이산 코사인 변환의 소인수 분해 계산을 위한 인덱스 매핑

°이 병 기
서울대학교 전자공학과

Index Mappings for a Prime-Factor-Decomposed Computation of Discrete Cosine Transform

Byeong Gi Lee
Seoul National University, Department of Electronics Engineering

요 약

본 논문은 N이 서로소인 두 자연수 N_1과 N_2로 분해되는 경우에 대하여 이산코사인 변환의 소인수 분해 계산기법 및 이를 위한 인덱스 매핑을 고찰하고 있다. 소인수 분해 알고리즘의 유도과정에 나타나는 관계식들을 표의 형태로 변형시키고, 이 인덱스 매핑 표들을 바탕으로 이산 코사인 변환을 구현하는 방법이 소개되어 있다.

1. 서론

1974년 Ahmed, Natarjan, Rao에 의해서 처음 소개된 이래, 이산 코사인 변환(DCT)은 음성과 영상신호처리 분야에 주로 응용되어 왔다 [1]-[5]. 이를 보조하기 위하여, 이산 코사인 변환의 고속 계산기법도 병행하여 발전되어 왔다 [6]-[9]. 최근에 이르러서는 $N=2^m$ 형의 DCT를 FFT와 근사한 방법으로 계산하는 기법이 소개되었고 [10], 이어서 $N=N_1 \cdot N_2$ 형의 DCT를 N_1개의 N_2점 DCT와 N_2개의 N_1점 DCT로 분해시켜 계산하는 소인수 알고리즘 (prime factor algorithm) 기법도 소개되었다 [11]. 이 두가지 계산 기법들은 곱셈수를 감소시킨다는 점 이외에도 계산구조가 체계적이라는 장점을 함께 갖는다. 다만, 소인수 알고리즘의 경우, 전단과 후단에 행하는 인덱스 매핑(index mapping)이 복잡하다는 어려움이 있을 뿐이다. 본 논문에서는 표를 이용해서 이 인덱스 매핑들을 수행하는 방법을 한가지 소개하고자 한다.

2. 이산 코사인 변환의 소인수 분해

시간영역 데이타를 x(n), n=0,1, ... , N-1, 변환영역 데이타를 X(k), k=0,1, ... , N-1 이라 할때, 이산 코사인 변환은

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

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

의 관계로 정의된다. 이때 e(k)는

$$e(k) = \begin{cases} 1/\sqrt{2}, & k=o \text{ 일때}, \\ 1\ , & \text{그 밖의 경우} \end{cases} \quad (3)$$

의 관계를 갖는다.

식 (2)를 구현하는 것은 식 (1)을 구현한 플로우그라프에서 화살표의 방향만 바꾸면 되므로, 또 식 (1)에서 e(k)는 데이타 X(k)를 약간 수정시키는 것에 불과하므로, 본 논문에서는 식(1)을 변형하여

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

를 구현하는 방법만을 고찰하기로 한다. 또 본 논문에서는 N이 서로소인 N_1과 N_2의 곱으로 나타날 수 있는 경우만을 취급하기로 한다.

만일 식(4)의 N점 DCT(실제로는 IDCT에 더 가깝지만, 본 논문에서는 편의상 DCT라 부르기로 한다.)가 N_2개의 N_1점 DCT와 N_1개의 N_2점 DCT로 직렬 분해된다면, 식(4)는

$$x(n_1,n_2) = \sum_{k_2=0}^{N_2-1} \left\{ \sum_{k_1=0}^{N_1-1} X(k_1,k_2)\cos\,[\pi(2n_1+1)k_1/2N_1] \right\} \cos\,[\pi(2n_2+1)k_2/2N_2], \quad (5)$$

$$n_1 = 0,\ 1,\ \ldots\ ,\ N_1-1,$$
$$n_2 = 0,\ 1,\ \ldots\ ,\ N_2-1$$

의 형태로 변형될 것이다. 이때 x(n)과 $x(n_1,n_2)$를 연관시켜 주는 것을 시간영역측 인덱스 매핑 또는 시간인덱스 매핑, X(k)와 $X(k_1,k_2)$를 연결시켜주는 것을 변환영역측 인덱스 매핑 또는 변환 인덱스 매핑이라고 각각 호칭하기로 한다.

3. 인덱스 매핑

먼저 $X(k)$와 $X(k_1,k_2)$ 간을 잇는 변환영역측의 인덱스 매핑을 살펴보기로 하자. 집합 $\mathcal{N}, \mathcal{N}_1, \mathcal{N}_2$를 각각 0부터 $N-1$, N_1-1, N_2-1 까지의 자연수로 구성되는 인덱스의 집합이라고 정의하고, "k modulo N"을 $\langle k\rangle_N$으로 표기하기로 한다. 이를 바탕으로 $\mathcal{N}_1\times\mathcal{N}_2$와 $\mathcal{N}$을 잇는 매핑 $\hat{f}$와 $\tilde{f}$를 각각

$$\hat{f}(k_1,k_2)=\begin{cases}\langle k_1N_2+k_2N_1\rangle_{2N}, & \langle k_1N_2+k_2N_1\rangle_{2N} < N \text{ 일때,}\\ 2N-\langle k_1N_2+k_2N_1\rangle_{2N}, & \text{그 밖의 경우,}\end{cases} \quad (6a)$$

$$\tilde{f}(k_1,k_2)=\begin{cases}\langle |k_1N_2-k_2N_1|\rangle_{2N}, & \langle |k_1N_2-k_2N_1|\rangle_{2N} < N \text{ 일때,}\\ 2N-\langle k_1N_2+k_2N_1\rangle_{2N}, & \text{그 밖의 경우}\end{cases} \quad (6b)$$

로 정의하고, 이에 따른 집합 $\hat{\mathcal{N}}$와 $\tilde{\mathcal{N}}$을 각각

$$\hat{\mathcal{N}}=\{k \mid k=\hat{f}(k_1,k_2),\ k_1\in\mathcal{N}_1,\ k_2\in\mathcal{N}_2\}, \quad (7a)$$
$$\tilde{\mathcal{N}}=\{k \mid k=\tilde{f}(k_1,k_2),\ k_1\in\mathcal{N}_1,\ k_2\in\mathcal{N}_2\} \quad (7b)$$

로 정의한다. 그러면

$$\hat{\mathcal{N}} \ Ⓤ \ \tilde{\mathcal{N}} = \mathcal{N} \ Ⓤ \ \mathcal{N} \quad (8)$$

의 관계가 존재한다. 이때 Ⓤ는 열거형 합집합으로 양쪽 집합의 원소를 모두 열거하여 (중복이 되더라도) 만든 합집합을 나타낸다.

따라서 식(4)는

$$x(n)=\frac{1}{2}\sum_{k\in\hat{\mathcal{N}}} s(k)\,X(k)\cos[\pi(2n+1)k/2N] + \frac{1}{2}\sum_{k\in\tilde{\mathcal{N}}} X(k)\cos[\pi(2n+1)k/2N]. \quad (9)$$

로 변형될 수 있고, 이때 $s(k)$는

$$s(k)=\begin{cases}1, & \langle k_1N_2+k_2N_1\rangle_{2N}<N \text{ 일때,}\\ -1, & \text{그밖의 경우}\end{cases} \quad (10)$$

로 정의된다.

또 $X(k_1,k_2)$를

$$X(k_1,k_2)=\begin{cases} s(k)X(k)\big|_{k=\hat{f}(k_1,k_2)} = X(k)\big|_{k=\tilde{f}(k_1,k_2)}, & k_1=0 \text{ 또는 } k_2=0 \text{ 일때,}\\ s(k)X(k)\big|_{k=\hat{f}(k_1,k_2)} + X(k)\big|_{k=\tilde{f}(k_1,k_2)}, & \text{그 밖의 경우}\end{cases} \quad (11)$$

로 정의하면, 이 식은 다시

$$x(n)=\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X(k_1,k_2)\cos[\pi(2n+1)k_1/2N_1]\cdot\cos[\pi(2n+1)k_2/2N_2] \quad (12)$$

로 변형됨을 곧 알 수 있다.

이어서, $x(n)$과 $x(n_1,n_2)$ 간을 잇는 시간영역측 인덱스 매핑을 살펴보기로 하자. $\mathcal{N}$과 $\mathcal{N}_1\times\mathcal{N}_2$를 잇는 매핑 g를

$$(n_1,n_2)=g(n),\ n_1\in\mathcal{N}_1,\ n_2\in\mathcal{N}_2,\ n\in\mathcal{N}, \quad (13a)$$

$$n_1=\begin{cases}\langle n\rangle_{2N_1}, & \langle n\rangle_{2N_1}<N_1 \text{ 일때,}\\ 2N_1-1-\langle n\rangle_{2N_1}, & \text{그 밖의 경우,}\end{cases} \quad (13b)$$

$$n_2=\begin{cases}\langle n\rangle_{2N_2}, & \langle n\rangle_{2N_2}<N_2 \text{ 일때,}\\ 2N_2-1-\langle n\rangle_{2N_2}, & \text{그 밖의 경우}\end{cases} \quad (13c)$$

로 정의하면, g는 일대일 대응이 된다. 따라서 $x(n_1,n_2)$를

$$x(n_1,n_2)=x(n)\big|_{n=g^{-1}(n_1,n_2)} \quad (14)$$

로 정의하면, 식(12)는 곧 식(5)와 같게 된다.

그러므로 식(4)는 식(11)과 식(14)의 인덱스 매핑들을 통해서 식(5)로 분해됨을 알 수 있다.

4. 인덱스 매핑 표

본 절에서는 앞서 정의된 변환영역측 및 시간영역측의 인덱스 매핑을 각각 표로 나타내는 방법을 찾아보도록 하자.

먼저 변환인덱스 매핑편을 위해서 N_1개의 행과 N_2개의 열로 구성되는 $\hat{k}$-표 및 $\tilde{k}$-표를 다음과 같이 정의한다.

$$\hat{k}(k_1,k_2) = \hat{f}(k_1,k_2), \quad \text{식(15a)}$$
$$\tilde{k}(k_1,k_2) = \tilde{f}(k_1,k_2). \quad \text{식(15b)}$$

단 $k_1=0, 1, \ldots, N_1-1$, $k_2=0, 1, \ldots, N_2-1$ 이고, $\hat{k}(k_1,k_2)$ 와 $\tilde{k}(k_1,k_2)$는 각각 $\hat{k}$-표와 $\tilde{k}$-표의 k_1행 k_2 열에 있는 성분을 나타낸다. 이 두식은 식(6a)와 식(6b)를 표의 형태로 변환시키는 식들이다. 이때 두 표들은 $k_1=0$ 또는 $k_2=0$ 일때

$$\hat{k}(k_1,k_2) = \tilde{k}(k_1,k_2) \quad \text{식(16a)}$$

이고 그밖의 경우에 대해서는

$$\hat{k}(k_1,k_2) = \hat{k}(N_1-k_1,\ N_2-k_2), \qquad 식(16b)$$

$$\tilde{k}(k_1,k_2) = \tilde{k}(N_1-k_1,\ N_2-k_2) \qquad 식(16c)$$

의 대칭성을 갖는다. 이 성질들을 바탕으로 k_V 표와 k_H-표를 다음과 같이 정의한다.

$$k_V(k_1,k_2) = \begin{cases} \hat{k}(k_1,k_2), & k_1=0 \text{ 또는 } k_2=0 \text{ 일때}, \\ \tilde{k}(k_1,k_2), & \text{세로방향 상반부}, \\ \hat{k}(k_1,k_2), & \text{세로방향 하반부}. \end{cases} \qquad 식(17a)$$

$$k_H(k_1,k_2) = \begin{cases} \hat{k}(k_1,k_2), & k_1=0 \text{ 또는 } k_2=0 \text{ 일때}, \\ \tilde{k}(k_1,k_2), & \text{가로방향 상반부}, \\ \hat{k}(k_1,k_2), & \text{가로방향 하반부}. \end{cases} \qquad 식(17b)$$

이때 세로방향 상반부는 $k_1=1,\ 2,\ \ldots,\ N_1-1,\ k_2=1,2,\ldots,N_2-1$에 해당하는 부분표의 구성 성분들을 세로방향으로 세어서 처음 절반에 해당하는 부분을 나타내며, 가로방향 상반부도 이와 유사하게 가로방향으로 정해진다.

이제 이 두 표들을 이용해서 인덱스 매핑을 도시하는 방법을 검토해보기로 하자. 우선 N_2 개의 N_1 점 DCT를 먼저 수행하고 이어서 N_1 개의 N_2 점 DCT를 수행하는 경우를 살펴보자. 이 경우 입력 $X(k)$는 $X(0,0),\ X(1,0),\ \ldots,\ X(N_1-1,0);\ X(0,1), X(1,1),\ \ldots,\ X(N_1-1,1);\ldots;X(0,N_2-1),\ X(1,N_2-1),\ \ldots,\ X(N_1-1,N_2-1)$ 순으로 배열되어야 하고, 이것은 표를 세로방향으로 읽어내는 것에 해당한다. 그러므로 k_V-표를 취하여 세로방향으로 읽어낸 인덱스 k를 이용해서 $X(k)$를 $X(k_1,k_2)$ 와 나란히 배열시키고, 이를 바탕으로 식(11)을 플로우그래프로 나타내면 다음과 같은 모양이 형성된다: $k_1=0$ 또는 $k_2=0$ 일때는 $X(k_1,k_2)$가 이와 나란히 있는 $X(k)$와 이득이 1인 선으로 연결되고, 이 밖의 경우에는 $X(k_1, k_2)$, $X(N_1-k_1\ ,N_2-k_2)$과 이 두점에 상응하는 $X(k)$의 두점이 나비모양을 형성한다. 이때 각 선의 이득은 식(10)으로 정의되는 대로 1이나 -1이 된다. N_1개의 N_2 점 DCT를 먼저 수행할 경우에는 k_H-표를 취해서 가로방향으로 읽어내고 위와 같은 절차를 따르면 된다.

시간영역측 인덱스 매핑을 위해서는 다음과 같이 정의된 n-표를 이용한다.

$$n(n_1,n_2) = g^{-1}(n_1,n_2). \qquad 식(18)$$

이때 $n_1=0,1,\ldots,N_1-1,\ n_2=0,1,\ldots,N_2-1$ 이고 $n(n_1,n_2)$는 앞서와 마찬가지로 n-표의 n_1 행 n_2 열 성분을 나타낸다. 이것은 식(13b)와 식(13c)를 표의 형태로 변환시킨 것이다.

이 표를 사용해서 시간영역측 인덱스 매핑을 도시하는 일은 비교적 간단하다. N_1개의 N_2 점 DCT를 나중에 수행하는 경우를 살펴보면, 이때 출력 $x(n)$은 $x(0,0),\ x(0,1),\ldots,x(0,N_2-1);x(1,0),x(1,1),\ldots,x(1,N_2-1);\ldots;x(N_1-1,0), x(N_1-1,1),\ldots,x(N_1-1,N_2-1)$ 순으로 배열된다. 그러므로 n-표를 가로방향으로 읽어내어 $x(n)$의 인덱스를 삼아서 $x(n_1, n_2)$와 나란히 배열시키고 이득 1의 선으로 연결시키면 시간영역측 인덱스 매핑이 완성된다. N_2개의 N_1 점 DCT를 나중에 수행하는 경우엔 n-표를 세로방향으로 읽어내서 이를 반복하면 된다.

5. 예제

앞절에서 설명한 인덱스 매핑 표들을 $N=12,\ N_1=3,\ N_2=4$의 경우를 예로 작성해보기로 하자.

식(15a)와 식(15b)에 정의된 대로 $\hat{k}$-표와 $\tilde{k}$-표를 작성하면, 각각 표-1(a),(b)를 얻는다. 이 표들로 부터 식(16a), 식(16b), 식(16c)의 관계들이 성립함을 볼 수 있다. 이어서 식(17a)와 식(17b)에 정의된 대로 k_V-표와 k_H-표를 작성하면 각각 표-1(c),(d)를 얻는다.

끝으로 식(18)에 정의된 n-표를 작성하면 표-1(e)와 같이 된다.

그러므로 윗 k_V-표와 n-표(가로방향)을 이용하면 3점 DCT들이 먼저 수행되고 4점 DCT들이 나중에 수행되는 (그림1)에 보인 것과 같은 계산구조를 찾아내게 된다. 또 k_H-표와 n-표(세로방향)을 이용하면 (그림2)에 보인 것과 같이 4점 DCT들이 먼저 수행되는 계산구조를 얻는다.

6. 결 론

본 논문에서는 N이 서로소인 두 자연수 N_1과 N_2로 분해되는 경우에 대하여 이산코사인 변환의 소인수 분해 계산기법 및 이를 위한 인덱스 매핑에 관하여 살펴보았다. 소인수 분해 알고리즘의 유도과정에 나타난 수식들을 표의 형태로 변형시키고, 이를 바탕으로 이산코사인 변환을 구현하는 방법을 소개하였다. 이 방법을 잘 이용하면 그동안 소인수 분해 계산기법을 활용하는데 장애가 되었던 인덱스 매핑 문제가 쉽게 해결될 수 있으리라 믿는다.

참고문헌

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

[2] R.Zelinski and P.Noll, "Adaptive transform coding of speech signals," IEEE Trans. Acoustics, Speech and Signal Processing, vol. ASSP-25, pp. 299-309, Aug. 1977.

[3] J.M.Tribolet and R.E.Crochiere, "Frequency domain coding of speech," IEEE Trans. Acoustics, Speech and Signal Processing, vol. ASSP-27, pp. 512-530, Oct. 1979.

[4] W.H.Chen and C.H.Smith, "Adaptive coding of monochrome and color images," IEEE Trans. Communications, vol. COM-25, pp. 1285-1292, Nov. 1977.

[5] J.A.Roese and G.S.Robinson, "Interframe cosine transform image coding," IEEE Trans. Communications, vol. COM-25, pp. 1329-1339, Nov. 1977.

[6] B.D.Tseng and W.C.Miller, "On computing the discrete cosine transform," IEEE Trans. Computers, vol. C-27, pp. 966-968, Oct. 1978.

[7] W.H.Chen, C.H.Smith and S.C.Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Communications, vol. COM-25, pp. 1004-1009, Sept. 1977.

[8] M.J.Narasimha and A.M.Peterson, "On the computation of discrete cosine transform," IEEE Trans. Communications, vol. COM-26, pp. 934-936, June 1978.

[9] 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.

[10] B.G.Lee, "A new algorithm to compute the discrete cosine transform," IEEE Trans. Acoustics Speech and Signal Processing, vol. ASSP-32, pp. 1243-1245, Dec. 1984.

[11] P.P.N.Yang, M.J.Narasimha and B.G.Lee, "A prime factor decomposition algorithm for the computation of the discrete cosine transform," in Proc. IEEE Int'l Conference on Computers, Systems and Signal Processing, Dec. 1984.

(표-1) 인덱스 매핑 표

(a) $\hat{k}$-표

$k_1 \backslash k_2$	0	1	2	3
0	0	3	6	9
1	4	7	10	11
2	8	11	10	7

(b) $\tilde{k}$-표

$k_1 \backslash k_2$	0	1	2	3
0	0	3	6	9
1	4	1	2	5
2	8	5	2	1

(c) k_V-표

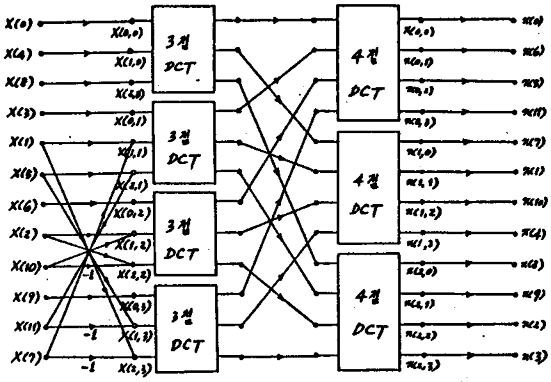
$k_1 \backslash k_2$	0	1	2	3
0	0	3	6	9
1	4	1	2	11
2	8	5	10	7

(d) k_H-표

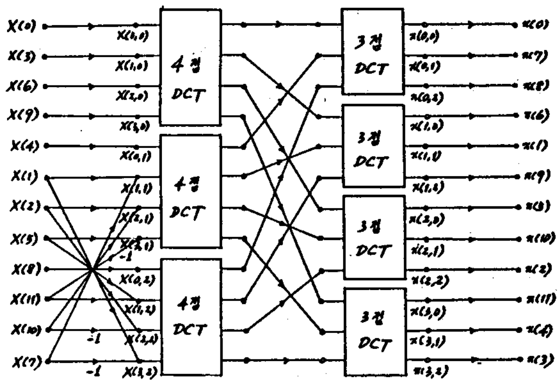
$k_1 \backslash k_2$	0	1	2	3
0	0	3	6	9
1	4	1	2	5
2	8	11	10	7

(e) n-표

$n_1 \backslash n_2$	0	1	2	3
0	0	6	5	11
1	7	1	10	4
2	8	9	2	3



(그림 1) 12점 DCT : 3점 DCT + 4점 DCT



(그림 2) 12점 DCT : 4점 DCT + 3점 DCT