 \end{bmatrix} B_8 \qquad (21)\text{-a}$$

$$\Phi_4 = C_{d4} T_4^t E_4 U_4^{-1} \qquad (21)\text{-b}$$

$$T_4^t = B_4^t \begin{bmatrix} T_2 & Z_2 \\ Z_2 & \Phi_2 \end{bmatrix} P_4 \qquad (21)\text{-c}$$

$$\Phi_2 = C_{d2} T_2^t E_2 U_2^{-1} \qquad (21)\text{-d}$$

의 관계를 얻는다.

위 식(21)의 a,b,c,d에서 $T_2^t(=T_2)$를 기본으로 하여 Φ_2를 구할 수 있고 그 결과로 Φ_4^t를, 그리고 Φ_4를 구할수 있음을 알 수 있다. 따라서 T_8은 T_4와 2개의 T_2^t에서부터 얻을수 있음을 알 수 있다.

그림.3의 8점 DCT 변환 흐름도 그자체에 T_4^t 및 $\Phi_4(=C_{d4}\ T_4^t\ E_4\ U_4^{-1})$의 흐름도가 있으므로 N=16인 경우의 T_{16} 흐름도를 마찬가지 순환방식으로 2개의 T_4^t와 T_8의 흐름도에서 구할 수 있음을 알수 있다.

일반적으로 N≥8 이상($N=2^k$, k≥3)인 경우에 $T_{N/2}$과 2개의 $T_{N/4}^t$(또는 $T_{N/4}^t$와 $\Phi_{N/4}$)의 흐름도로부터 T_N의 흐름도를 순환방식으로 구할수 있음을 알수있다(그림.4).

4) 연산수 및 반올림 오차량

그림.2,3의 결과와 그림.4의 N점 DCT 흐름도를 참고하여 승산수, 가감수, 고정 소숫점 승산시의 반올림 오차 분산값을 구하여 보겠다.

우선 승산 오차신호는 상호 독립잡음원으로 간

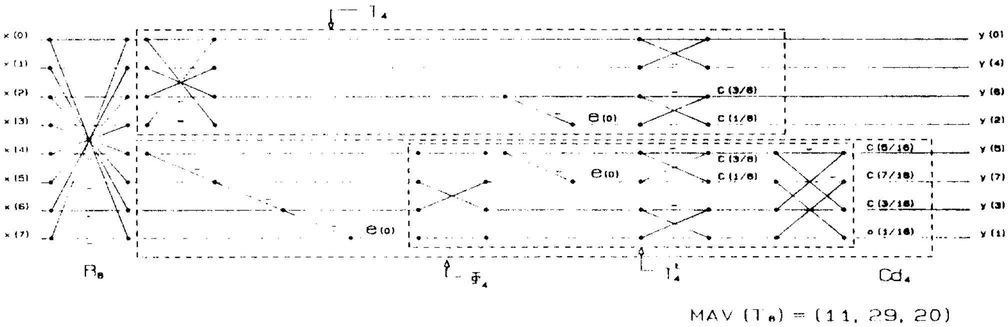


그림 3. 8점 DCT 흐름도.
Fgi. 3. Signal flow graphs of 8 point DCT.

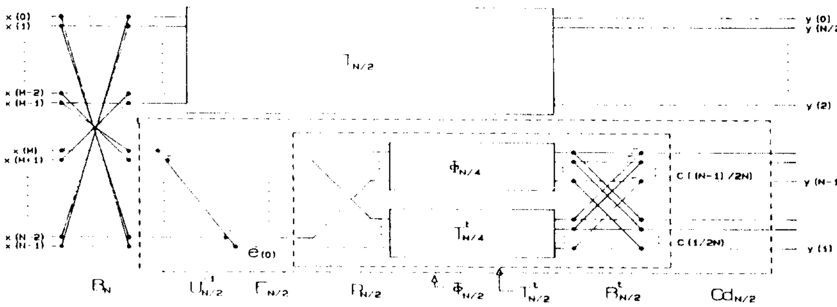


그림 4. N점 DCT 흐름도.(N=2^k, k≥3)
Fig. 4. Signal flow graphs of N point DCT.(N=2^k, k≥3).

www.dbpia.co.kr

주하고 하나의 오차 신호의 분산값이 1이라 가정하자($\sigma^2=1$). 그리고 $M(T_N)$, $A(T_N)$, $V_a(T_N)$은 각각 N점 DCT(T_N) 변환시 필요한 승산수, 가감수, 및 고정 소수점 승산에 의한 반올림 오차의 누계 분산값을 나타내는 함수라 하자.
식 (7)및 식(18)-c에서,

$$\left.\begin{aligned} M(T_N) &= M(T_N^t) \\ A(T_N) &= A(T_N^t) \\ M(T_2) &= 0 \\ A(T_2) &= 2 \\ V_a(T_2) &= 0 \end{aligned}\right\} \qquad (22)$$

임을 쉽게 알수 있다.
그림.2,3,4에서,

$$\left.\begin{aligned} M(T_N) &= M(T_{N/2}) + M(\Phi_{N/2}) \\ A(T_N) &= A(T_{N/2}) + A(\Phi_{N/2}) + N \\ Va(T_N) &= Va(T_{N/2}) + Va(\Phi_{N/2}) \end{aligned}\right\} \qquad (23)$$

그리고

$$\left.\begin{aligned} M(\Phi_{N/2}) &= M(T_{N/2}^t) + N/2 + 1 \\ A(\Phi_{N/2}) &= A(T_{N/2}^t) + N/2 - 1 \\ Va(\Phi_{N/2}) &= Va(T_{N/2}^t) + N \\ Va(T_N^t) &= 2\{Va(T_{N/2}^t) \\ &\quad + Va(\Phi_{N/2})\} \end{aligned}\right\} \qquad (24)$$

의 관계식을 구할수 있다
식(23) 및 (24)에서,

$$\left.\begin{aligned} M(T_N) &= 2M(T_{N/2}) + N/2 + 1 \\ A(T_N) &= 2A(T_{N/2}) \\ &\quad + (3/2)N - 1 \\ Va(T_N) &= Va(T_{N/2}) + 4Va(T_{N/4}^t) \\ &\quad + 2N \\ Va(T_N^t) &= 4Va(T_{N/2}^t) + 2N \end{aligned}\right\} \qquad (25)$$

의 수열형태의 식을 구할 수 있다.

식 (22)의 초기값을 이용하여 위 식(25)를 풀면,

$$\left.\begin{aligned} M(T_N) &= (N/2)\log_2 N - 1 \\ &= M(T_N^t) \\ A(T_N) &= (3/2)N\log_2 N \\ &\quad + 1 - N = A(T_N^t) \\ Va(T_N) &= (N^2-4)/3 \\ Va(T_N^t) &= N^2 - 2N \\ M(\Phi_N) &= (N/2)\log_2 N + N \\ A(\Phi_N) &= (3/2)N\log_2 N \\ Va(\Phi_N) &= N^2 \end{aligned}\right\} \qquad (26)$$

의 결과를 얻는다.

위 식 (26)의 결과는 N=4 및 8인 경우의 신호흐름도인 그림.2와 3에서 쉽게 확인할 수 있다.

한편 이 결과의 비교대상으로 고속 DCT연산법의 효시라 볼수 있는 Chen[15]과 역 Cosine계수를 사용하여 보다 빠른 결과를 얻는 Lee[18] 방법을 택하겠다. 이미 참고문헌[28]에서 이들에 대하여 고정 소수점 승산에 의한 반올림 오차를 비교한바있다. 이들에 대한 승산수, 가감수, 누계분산값 함수에 첨자 Chen, Lee를 사용하겠다.

www.dbpia.co.kr

$$
\begin{aligned}
M(T_N)_{Lee} &= (N/2)\log_2 N \\
&= M(T_N) + 1 \\
M(T_N)_{Chen} &= N\log_2 N \\
&\quad - (3/2)N + 4 \\
A(T_N)_{Lee} &= (3/2)N\log_2 N \\
&= A(T_N) \\
A(T_N)_{Chen} &= (3/2)N\log_2 N \\
&\quad - (3/2)N + 2 \\
Va(T_N)_{Lee} &= (9N^3/112) \\
&\quad + 18\log_2 N - (225/7) \\
Va(T_N^t)_{Lee} &= N^2 - N \\
&= Va(T_N^t) + N \\
Va(T_N)_{Chen} &= (2N/15)3^{\log_2 N} \\
&\quad + (16/5)
\end{aligned}
\qquad (27)
$$

위 참고문헌[28]에 의한 결과와 더불어 Chen과 Lee방식의 결과를 정리하면 다음과 같다.

5) 결과 검토

식 (26),(27)에서 본 논문에서 제안한 방법이 가감수 및 승산수는 Lee방법과 대등하고 승산시 반올림에 의한 누계오차 분산값 면에서는 월등함을 알수있고 IDCT인 경우도 Lee방법보다 나은 것은 식(1)처럼 DCT및 IDCT에 $\sqrt{2}$라는 값으로 정규화 하였기에 T_2에 또는 T_2^t에서 승산이 없고 단순 가감이 가능했기 때문에 e(0)의 영향이(전체적으로 $N\sigma^2$) 없어졌기 때문이다.

표.1은 본논문에서 제안한 연산법의 $M(T_N)$, $A(T_N)$, $V_a(T_N)$의 값을 이들과 비교한 것이다. 그리고 식 (26)에서 Φ 변환의 경우에 가감수 및 승산수는 참고문헌[21] 및 [illegible]의 결과와 동일함을 알수 있다.

마지막으로 그림.4의 N점 DCT변환의 흐름도에서의 출력노드의 변환계수 순서에 대하여 검토해 보자.

출력단의 k번째 노드(k=0,1,⋯N－1 : 출력단의 맨아래(k=0)로 부터 맨위(k=N－1)로 순서를 정했음)의 변환계수를 $R_N(k)$이라 하자. 그러면 N=2인 경우 $R_2(0)=1$, $R_2(1)=0$임을 알 수

표 1. N점 고속 DCT연산법의 성능 비교
Table. 1. Performance Comparisons of N point Fast DCT algorithms

	승산수		가감수		반올림 오차 누계분산 값			
N	Chen	본논문	Chen	본논문	Chen	Lee	Lee(T_N^t)	본논문
4	6	3	8	9	8	9	12	4
8	16	11	26	29	32	63	56	20
16	44	31	74	81	176	369	240	84
32	116	79	194	209	1040	2691	992	340
64	292	191	482	513	6224	21141	4032	1364
128	708	447	1154	1217	37328	168615	16256	5460

$$
\begin{aligned}
M(T_N)_{Lee} &= M(T_N) + 1 \\
A(T_N)_{Lee} &= A(T_N) \\
Va(T_N^t)_{Lee} &= Va(T_N^t) + N
\end{aligned}
$$

www.dbpia.co.kr

있다. 그림.2,3의 결과를 유추하여 N이 4이상인 경우에 k번째 노드의 변환계수 $R_N(k)$는 다음의 수식과 같음을 알수 있어서 N이 2배 증가할때마다 순환방식으로 출력노드의 변환계수값을 구할 수 있다.

$$R_N(k) = \begin{cases} 2R_{N/2}(k-N/2) & , N/2 \le k \le N-1 \\ N - R_{N/2}(k-N/4), & N/4 \le k \le (N/2)-1 \\ R_{N/2}(k) & , 0 \le k \le (N/4)-1 \end{cases}$$

V. 결　론

행렬분해 방식에 의한 고속 DCT 변환 방법을 보였다.

이 행렬분해과정에서 나타난 직교 대칭행렬 Φ의 분해가 고속연산의 중요 관건인데 본 논문에서는 이 Φ를 DCT행렬 T와 전치행렬 T^t를 포함하는 서로다른 4가지 형태로 분해할 수 있었고 그중 곱셈작용이 신호흐름도에서 보다 출력쪽에 가까이 일어나는 형태를 사용함으로서 기존방법에 비하여, 고정 소수점 승산오차량 면에서 매우 효과적이고 연산량 면에서도 대등한 고속 DCT 연산법을 얻을 수 있었다.

참 고 문 헌

1. N.Ahmed, T.Natarajan, and K.R.Rao, "Discrete cosine transform," IEEE trans. Comput., vol.C-23, pp.90-93, Jan.1974.
2. M.Hamidi. and J.pearl, "Comparison of the cosine and Fourier transforms of Markov-1 signals," IEEE Trans.Acoust., Speech, and Signal process, vol. ASSP-24, pp.428-429, Oct.1976.
3. H.Kitajima, T.Saito and T.Kurobe, "comparison of the discrcete cosine and Fourier transforms as possible substitutes for the Karhumer-Loéve transform," Trans. of IECE of Japan, vol.E60, No.6, pp.279-283, June 1977.
4. R.Zelinski and P.Noll, "Adaptive transform coding of speech signals," IEEE trans.Acoust., speech, and process., vol.ASSP-25, pp.299-309, Aug.1977.
5. J.A.Rose and G.S.Robinson, "Interframe cosine transform image coding," IEEE trans.Commun., vol. COM-25, pp.1285-1292, Nov.1977.
6. A.Ploysongsang and K.R.Rao, "DCT/DPCM processing of the NTSC composite video signal," IEEE Trans. Commun., vol.COM-30, pp.541-549, Malch 1982.
7. 박섭형, 이상욱, "계층적 피라미드 구조와 DCT 영역에서의 분류벡터 양자화기를 이용한 점진적 영상전송," 전자공학회 논문지 제26권, 제8호, pp.1277-1236, 1989년 8월.
8. W.K.Pratt, Digital Image signal processing, Chapter 10 and 23, New York, Wiley Interscience, 1973.
9. N.S.Jayant and P.Noll, Digital coding of waveforms, Chapter 12.Englewood Cliffs, NJ, Prentice-Hall, 1984.
10. J.S.Lim, Two-dimensional signal and image processing, Chapter 10, Prentice-Hall International, 1990.
11. K.R.Rao and P.Yip, Discrete Cosine Transform Algorithms, Advantages, Apllications, New York, Academic press, 1990.
12. B.D.Tseng and W.C.miller, "On computing the discrete cosine transform," IEEE Trans.Comput., vol. C-27, No.10, pp.966-968, Oct.1978.
13. M.Vetterli and H.J.Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," Signal Processing, vol.6, pp.267-278, Aug.1984.
14. S.Venkataraman, V.R.Kanchan, K.R.Rao and M. Mohanty, "Discrete transform via the Walsh-Hadamard transform," Signal Process., 14, pp.371-382, 1988.
15. W.H.Chen, C.H.Smith, and S.C.Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Commun., vol.COM-25, No.9, pp.1004-1009, Sept.1977.
16. A.K.Jain, "A sinusoidal family of unitary transforms," IEEE trans.Pattern Anal. and Machine Intell., vol.PAMI-1, pp.356-365, Oct.1979.
17. Z.Wang, "Fast algorithms for the dicrete W transform and for discrete Fourier trasform," IEEE

www.dbpia.co.kr

Tran., Acoust., Speech, and Signal Process., vol. ASSP-32, No.4, pp.803-816, Aug.1984
18. B.G.Lee, "A new algorithm for the discrete cosine transform," IEEE Trans. Acoust., Speech and Signal Process., vol. ASSP-32, No.6, pp.1243-1245, Dec.1984.
19. Z.Wang, "On computing the discrete Fourier and cosine transforms," IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-33, No.4, pp. 1341-1344, Oct.1985.
20. M.A.Haque, "A two-dimensional fast cosine transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-33, No.6, pp.1532-1538, Dec. 1985.
21. N.Suehiro and M.Hatori, "Fast algorithms for the DFT and other sinusoidal transforms," IEEE Trans.Acoust., Speech, and Signal Process., vol. ASSP-34, No.3, pp.642-644, June 1986.
22. H.S.Malvar, "Fast computation of the discrete cosine transform and discrete Hartley transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-35, No.10, pp.1484-1485, Oct.1987.
23. V.Nagesha, Comments on "fast computation of the discrete cosine transform and discrete Hartley transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-37, No.3, pp.439-440, March 1989.
24. H.S.Malvar, Corrections to, "Fast computation of the discrete cosine transform and the dicrete Hartley transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-36, No.4, pp.610, april 1988.
25. H.S.Hou, "A fast recursive algorithm for computing the cosine transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-35, No.10, pp.1455-1461, Oct.1987.
26. B.G.Lee "Input and output index mappings for a prime-factor-decomposed computation of discrete cosine transform," IEEE Trans.Acoust., Speech, and Signal Process., vol.ASSP-37, No.2, pp. 237-244, Feb.1989.
27. 박종연, "DCT, DST와 DHT의 고속 알고리즘 개발," 한국 통신학회 논문지, 제12권 제3호, pp.239-253, 1987년 6월.
28. 전준현, 고종석, 김성대, 김재균, "고속 DCT 변환방식의 정수형 연산에 관한 연구," 한국 통신학회 논문지 제12권 제4호, pp.309-324, 1987.8월.

www.dbpia.co.kr

부 록

행렬의 정의

다음에 정의한 행렬과 벡타의 크기는 각각 N×N, N×1이고 크기가 (N/2)×(N/2), (N/2)×1인 행렬과 벡타인 경우에는 행렬명칭에 N/2첨자를 부쳤음. 그리고 행렬과 벡타의 원소표기는 명칭의 소문자를 사용하여 나타냈으며 m과 n은 행렬의 행(row)과 열(column) 번호이다.(m, n=0,1,2,⋯,N−1)

X : x(m) : 데이타 벡타

Y : y(m) : DCT 변환된 데이타 벡타

I : $i(m,n)=\delta(m-n)$: 단위행렬

$$T : t(m,n) = \sqrt{2}\, e(m) \cos \frac{\pi m(2n+1)}{2N} \qquad \text{:DCT행렬}$$

$$T^tT = NI$$

$$\Phi : \phi(m,n) = \sqrt{2} \cos \frac{\pi(2m+1)(2n+1)}{4N}$$

$$\Phi\Phi = NI$$

$$Z : z(m,n) = 0 \quad \text{: 영행렬}$$

$$P : p(m,n) = \begin{cases} \delta(2m-n) & , 0 \le m \le (N/2)-1 \\ \delta(2(m-N/2)-n+1), & (N/2) \le m \le N-1 \end{cases} \quad \text{: 짝수 및 홀수 계수 분리}$$

Permutation 행렬, $P^tP=I$

$$\hat{I} : \hat{i}(m,n) = \delta(m+n-N+1)$$

$$B = \begin{bmatrix} I_{N/2} & \hat{I}_{N/2} \\ I_{N/2} & -\hat{I}_{N/2} \end{bmatrix} = \begin{bmatrix} 1 & & & & & 1 \\ & \cdot & & & \cdot & \\ & & 1 & 1 & & \\ 1 & & & & & -1 \\ & \cdot & & & \cdot & \\ & & 1 & -1 & & \end{bmatrix}$$

$$E : e(m,n) = e(m)\delta(m-n) \quad \text{: 대각선 행렬} \; (e(0,0)=1/\sqrt{2},\; e(m,m)=1,\; m\neq 0)$$

$$U : u(m,n) = \delta(m-n) + \delta(m-n+1) \quad \text{: 대각선 및 상측 제1대각선 원소 모두 1}$$

$$L : l(m,n) = \delta(m-n) + \delta(m-n-1) \quad \text{: 대각선 및 하측 제1대각선 원소 모두 1}$$

$$U^t = L$$

$$C_d : c_d(m,n) = 2\cos\frac{\pi(2m+1)}{4N}\,\delta(m-n) \quad \text{: 대각선 행렬}$$

$$C_d^{-1} = C_{di}, \quad c_{di}(m,n) = 1 \Big/ \left[2\cos\frac{\pi(2m+1)}{4N}\right]\delta(m-n) \quad \text{: 대각선 행렬}$$

www.dbpia.co.kr

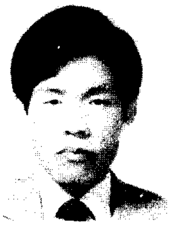
$$F_k : f_k(m,n) = \delta(m-n) - \delta(m-k, n-k-1)$$

$$U^{-1} = F_0 F_1 \cdots F_{N-2}$$

$$= U_i \qquad : u_i(m,n) = (-1)^k \delta(m-n+k), \; (0 \le k \le (N/2)-1)$$

$$U^{-1} = \begin{bmatrix} 1 & -1 & 1 & -1 & \cdot & \cdot & \cdot & -1 \\ & \cdot & \cdot & \cdot & \cdot & & & \cdot \\ & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & -1 \\ & & & & & \cdot & \cdot & 1 \\ & 0 & & & & & \cdot & -1 \\ & & & & & & & 1 \end{bmatrix}$$

$$L^{-1} = (U^{-1})^t = L_i : l_i(m,n) = (-1)^k \delta(n-m+k), \; (0 \le k \le (N/2)-1)$$



崔 太 永(Tae Young CHOI) 正會員
1950年 10月 24日生
1970年 2月～1974年 2月 : 서울大學校 工科大學電子工學科卒業(工學士)
1976年 9月～1978年 8月 : 서울大學校 大學院 電子工學科(工學碩士)
1979年10月～1982年12月 : 프랑스 Aix-Marseille 3大學(工學博士)
1983年 3月～現在 : 亞洲大學校 工科大學 電子工學科 副教授
관심분야 : 신호처리, 통신시스템

www.dbpia.co.kr