

단순 엘레먼트 인버스를 이용한 DCT/DFT 하이브리드 알고리즘

정회원 이 광 재*, 박 대 철**, 이 문 호***, 준회원 신 태 철***, 진 주***

DCT/DFT Hybrid Algorithm using Simple Element Inverse

Kwang-jae Lee*, Dae-chul Park**, Moon-ho Lee*** *Regular Members,*
Tae-chol Sin***, Zhu Chen*** *Associate Members*

요 약

본 논문에서는 하이브리드 구조에 의한 DCT/DFT 행렬의 새로운 표현을 제시한다. 엘레먼트 인버스 소행렬 분해 알고리즘에 근거하여 DCT와 DFT가 동일한 반복 연산 패턴을 갖고 있음을 보이며, 몇몇 대각행렬을 이용한 하이브리드 구조를 개발한다.

Key Words : DCT, DFT, Element inverse, Matrix factorization.

ABSTRACT

In this paper, we present new representation of DCT/DFT matrices via one hybrid architecture. Based on a element inverse matrix factorization algorithm, we show that the DCT and DFT have a same recursive computational pattern, and we can develop an hybrid architecture by using some diagonal matrices.

I. 서 론

DCT(discrete cosine transform)는 신호의 분류 및 표현을 위한 다양한 응용분야에서 이용되고 있다^[1,2,3]. DCT-II는 통계적으로 최적 Karhunen-Loeve 변환에 근접하는 성능을 가지는 우수한 부 최적 변환으로 잘 알려져 있다^[3,4,5]. 또한 DFT 역시 신호처리와 통신을 위해 자주 이용되는 변환이다^[6,7,8]. 이 두 가지 서로 다른 변환을 해석하기 위해 이들 전달 행렬에 대한 희소 행렬 분해를 고찰한다. 희소 행렬 분해 및 합성은 특성 생성 및 고속 연산의 개발을 도구로서 활용되어 왔다^[9,10,11].

한편, 엘레먼트 인버스 희소 행렬 분해의 기본 개념은 문헌^[12]의 저자에 의해 가장 포괄적인 직교

변환의 형태로 일반화시킨 재킷 변환(Jacket transform)에서 소개하고 있으며, 재킷 행렬의 주요 특성은 이 행렬의 원소에 대한 역행렬이 매우 용이하게 얻어질 뿐만 아니라 특별한 구조를 갖는다는 점이다. 재킷변환 외에도 다양한 직교변환의 소행렬 분해과정에서 엘레먼트 인버스 특성을 확인할 수 있다. DCT-II 및 DFT 행렬 또한 문헌^[9-13]와 유사하게 하나의 특성 행렬과 특별한 희소 행렬로 분해할 수 있으며, 이러한 형태에서 희소 행렬의 역은 블록 단위 역이거나 또는 엘레먼트 단위의 역의 형태를 갖는다.

본 논문에서는 희소 행렬 분해의 구조 그리고 DCT와 DFT를 함께 결합한 하이브리드 구조를 제시하며, 제안 방법을 엘레먼트 인버스 희소 행렬 분

※ 본 연구는 MOC ITRC, KOSEF, ETRI, Post BK21 지원에 의한 결과임.

* 한려대학교 멀티미디어정보통신공학과 (kjlee@hanlyo.ac.kr), ** 한남대학교 정보통신공학과 (daechul@hannam.ac.kr)

*** 전북대학교 전자정보공학부 (moonho@chonbuk.ac.kr), (tcshin@naver.com), (chenzhu@chonbuk.ac.kr)

논문번호 : KICS2006-10-415, 접수일자 : 2006년 10월 10일, 최종논문접수일자 : 2007년 5월 21일

해로 명명한다^[10,11].

II. DCT-II 행렬을 위한 엘레먼트 인버스 최소 행렬 분해

재킷 행렬^[12,14]의 정의와 유사하게 $N \times N$ 최소 행렬의 역은 단지 블록 단위 역이거나 또는 엘레먼트 단위의 역의 형태를 가지므로 이를 엘레먼트 인버스 최소 행렬로 부르기로 한다. 전형적인 DCT 행렬은 DCT-II의 경우이며, 이는 다음과 같이 정의된다.

$$[C_N]_{m,n} = \sqrt{\frac{2}{N}} k_m \cos \frac{m(n + \frac{1}{2})\pi}{N}, m, n = 0, 1, \dots, N-1. \quad (1)$$

$$\text{여기서 } k_j = \begin{cases} 1, & j=1, 2, \dots, N-1 \\ \sqrt{\frac{1}{2}}, & j=0, N \end{cases}$$

본 논문에서는 DCT-II 행렬 위주로 간단한 행렬 분해 알고리즘을 고찰한다. 먼저, 2×2 DCT-II 행렬은 다음과 같이 주어지며

$$[C_2] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}}, \quad (2)$$

여기서 $1/\sqrt{2}$ 은 이의 역이 $\sqrt{2}$ 이면 차수 1의 특별한 엘레먼트 인버스 최소 행렬로 간주할 수 있으며, $C_4^i = \cos(i\pi/4)$ 은 DCT 계산을 위한 cosine unit이다. 다음으로, 4×4 DCT-II 행렬은 다음과 같이 주어진다.

$$[C_4] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^4 & C_8^0 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} \quad (3)$$

행 연산 행렬 $[Pr]_N$ 은 다음과 같이 정의된다.

$$[pr]_2 = [I]_2 \text{ and } [pr]_N = [pr_{i,j}]_N, N \geq 4 \quad (4)$$

여기서

$$\begin{cases} pr_{i,j} = 1, & \text{if } i=2j, 0 \leq j \leq N/2-1 \\ pr_{i,j} = 1, & \text{if } i=(2j+1) \bmod N, 0 \leq j \leq N/2-1, \\ pr_{i,j} = 0, & \text{others} \end{cases}$$

그리고 $i, j \in \{0, 1, \dots, N-1\}$. 또한 가역(reversible) 치환행렬 $[Pc]_N$ 은 다음과 같이 정의된다.

$$[Pc]_2 = [I]_2, [Pc]_4 = \begin{bmatrix} \frac{I_N}{4} & 0 & 0 & 0 \\ 0 & \frac{I_N}{4} & 0 & 0 \\ 0 & 0 & 0 & C_8^3 \\ 0 & 0 & I_N & 0 \end{bmatrix}, N \geq 4 \quad (5)$$

따라서 이는 다음과 같이 쓸 수 있다.

$$[Pr]_4 [C_4 [Pc]_4 = \left(\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix} \right)^T \quad (6)$$

$$\text{여기서 } C_2 \text{는 식 (2)의 } [C_2] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, (\cdot)^T \text{는}$$

행렬의 전치, 그리고 $[B]_2 = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix}$ 를 나타낸다.

명백하게 다음과 같은 블록단위 인버스 최소행렬을 얻을 수 있다.

$$\begin{bmatrix} C_2 & 0 \\ 0 & B_2 \end{bmatrix} = \begin{bmatrix} (C_2)^{-1} & 0 \\ 0 & (B_2)^{-1} \end{bmatrix} \quad (7)$$

일반적으로 치환된 DCT-II 행렬 $[\tilde{C}]_N$ 은 다음의 관계를 이용하여 반복적(recursively)으로 구성할 수 있다.

$$[\tilde{C}]_N [Pr]_N [C_N [Pc]_N = \left(\begin{bmatrix} \frac{I_N}{2} & \frac{I_N}{2} \\ \frac{I_N}{2} & -\frac{I_N}{2} \end{bmatrix} \begin{bmatrix} C_N & 0 \\ 0 & B_N \end{bmatrix} \right)^T. \quad (8)$$

여기서, $[B]_N$ 은 다음과 같이 계산된다.

$$\begin{aligned} [B]_N &= [(C_2^{f(m,n)})_{m,n}]_{\frac{N}{2}}, \\ \{f(m,1)\} &= 2m-1, \\ \{f(m,n+1)\} &= f(m,n) + 2f(m,1). \end{aligned} \quad (9)$$

여기서 $m, n \in \{1, 2, \dots, N/2\}$. 식 (8)의 역은 다음과 같이 간단히 계산할 수 있다.

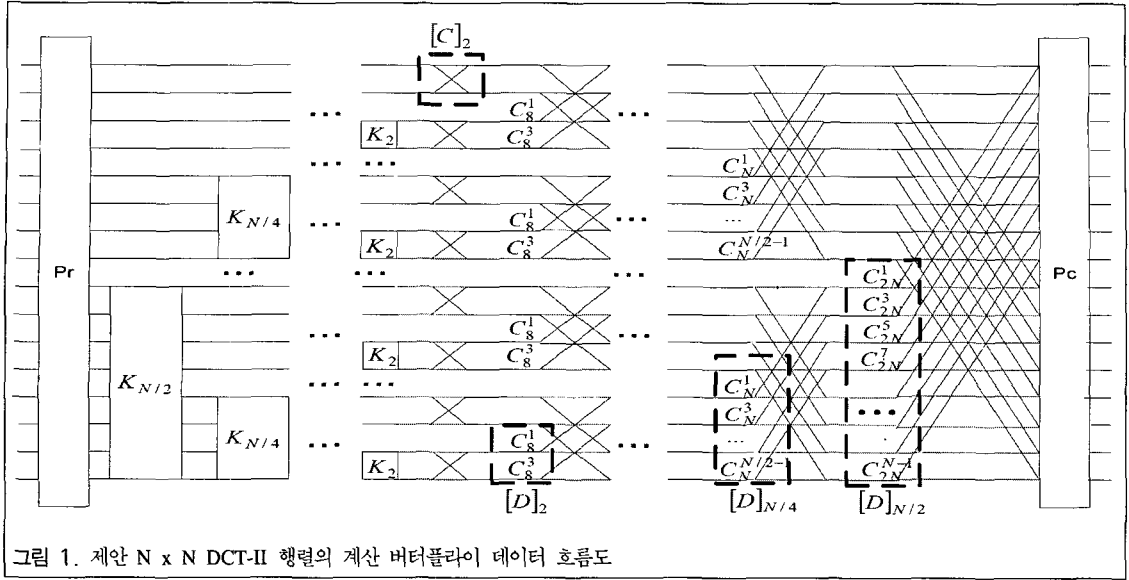


그림 1. 제안 N x N DCT-II 행렬의 계산 버터플라이 데이터 흐름도

$$(\tilde{C}_v)^{-1} = \frac{N}{2} \left(\begin{bmatrix} (C_N)^{-1} & 0 \\ 0 & -(B_N)^{-1} \end{bmatrix} \begin{bmatrix} I_N & I_N \\ I_N & -I_N \end{bmatrix} \right)^T \quad (10)$$

또한, 부행렬 $[B]_N$ 은 다음과 같이 나타낼 수 있다.

$$[B]_N = [K]_N [C]_N [D]_N \quad (11)$$

여기서,

$$[K]_N = \begin{bmatrix} \sqrt{2} & 0 & \dots & 0 \\ -\sqrt{2} & 2 & 0 & \dots \\ \sqrt{2} & -2 & 2 & \dots \\ \dots & \dots & \dots & 2 \end{bmatrix}, [D]_N = \text{diag} \{ C_{4N}^{\phi_0}, \dots, C_{4N}^{\phi_{N-1}} \},$$

그리고

$$\phi_i = 2i+1, i = \{0, 1, \dots, N-1\}.$$

식 (11)의 증명: $N \times N$ DCT-II 행렬은 다음과 같은 구성된다.

$$[C]_N = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\phi_0} & C_{4N}^{2k_0\phi_1} & \dots & C_{4N}^{2k_0\phi_{N-1}} \\ \dots & \dots & \dots & \dots \\ C_{4N}^{2k_{N-1}\phi_0} & C_{4N}^{2k_{N-1}\phi_1} & \dots & C_{4N}^{2k_{N-1}\phi_{N-1}} \end{bmatrix} \quad (12)$$

여기서 $k_i = i+1, i \in \{0, 1, 2, \dots\}$. 식 (9)로부터 $[C]_{2N}$ 으로 부터의 행렬 $[B]_N$ 은 다음과 같이 나타낼 수 있다.

$$[B]_N = \begin{bmatrix} C_{4N}^{\phi_0} & C_{4N}^{\phi_1} & \dots & C_{4N}^{\phi_{N-1}} \\ C_{4N}^{(2k_0+1)\phi_0} & C_{4N}^{(2k_0+1)\phi_1} & \dots & C_{4N}^{(2k_0+1)\phi_{N-1}} \\ \dots & \dots & \dots & \dots \\ C_{4N}^{(2k_{N-1}+1)\phi_0} & C_{4N}^{(2k_{N-1}+1)\phi_1} & \dots & C_{4N}^{(2k_{N-1}+1)\phi_{N-1}} \end{bmatrix} \quad (13)$$

분명히 다음의 관계를 얻게 된다.

$$C_{4N}^{\phi_m} - 2C_{4N}^{2k_0\phi_m} C_{4N}^{\phi_m} = -C_{4N}^{(2k_0+1)\phi_m} \quad (14)$$

$$C_{4N}^{\phi_m} - 2C_{4N}^{2k_1\phi_m} C_{4N}^{\phi_m} + 2C_{4N}^{2k_2\phi_m} C_{4N}^{\phi_m} = -C_{4N}^{(2k_0+1)\phi_m} \quad (15)$$

식 (14) 및 (15)를 이용하면 식 (11)은 다음과 같이 나타낼 수 있다.

$$[K]_N [C]_N [D]_N = \begin{bmatrix} C_{4N}^{\phi_0} & C_{4N}^{\phi_1} & \dots & C_{4N}^{\phi_{N-1}} \\ C_{4N}^{(2k_0+1)\phi_0} & C_{4N}^{(2k_0+1)\phi_1} & \dots & C_{4N}^{(2k_0+1)\phi_{N-1}} \\ C_{4N}^{(2k_1+1)\phi_0} & C_{4N}^{(2k_1+1)\phi_1} & \dots & C_{4N}^{(2k_1+1)\phi_{N-1}} \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad (16)$$

이상과 같이 증명된다. 따라서 DCT-II 행렬은 다음과 같이 쓸 수 있다.

$$[\tilde{C}]_N = \begin{bmatrix} I_N & 0 \\ 0 & K_N \end{bmatrix} \begin{bmatrix} C_N & 0 \\ 0 & C_N \end{bmatrix} \begin{bmatrix} I_N & 0 \\ 0 & D_N \end{bmatrix} \begin{bmatrix} I_N & I_N \\ I_N & -I_N \end{bmatrix} \quad (17)$$

또한 반복식의 일반형은 다음과 같이 주어진다.

$$[C]_N = ([P]_N)^{-1} \begin{bmatrix} I_N & 0 \\ 0 & K_N \end{bmatrix} \left[\frac{I_N}{4} \otimes ([P]_N)^{-1} \right. \\ \left. \cdot \left[\frac{I_N}{4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \right] \left[\frac{I_N}{2} \otimes C_2 \right] \right. \\ \left. \cdot \left[\frac{I_N}{4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \right] \left[\frac{I_N}{4} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \right] \left[\frac{I_N}{4} \otimes ([P]_4)^{-1} \right] \right]$$

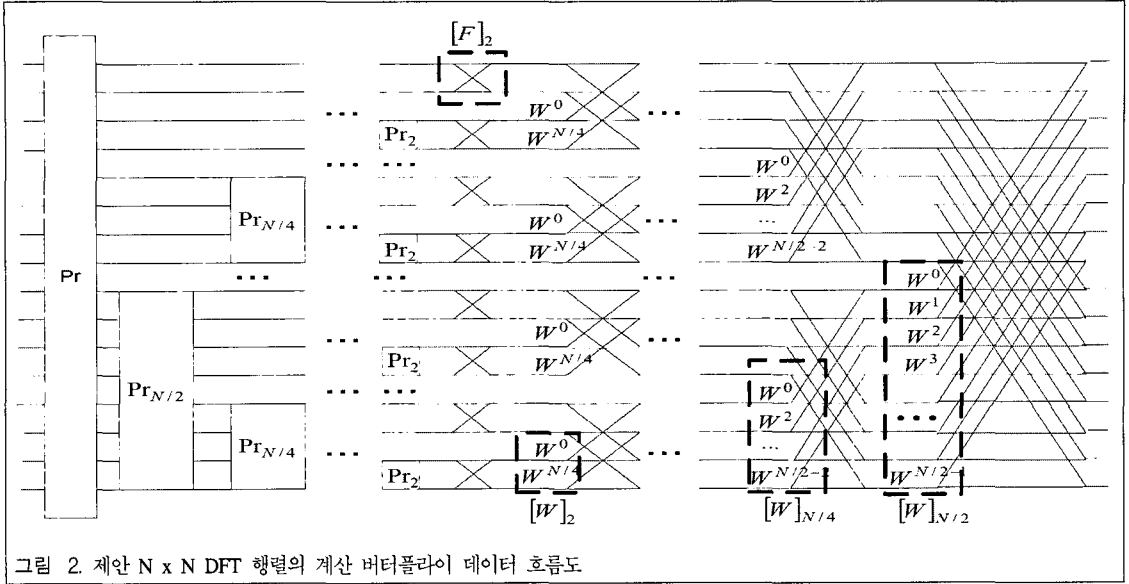


그림 2. 제안 N x N DFT 행렬의 계산 버터플라이 데이터 흐름도

$$\cdot \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} ([Pc]_N)^{-1} \quad (18)$$

식 (18)은 다음과 같이 간략한 형태로 다시 쓸 수 있다.

$$\begin{aligned} [C]_N &= ([\tilde{Pr}]_N)^{-1} \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \\ I_{N/2} \otimes C_2 \end{bmatrix} \\ &\cdot \begin{bmatrix} I_{N/4} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \\ I_{N/4} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \end{bmatrix} \\ &\cdot \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} ([\tilde{Pr}]_N)^{-1} \quad (19) \end{aligned}$$

그림 1은 식 (19)와 관련된 버터플라이 데이터 흐름도를 나타낸다.

III. DFT 행렬을 위한 엘레먼트 인버스 회소 행렬 분해

DFT는 주어진 수열 $x(m)$, $0 \leq m \leq N-1$ 의 푸리에 표현이며, 이는 다음과 같이 정의된다.

$$X(n) = \sum_{m=0}^{N-1} x(m) W^{nm}, \quad 0 < n < N-1 \quad (20)$$

여기서, $W = e^{-j\frac{2\pi}{N}}$, $j = \sqrt{-1}$. N점 DFT 행렬은

$[F]_N = [W^{nm}]_N$ 과 같이 정의할 수 있다. 제 2장에 서와 유사하게 치환된 4×4 DFT 행렬을 다음과 같이 쓸 수 있다.

$$[\tilde{F}]_4 = [Pr]_4 [F]_4 = \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \begin{bmatrix} F_2 & 0 \\ 0 & E_2 \end{bmatrix}^T \quad (21)$$

여기서, $[E]_2 = \begin{bmatrix} 1 & -j \\ 1 & -j \end{bmatrix}$, 그리고 이의 역은 다음과 같이 얻어진다.

$$[E]_2^{-1} = \left(\begin{bmatrix} 1/1 & -1/f \\ 1/1 & 1/f \end{bmatrix} \right)^{-1} = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix} \quad (22)$$

일반적으로 $N \times N$ 치환 DFT 행렬은 다음과 같이 나타낼 수 있다.

$$[\tilde{F}]_N = [Pr]_N [F]_N = \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \tilde{F}_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix}^T \quad (23)$$

여기서, $[\tilde{F}]_2 = [F]_2$, 그리고 부행렬 $[E]_N$ 은 다음과 같이 쓸 수 있다.

$$[E]_N = [Pr]_N [\tilde{F}]_N [W]_N \quad (24)$$

여기서, $[W]_N = \text{diag}[W^0, W^1, \dots, W^{N-1}]$, 그리고 W 는 $2N$ 점 DFT 행렬에 대한 복소 unit이다. 식 (17)과 유사하게 치환 DFT 행렬을 다음과 같이 쓸 수 있다.

IV. 결론

본 논문에서는 DCT-II 및 DFT 행렬에 대한 반복식을 도출하였으며, 결과적으로 DCT-II 및 DFT 행렬은 동일한 최소 행렬 분해 알고리즘과 몇 가지 특성 변경을 수반하는 반복 구조를 이용하여 단일 화할 수 있음을 나타내고 있다. 그림 1 및 그림 2에 나타낸 것처럼 DFT 계산은 부 행렬 $[D]_N$ 을 $[W]_N$ 으로, 치환행렬 $[Pr]_N$ 을 $[K]_N$ 으로 대체함으로써 DCT 행렬 계산으로부터 얻어질 수 있으며, 고속 알고리즘에 의한 계산 회수를 표 1에 나타내었다^[15]. 결과적으로 DCT/DFT 하이브리드 구조를 위한 일 반화된 간단한 블록 다이어그램과 이의 고속 알고리즘을 그림 3과 같이 나타낼 수 있다. 그림 3에서 출력 데이터 흐름을 제어하기 위해 하나의 스위칭 박스를 두고 있으며, DCT 및 DFT 계산은 단일 칩 또는 단일 연산처리 구조로 설계할 수 있다. 이는 비디오 코딩과 디지털 변조를 위한 단일 칩 설계에 있어서 매우 유용한 방법이 될 것으로 기대한다.

$$\begin{aligned}
 [\tilde{F}]_N &= \begin{pmatrix} \frac{F_N}{2} & 0 \\ 0 & Pr_{N/2} \frac{F_N}{2} W_{N/2} \end{pmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \\
 [\tilde{F}]_N &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pr_{N/2} \end{bmatrix} \begin{bmatrix} \frac{F_N}{2} & 0 \\ 0 & \frac{F_N}{2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}
 \end{aligned}
 \tag{25}$$

결과적으로 DFT 행렬에 대한 반복식의 일반형은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 [F]_N &= ([Pr]_N)^{-1} [\tilde{F}]_N \\
 ([Pr]_N)^{-1} &= \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pr_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/4} \otimes I_2 & 0 \\ 0 & Pr_2 \end{bmatrix} \begin{bmatrix} I_{N/2} \otimes F_2 \\ I_{N/4} \otimes W_2 \end{bmatrix} \\
 &= \begin{bmatrix} I_{N/4} \otimes I_2 & 0 \\ 0 & Pr_2 \end{bmatrix} \begin{bmatrix} I_{N/4} \otimes I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}
 \end{aligned}
 \tag{26}$$

명백하게 식 (26)은 식 (19)와 같은 형태이며, 여기서 단지 K_i 을 Pr_i 로, D_i 을 W_i 로 바꾸면 된다. 단, $i \in \{2, 4, 8, \dots, N/2\}$. 그림 2는 식 (26)과 관련된 버터플라이 데이터 흐름도를 나타낸다.

참고 문헌

- [1] K.R. Rao, and P. Yip. Jones, Discrete Cosine Transform Algorithms, Advantages, Applications, Academic Press, USA, 1990.
- [2] K.R. Rao, and J.J. Huwang, Techniques & Standards for Image Video & Audio Coding, Prentice Hall, USA, 1996.
- [3] M.H. Lee, "Simple systolic arrays for discrete cosine transform," Multidimensional system and processing, no.1, pp.389- 398, 1990, Kluwer Academic Publishers.
- [4] M.H. Lee, "On computing 2-D systolic algorithm for discrete cosine transform," IEEE Trans. on Circuit and systems, vol.37, no.10, Oct.1990.
- [5] Seung Son Kang, and Moon Ho Lee, "An expanded 2-D DCT algorithm based on convolution," IEEE Trans. on Consumer Electronics, vol.39, no.3, pp.159-165, 1993.
- [6] N. Ahmed and K.R. Rao, Orthogonal Transforms for Digital Signal Processing, Berlin, Germany: Springer-Verlag, 1975.
- [7] M.H. Lee, "High Speed Multidimensional Systolic Arrays for Discrete Fourier Transform," IEEE Trans. on Circuits Syst. II, vol. 39 no.12, pp 876-879, Dec.,1992
- [8] D.C. Park, M.H. Lee, and Euna Choi,

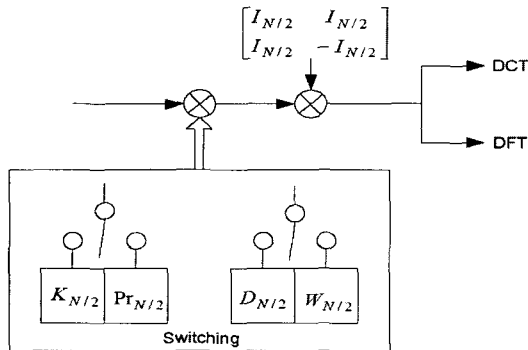


그림 3. 단순한 DCT/DFT 하이브리드 아키텍처

표 1. 연산 복잡도

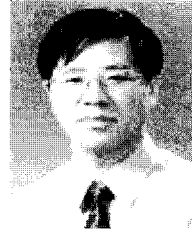
	Number of computation
FFT	Complex additions : $N \log_2 N$
DCT-II (Chen Wang)	Real additions : $3N/2 (\log_2 N - 1) + 2, M \geq 4$
DCT/DFT hybrid (proposed)	Real additions : $N \log_2 N + N/2 - 1$

“Revisited DFT matrix via the reverse jacket transform and its application to communication,” The 22nd symposium on Information theory and its applications (SITA 99), Yuzawa, Niigata, Japan, Nov.30-Dec.3, 1999.

- [9] M.H. Lee, “The Center Weighted Hadamard Transform,” IEEE Trans. On Circuits and Systems, vol. 36, no.9, pp.1247~1249, Sep. 1989
- [10] Chih-Peng Fan, Jar-Ferr Yang, “Fast Center Weighted Hadamard Tranform Algorithm”, IEEE Trans.on CAS-II, vol.45, No.3, pp.429~432, March. 1998.
- [11] S. R. Lee, J. H. Yi, “Fast Reverse Jacket Transform as an Altenative Representation of N point Fast Fourier Transform,” Journal of Mathematical Imaging and Vision, KL1419-03, pp.1413-1420, Nov. 2001.
- [12] M.H. Lee, B. S. Rajan, and J.Y. Park, “A Generalized Reverse Jacket Transform,” IEEE Trans. Circuits Syst. II, vol.48, no. 7, pp.684-690, July, 2001.
- [13] M.H. Lee, “A New Reverse Jacket Transform and Its Fast Algorithm,” IEEE, Trans. On Circuit and System, vol. 47. no. 1, pp.39-47, Jan. 2000.
- [14] M.H. Lee, and Ken Finlayson, “A simple element inverse Jacket transform coding,” IEEE Information Theory Workshop 2005, ITW 2005, Rotorua, New Zealand, 29 Aug - 1st Sept. 2005.
- [15] Chen, Wang, “High throughput VLSI architecures for the 1-D and 2-D discrete cosine transforms”, IEEE Trans. Circuits Syst. Video Technol., 1995.

박 대 철 (Dae-chul Park)

정회원



1977년 2월 서강대학교 전자 공
학과 졸업
1985년 12월 Univ. of New
Mexico 전자공학과 석사
1989년 5월 Univ. of New
Mexico 전자공학과 박사
1993년 9월~현재 한남대학교 정

보통신공학과 교수

<관심분야> 영상해석, 신호처리

이 문 호 (Moon-ho Lee)

정회원



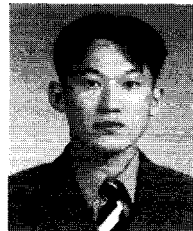
1967년 2월 전북대학교 전기 공
학과 졸업
1976년 2월 전북대학교 전기 공
학과 석사
1990년 2월 일본 Tokyo University
정보통신공학과 박사
1981년 3월~현재 전북대학교 전

자정보공학부 교수

<관심분야> 이동통신, 정보이론, UWB

신 태 철 (Tae-chol Sin)

준회원



1998년 2월 원광대학교 전자 공
학과 졸업
2001년 2월 전북대학교 정보 통
신공학과 석사
2007년 3월~현재 전북대학교 정
보통신공학과 박사과정

<관심분야> 데이터통신, OFDM,

Synchronization

이 광 재 (Kwang-jae Lee)

정회원



1986년 2월 전북대학교 전자 공
학과 졸업
1990년 2월 전북대학교 전자 공
학과 석사
2006년 2월 전북대학교 전기 공
학과 박사

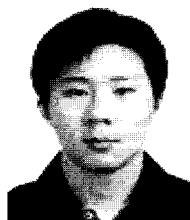
1995년 3월~현재 한려대학교 멀

티미디어정보통신공학과 전임강사

<관심분야> 이동통신, 부호이론, WLAN

진 주 (Zhu Chen)

준회원



2006년 7월 중남민족대학 자동차
공학과 졸업
2006년 9월 전북대학교 전자 정
보공학부 석사과정

<관심분야>