論文 84－21－6－10

DST를 위한 高速 計算 알고리즘

(A Fast Computational Algorithm for the Discrete Sine Transform)

郭 勳 星*, 申 建 淳**

(Hoon Sung Kwak and Gun Soon Shin)

要 約

本 論文은 Kekre와 Solanki가 定義한 discrete sine transform(DST)의 高速 計算 알고리즘을 提示하였다.

技巧들은 DST메트릭스를 M=log₂ 2N 개의 메트릭스로 계승화하여 구현하였으며 샘플 데이타 수 N은 2의 累乘이다. 계승화된 각 인수 메트릭스는 어느 행열에서도 零아닌 실수값을 둘이상 포함하지 않는다. 이 방법의 결과로서 DST의 高速 計算을 위한 정확한 알고리즘이 얻어졌다. 또한 본 알고리즘은 하드웨어 또는 소프트웨어 구현이 될 수 있도록 신호흐름선도로 예시되었다.

Abstract

This paper represents a fast computational algorithm for the discrete sine transform defined by Kekre and Solanki. Techniques are developed to factor the discrete sine transform matrix into $M=\log_2 2N$ matrices, where the number(N) of sampled data points is a power of two. Each factorial matrix contains not more than two non-zero real elem ents in any row or column. As a result of this method, the exact algorithm for the fast discrete sine transform is accomplished. The algorithm is illustrated by a signal flow graph, which may be readily translated to hardware or software implementation.

I. 序 論

最近 디지탈 이미지 처리에 變換(transform) 技術이 導入되고 있다.[1,2,4,5]

푸리에 變換은 高速푸리에 變換(fast Fourier transform) 알고리즘이 發表되므로서 더욱 널리 使用되고 있음은 잘 알려진 事實이다.

Karhunen-Loeve 變換(KLT)은 이미지처리에 있어 統計的으로 最適인 性質을 가지나 高速 計算 알고리즘을 찾기 어려워 KLT에 가까운 變換들이 研究 發表되고 있다.[7,8,9]

1974年 Ahmed와 Rao[7] 등은 discrete cosine transform(DCT)을 提案하였으며 1976年 Jain[8]은 discrete sine transform 1(DST 1)을 定義하였다.

그후 1978年 Kekre와 Solanki[9]는 DST1과는 性質이 다른 discrete sine transform 2(DST2)를 다시 定義하였다.

1977年 Chen[10] 등과 1983年 곽[14] 등은 DCT에 대한 高速 알고리즘을 提示하였으며 1980年 Yip와 Rao[11]는 DST1에 對한 高速 알고리즘을, 그리고 1982年 Wang[12]은 DST2에 對한 高速 알고리즘을 Chen 의 方法과 類似한 方法으로 各各 發表하였다.

*正會員, 全北大學校 工科大學 電氣工學科

(Dept. of Electrical Eng., Chonbuk National Univ.)

**正會員, 金烏工大 電子工學科

(Kum Oh Institute of Technology)

接受日字：1984年 9月 5日

그러나 Chen이 提案한 高速 알고리즘은 近似化 알고리즘이므로 Wang에 의한 DST 2 高速 알고리즘 역시 誤差를 갖는다.

이러한 誤差는 어떤 파라메타 例를 들면, 상관계수(ρ)의 變化에 따라 變換들의 特性과 성능에 영향을 준다.[13]

本 硏究는 定義된 DST 2 에 일치하는 高速 알고리즘에 대해 고찰한다. 샘플데이타 N이 8인 경우[15]와 똑같은 方法을 N이 16인 경우에 적용하였으며 나아가 더 큰 N에 대해서도 확장할 수 있음을 고찰하였다.

또 計算 結果를 定義된 DST 2 의 값들과 일치되는지 여부를 확인하였으며, 誤差의 程度를 알기 위하여 Wang의 값들과 비교하였다. 高速 變換과 逆變換을 信號흐름線圖로 제시하였고 다른 高速 變換 알고리즘들과 비교하였다.

Ⅱ. DST 2 高速 計算 알고리즘

지금 샘플데이타를 N개씩 취하여 變換시킨다고 할 때 Kekre와 Solanki에 의해 定義된 DST 2 는 다음式(1)과 같다.

$$\Phi_{ij} = \sqrt{\frac{2}{N}}\sin\left[\frac{(i+1)(2j+1)\pi}{2N}\right] \quad \begin{matrix} i=0,1,\cdots,N-2 \\ j=0,1,\cdots,N-1 \end{matrix}$$
$$= (-1)^{j}(N)^{-1/2} \quad \begin{matrix} i=N-1 \\ j=0,1,\cdots,N-1. \end{matrix} \qquad (1)$$

여기서 N은 高速 計算을 위하여

$$N=2^{n} \qquad (2)$$

을 택한다. 단, n은 正의 整數

N이 16일 때 式(1)로 부터 얻은 變換 메트릭스$[\Phi_{ij}]_{16}$은 다음 式(3)과 같다.

$$[\Phi_{ij}]_{16} = \sqrt{\frac{2}{16}}\begin{bmatrix} \sin\frac{\pi}{32} & \sin\frac{3\pi}{32} & \sin\frac{5\pi}{32} & \cdots & \sin\frac{31\pi}{32} \\ \sin\frac{2\pi}{32} & \sin\frac{6\pi}{32} & \sin\frac{10\pi}{32} & \cdots & \sin\frac{62\pi}{32} \\ \sin\frac{3\pi}{32} & \sin\frac{9\pi}{32} & \sin\frac{15\pi}{32} & \cdots & \sin\frac{93\pi}{32} \\ \vdots & \vdots & \vdots & & \vdots \\ \sin\frac{14\pi}{32} & \sin\frac{42\pi}{32} & \sin\frac{70\pi}{32} & \cdots & \sin\frac{434\pi}{32} \\ \sin\frac{15\pi}{32} & \sin\frac{45\pi}{32} & \sin\frac{75\pi}{32} & \cdots & \sin\frac{465\pi}{32} \\ 0.707 & -0.707 & 0.707 & \cdots & 0.707 \end{bmatrix} \qquad (3)$$

여기서 Sn, Cn, C_1 等을 다음과 같이 놓는다.

즉,
$$Sn = \sin\frac{(n-1)\pi}{32}$$
$$Cn = \cos\frac{(n-1)\pi}{32} \qquad (4)$$
(단, $n=2,3,\cdots\cdots$)
$$C_1 = 0.7071$$

$[\Phi_{ij}]_{16}$의 sin 값들은 표 1에 나타난 값을 계속 반복가지므로 Sn과 Cn으로 表示할 수 있다. 여기서 n은 9 以下이다.

표 1. N=16 DST 2 을 위한 곱셈기

Table 1. Multipliers for 16-point discrete sine transform 2.

n	Cn	Sn
2	0.9951	0.0980
3	0.9807	0.1950
4	0.9569	0.2902
5	0.9238	0.3826
6	0.8819	0.4713
7	0.8314	0.5555
8	0.7730	0.6343
9	0.7071	

이제 계승 메트릭스로 만들기 위해 $[\Phi_{ij}]_{16}$ 메트릭스를 式(5)와 같은 우수 열 부호교환 메트릭스 $[W_{ij}]_{16}$을 곱하여 그림 1과 같은 $[\phi_{ij}]_{16}$를 얻는다.

$$\begin{array}{cc} & 0 \longrightarrow j \quad 15 \\ [W_{ij}]_{16} = & \begin{array}{c} 0 \\ \downarrow \\ i \\ 15 \end{array}\begin{bmatrix} 1 &&&&&&&&&&&&&&& \\ & -1 &&&&&&&&&&&&&& \\ && 1 &&&&&&&&&&&&& \\ &&& -1 &&&&&&&&&&&& \\ &&&& 1 &&&&&&&&&&& \\ &&&&& -1 &&&&&&&&&& \\ &&&&&& 1 &&&&&&&&& \\ &&&&&&& -1 &&&&&&&& \\ &&&&&&&& 1 &&&&&&& \\ &&&&&&&&& -1 &&&&&& \\ &&&&&&&&&& 1 &&&&& \\ &&&&&&&&&&& -1 &&&& \\ &&&&&&&&&&&& 1 &&& \\ &&&&&&&&&&&&& -1 && \\ &&&&&&&&&&&&&& 1 & \\ &&&&&&&&&&&&&&& -1 \end{bmatrix} \end{array} \qquad (5)$$

이제 삼각함수의 公式 (6)은 式(7)과 같은 관계에 있다.

$$\sin nx\ \sin mx = \frac{1}{2}\cos(n-m)x - \frac{1}{2}\cos(n+m)x$$
$$\cos nx\ \cos mx = \frac{1}{2}\cos(n-m)x + \frac{1}{2}\cos(n+m)x \qquad (6)$$
$$\cos nx\ \sin mx = \frac{1}{2}\sin(n+m)x - \frac{1}{2}\sin(n-m)x$$

$$S_mS_n = \frac{1}{2}(C_{m-n+1} - C_{m+n-1})$$
$$C_mC_n = \frac{1}{2}(C_{m-n+1} + C_{m+n-1}) \qquad (7)$$
$$C_mS_n = \frac{1}{2}(S_{m+n-1} - S_{m-n+1})$$

$$S_mC_n = \frac{1}{2}(S_{m+n-1} + S_{m-n+1})$$

$$0 \xrightarrow{\hspace{6cm}} j \qquad 15$$

$$\begin{array}{c} 0 \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ i \\ \\ 15 \end{array}\left[\begin{array}{cccccccccccccccc}
S_2 & -S_4 & S_6 & -S_8 & C_8 & -C_6 & C_4 & -C_2 & C_2 & -C_2 & C_6 & -C_8 & S_8 & -S_6 & S_4 & -S_2 \\
S_3 & -S_7 & C_7 & -C_3 & C_3 & -C_7 & S_7 & -S_3 & -S_3 & S_7 & -C_7 & C_3 & -C_3 & C_7 & -S_7 & S_3 \\
S_4 & -C_8 & C_2 & -C_6 & S_6 & S_2 & -S_8 & C_4 & -C_4 & S_8 & -S_2 & -S_6 & C_6 & -C_2 & C_8 & -S_4 \\
S_5 & -C_5 & C_5 & -S_5 & -S_5 & C_5 & -C_5 & S_5 & S_5 & -C_5 & C_5 & -S_5 & -S_5 & C_5 & -C_5 & S_5 \\
S_6 & -C_2 & S_8 & S_4 & -C_4 & C_8 & S_2 & -C_6 & C_6 & -S_2 & -C_8 & C_4 & -S_4 & -S_8 & C_2 & -S_6 \\
S_7 & -C_3 & S_3 & C_7 & -C_7 & -S_3 & C_3 & -S_7 & -S_7 & C_3 & -S_3 & -C_7 & C_7 & S_3 & -C_3 & S_7 \\
S_8 & -C_6 & -S_2 & C_2 & -S_2 & -C_4 & S_6 & C_8 & -C_8 & -S_6 & C_4 & S_2 & -C_2 & S_4 & C_6 & -S_8 \\
C_9 & -C_9 & -C_9 & C_9 & C_9 & -C_9 & -C_9 & C_9 & C_9 & -C_9 & -C_9 & C_9 & C_9 & -C_9 & -C_9 & C_9 \\
C_8 & -S_6 & -C_4 & S_2 & C_2 & S_4 & -C_6 & -S_8 & S_8 & C_6 & -S_4 & -C_2 & -S_2 & C_4 & S_6 & -C_8 \\
C_7 & -S_3 & -C_3 & -S_7 & S_7 & C_3 & S_3 & -C_7 & -C_7 & S_3 & C_3 & S_7 & -S_7 & -C_3 & -S_3 & C_7 \\
C_6 & S_2 & -C_8 & -C_4 & -S_2 & S_8 & C_2 & S_6 & -S_6 & -C_2 & -S_8 & S_4 & C_4 & C_8 & -S_2 & -C_6 \\
C_5 & S_5 & -S_5 & -C_5 & -C_5 & -S_5 & S_5 & C_5 & C_5 & S_5 & -S_5 & -C_5 & -C_5 & -S_5 & S_5 & C_5 \\
C_4 & S_8 & S_2 & -S_6 & -C_6 & -C_2 & -C_8 & -S_4 & S_4 & C_8 & C_2 & C_6 & S_6 & -S_2 & -S_8 & -C_4 \\
C_3 & C_7 & S_7 & S_3 & -S_3 & -S_7 & -C_7 & -C_3 & -C_3 & -C_7 & -S_7 & -S_3 & S_3 & S_7 & C_7 & C_3 \\
C_2 & C_4 & C_6 & C_8 & S_8 & S_6 & S_4 & S_2 & -S_2 & -S_4 & -S_6 & -S_8 & -C_8 & -C_6 & -C_4 & -C_2 \\
C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1
\end{array}\right]$$

그림 1. DST 2, N=16
Fig. 1. Discrete sine transform 2, N=16.

式(7)을 利用하여 그림 1 의 $-S_4, S_6\ -S_8$ 等의 값들을 다음 式의 값들로 변형시킨다. 즉, C_5S_2 는

$$C_5S_2 = \frac{1}{2}(S_6 - S_4) \qquad (8-a)$$

$$S_5C_2 = \frac{1}{2}(S_6 + S_4) \qquad (8-b)$$

그러므로

$$-S_4 = C_5S_2 - S_5C_2 \qquad (9)$$

또 (8-a)와 (8-b)를 합하여

$$S_6 = C_5S_2 + S_5C_2 \qquad (10)$$

이다. $-S_8$을 구하기 위해

$$S_9C_2 = \frac{1}{2}(S_{10} + S_8)$$

$$C_9S_2 = \frac{1}{2}(S_{10} - S_8)$$

그러므로

$$\begin{array}{c} -S_8 = C_9S_2 - S_9C_2 \\ \vdots \qquad\quad \vdots \end{array} \qquad (11)$$

이렇게 하여 얻은 式(9), (10), (11) 等의 값들을 그림 1 의 값들에 치환하므로서 다음 그림 2 와 같은 결과를 얻는다.

그림 2 는 그림 3 ($[A]_{16}$ 메트릭스)과 그림 4 ($[R]_{16}$ 메트릭스)와 같은 메트릭스 곱으로 놓을 수 있다.

$$0 \xrightarrow{\hspace{4cm}} j \qquad 15$$

$$\begin{array}{c} 0 \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ i \\ 15 \end{array}\left[\begin{array}{cccccccccccccccc}
 & -C_2 & & & & & & & & & & & & & S_2 & \\
 & & -C_3 & & & & & & & & & & & S_3 & & \\
 & & & -C_4 & & & & & & & & & S_4 & & & \\
 & & & & -C_5 & & & & & & & S_5 & & & & \\
 & & & & & -C_6 & & & & & S_6 & & & & & \\
 & & & & & & -C_7 & & & S_7 & & & & & & \\
 & & & & & & & -C_8 & S_8 & & & & & & & \\
-C_9 & & & & & & & & & & & & & & & C_9 \\
 & & & & & & & S_8 & C_8 & & & & & & & \\
 & & & & & & S_7 & & & C_7 & & & & & & \\
 & & & & & S_6 & & & & & C_6 & & & & & \\
 & & & & S_5 & & & & & & & C_5 & & & & \\
 & & & S_4 & & & & & & & & & C_4 & & & \\
 & & S_3 & & & & & & & & & & & C_3 & & \\
 & S_2 & & & & & & & & & & & & & C_2 & \\
S_1 & & & & & & & & & & & & & & & C_1
\end{array}\right]$$

그림 3. DST 2 의 $[A]_{16}$ 메트릭스
Fig. 3. Matrix $[A]_{16}$ of discrete sine transform 2.

또 그림 4 는 그림 5 ($[S]_{16}$ 메트릭스)와 그림 6 ($[T]_{16}$ 메트릭스)의 곱 메트릭스로 나눌 수 있는데 그림 5 와

$$0 \xrightarrow{\hspace{12cm}} j \qquad 15$$

$$\begin{array}{c} 0 \\ \\ \\ \\ \\ \\ \\ \\ \\ i \\ \\ 15 \end{array}\left[\begin{array}{cccccccccccccccc}
S_2 & \substack{S_2C_5\\-C_2S_5} & \substack{S_2C_5\\+C_2C_5} & \substack{S_2C_9\\-C_2C_9} & \substack{S_2C_9\\+C_2C_9} & \substack{S_2S_5\\-C_2C_5} & \substack{S_2S_5\\+C_2C_5} & -C_2 & C_2 & \substack{-S_2S_5\\-C_2C_5} & \substack{-S_2S_5\\+C_2C_5} & \substack{-S_2C_9\\-C_2C_9} & \substack{-S_2C_9\\+C_2C_9} & \substack{-S_2C_5\\-C_2S_5} & \substack{-S_2C_5\\+C_2S_5} & -S_2 \\
S_3 & \substack{S_3C_9\\-C_3C_9} & \substack{S_3C_9\\+C_3C_9} & -C_3 & C_3 & \substack{-S_3C_9\\-C_3C_9} & \substack{-S_3C_9\\+C_3C_9} & -S_3 & -S_3 & \substack{-S_3C_9\\+C_3C_9} & \substack{-S_3C_9\\-C_3C_9} & C_3 & -C_3 & \substack{S_3C_9\\+C_3C_9} & \substack{S_3C_9\\-C_3C_9} & S_3 \\
S_4 & \substack{S_4S_5\\-C_4C_5} & \substack{S_4S_5\\+C_4C_5} & \substack{-S_4C_9\\-C_4C_9} & \substack{-S_4C_9\\+C_4C_9} & \substack{-S_4C_5\\+C_4S_5} & \substack{-S_4C_5\\-C_4S_5} & C_4 & -C_4 & \substack{S_4C_5\\+C_4S_5} & \substack{S_4C_5\\-C_4S_5} & \substack{S_4C_9\\-C_4C_9} & \substack{S_4C_9\\+C_4C_9} & \substack{-S_4S_5\\-C_4C_5} & \substack{-S_4S_5\\+C_4C_5} & -S_4 \\
S_5 & -C_5 & C_5 & -S_5 & -S_5 & C_5 & -C_5 & S_5 & S_5 & -C_5 & C_5 & -S_5 & -S_5 & C_5 & -C_5 & S_5 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C_5 & S_5 & -S_5 & -C_5 & -C_5 & -S_5 & S_5 & C_5 & C_5 & S_5 & -S_5 & -C_5 & -C_5 & -S_5 & S_5 & C_5 \\
C_4 & \substack{C_4S_5\\+S_4C_5} & \substack{C_4S_5\\-S_4C_5} & \substack{-C_4C_9\\+S_4C_9} & \substack{-C_4C_9\\-S_4C_9} & \substack{-C_4C_5\\-S_4S_5} & \substack{-C_4C_5\\+S_4S_5} & -S_4 & S_4 & \substack{C_4C_5\\-S_4S_5} & \substack{C_4C_5\\+S_4S_5} & \substack{C_4C_9\\+S_4C_9} & \substack{C_4C_9\\-S_4C_9} & \substack{-C_4S_5\\+S_4C_5} & \substack{-C_4S_5\\-S_4C_5} & -C_4 \\
C_3 & \substack{C_3C_9\\+S_3C_9} & \substack{C_3C_9\\-S_3C_9} & S_3 & -S_3 & \substack{-C_3C_9\\+S_3C_9} & \substack{-C_3C_9\\-S_3C_9} & -C_3 & -C_3 & \substack{-C_3C_9\\-S_3C_9} & \substack{-C_3C_9\\+S_3C_9} & -S_3 & S_3 & \substack{C_3C_9\\-S_3C_9} & \substack{C_3C_9\\+S_3C_9} & C_3 \\
C_2 & \substack{C_2C_5\\+S_2S_5} & \substack{C_2C_5\\-S_2S_5} & \substack{C_2C_9\\+S_2C_9} & \substack{C_2C_9\\-S_2C_9} & \substack{C_2S_5\\+S_2C_5} & \substack{C_2S_5\\-S_2C_5} & S_2 & -S_2 & \substack{-C_2S_5\\+S_2C_5} & \substack{-C_2S_5\\-S_2C_5} & \substack{-C_2C_9\\+S_2C_9} & \substack{-C_2C_9\\-S_2C_9} & \substack{-C_2C_5\\+S_2S_5} & \substack{-C_2C_5\\-C_2S_5} & -C_2 \\
C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1
\end{array}\right]$$

그림 2. 계승화를 위한 DST 2, N=16
Fig. 2. Discrete sine transform 2 prepared for factoring, N=16.

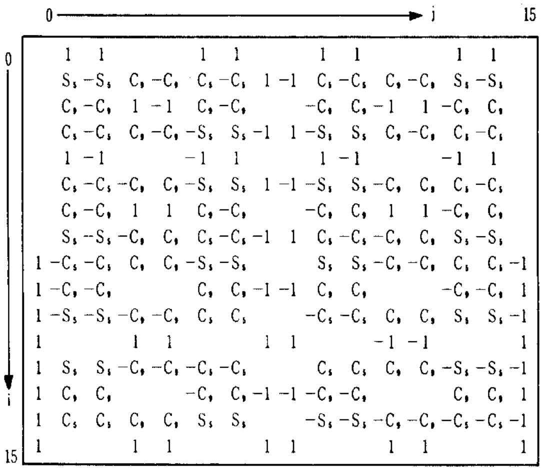

그림 4. $[R]_{16}$ 메트릭스 : $[S]_{16}$와 $[T]_{16}$ 메트릭스의 곱

Fig. 4. Matrix $[R]_{16}$: product of matrix $[S]_{16}$ and matrix $[T]_{16}$

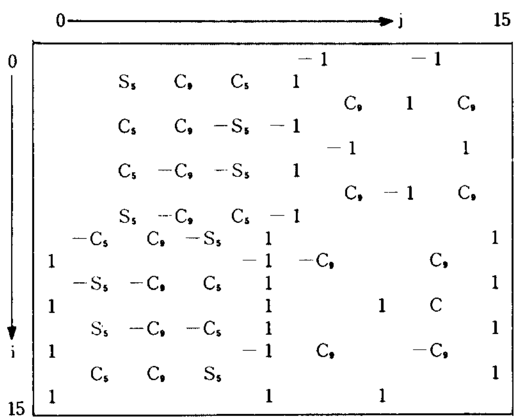

그림 5. $[S]_{16}$ 메트릭스 : $[B]_{16}$과 $[C]_{16}$ 메트릭스의 곱

Fig. 5. Matrix $[S]_{16}$: product of matrix $[B]_{16}$ and matrix $[C]_{16}$.

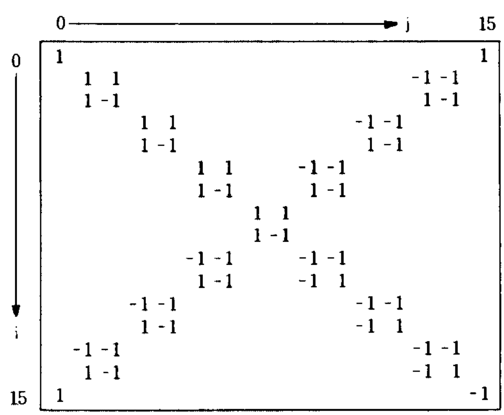

그림 6. $[T]_{16}$ 메트릭스 : $[D]_{16}$과 $[E]_{16}$ 메트릭스의 곱

Fig. 6. Matrix $[T]_{16}$: porduct of matrix $[D]_{16}$ and matrix $[E]_{16}$.

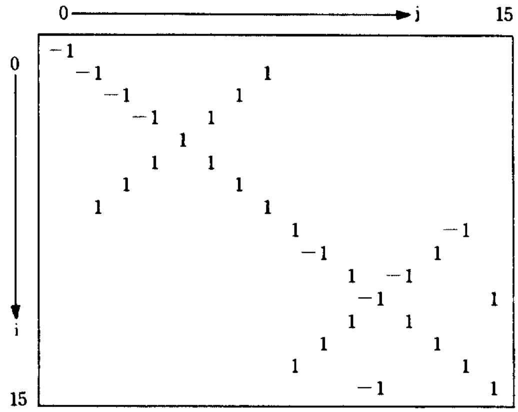

그림 7. DST 2 의 $[B]_{16}$ 메트릭스

Fig. 7. Matrix $[B]_{16}$ of discrete sine transform 2.

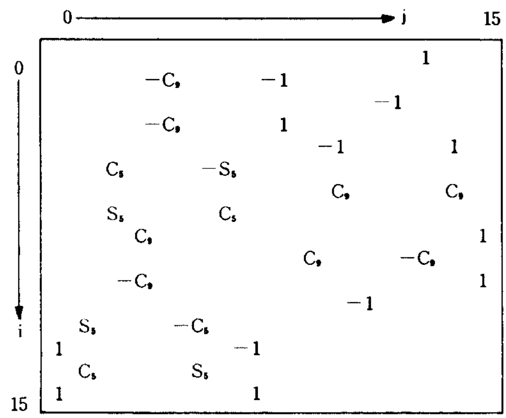

그림 8. DST 2 의 $[C]_{16}$ 메트릭스

Fig. 8. Matrix $[C]_{16}$ of discrete sine transform 2.

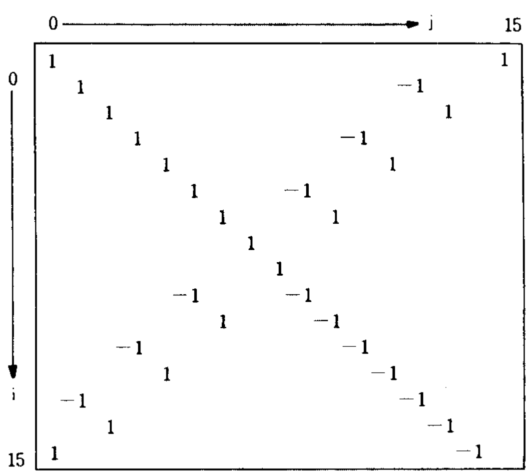

그림 9. DST 2 의 $[D]_{16}$ 메트릭스

Fig. 9. Matrix $[D]_{16}$ of discrete sine transform 2.

$$
\begin{array}{c}
0 \longrightarrow n \qquad\qquad\qquad\qquad\qquad 15 \\
\begin{array}{c} 0 \\ \downarrow \\ k \\ 15 \end{array}
\begin{bmatrix}
1 &&&&&&&&&&&&&&& \\
& 1 & 1 &&&&&&&&&&&&& \\
& 1 & -1 &&&&&&&&&&&&& \\
&&& 1 & 1 &&&&&&&&&&& \\
&&& 1 & -1 &&&&&&&&&&& \\
&&&&& 1 & 1 &&&&&&&&& \\
&&&&& 1 & -1 &&&&&&&&& \\
&&&&&&& 1 & 1 &&&&&&& \\
&&&&&&& 1 & -1 &&&&&&& \\
&&&&&&&&& 1 & 1 &&&&& \\
&&&&&&&&& 1 & -1 &&&&& \\
&&&&&&&&&&& 1 & 1 &&& \\
&&&&&&&&&&& 1 & -1 &&& \\
&&&&&&&&&&&&& 1 & 1 & \\
&&&&&&&&&&&&& 1 & -1 & \\
&&&&&&&&&&&&&&& 1
\end{bmatrix}
\end{array}
$$

그림10. DTS 2 의 $[E]_{16}$ 메트릭스

Fig. 10. Matrix$[E]_{16}$ of discrete sine transform 2.

6 은 각 행열에 2 개 이상의 값 을 가지므로 이를 다시 분해시켜 각 행열에 적어도 2개의 값만을 갖도록 한다. 즉 그림 5 는 그림 7 ($[B]_{16}$ 메트릭스)과 8 ($[C]_{16}$ 메트릭스)의 메트릭스로, 그림 6 은 그림 9 ($[D]_{16}$ 메트릭스)와 10 ($[E]_{16}$ 메트릭스)의 메트릭스로 다시 바꾼다.

결국, 그림 1 의 변형인 그림 2 메트릭스는 $[A]_{16}[B]_{16}[C]_{16}[D]_{16}[E]_{16}$로 구성되며 각 메트릭스는 어느 행열에서도 둘 이상의 값을 가지지 않는다. 이제까지 고찰한 N=16인 경우는 N이 8 인 경우[15]와 똑같은 방법으로 전개한 것이다. 이제 N이 8, 16인 경우를 정리하면서 더 큰 N에 대해서 확장할 수 있음을 고찰한다. N = 8 일때 변환코저하는 메트릭스 $[\phi_{ij}]_8$은 $[A]_8$의 인수를 가지고 있다. 즉, $[\phi_{ij}]_8=[A]_8\cdot[R]_8$

여기서,

$$
[A]_8=\begin{bmatrix}
& -C_2 &&&&& S_2 & \\
&& -C_3 &&& S_3 && \\
&&& -C_4 & S_4 &&& \\
-C_5 &&&&&&& C_5 \\
&&& S_4 & C_4 &&& \\
&& S_3 &&& C_3 && \\
& S_2 &&&&& C_2 & \\
C_1 &&&&&&& C_1
\end{bmatrix} \tag{12}
$$

변환 메트릭스 $[\phi_{ij}]_8$과 규칙적 배열을 갖는 $[A]_8$을 알면 $[R]_8$를 구할 수 있으며, $[R]_8$은 다시 $[C]_8$과 규칙적인 배열을 갖는 $[T]_8$로 나눌 수 있다.

$[R]_8=[C]_8\cdot[T]_8$

$$
[R]_8=\begin{bmatrix}
& 1 & 1 &&& 1 & 1 & \\
& C_5 & -C_5 & 1 & -1 & C_5 & -C_5 & \\
& 1 & -1 &&& -1 & 1 & \\
& C_5 & -C_5 & -1 & 1 & C_5 & -C_5 & \\
1 & -C_5 & -C_5 &&& C_5 & C_5 & -1 \\
1 &&& -1 & -1 &&& 1 \\
1 & C_5 & C_5 &&& -C_5 & -C_5 & -1 \\
1 &&& 1 & 1 &&& 1
\end{bmatrix} \tag{13}
$$

$$
[C]_8=\begin{bmatrix}
&&&&& -1 && \\
&& C_5 && 1 &&& \\
&&&&&& 1 & \\
&& C_5 && -1 &&& \\
& -C_5 &&&&&& 1 \\
1 &&& -1 &&&& \\
& C_5 &&&&&& 1 \\
1 &&& 1 &&&&
\end{bmatrix} \tag{14}
$$

$$
[T]_8=\begin{bmatrix}
1 &&&&&&& 1 \\
& 1 & 1 &&& -1 & -1 & \\
& 1 & -1 &&& 1 & -1 & \\
&&& 1 & 1 &&& \\
&&& 1 & -1 &&& \\
& -1 & -1 &&& -1 & -1 & \\
& -1 & -1 &&& -1 & 1 & \\
1 &&&&&&& -1
\end{bmatrix} \tag{15}
$$

$[R]_8$과 $[T]_8$을 알면 $[C]_8$을 구할 수 있게 된다. N=16인 경우에도 $[\phi_{ij}]_{16}$은 $[A]_{16}$과 $[R]_{16}$의 곱으로 되며 $[A]_{16}$은 $[A]_8$에서와 같이 규칙적인 배열을 가지므로 $[\phi_{ij}]_{16}$와 $[A]_{16}$으로 부터 $[R]_{16}$을 구한다.

$$[\phi_{ij}]_{16}=[A]_{16}\cdot[R]_{16} \tag{16}$$

여기서 $[A]_{16}$와 $[R]_{16}$은 그림3, 4에 나타나 있다. $[R]_{16}$은 다시 규칙적인 배열을 갖는 $[T]_{16}$을 포함하고 있다.

즉, $[R]_{16}=[S]_{16}\cdot[T]_{16}$ (17)

$[T]_{16}$은 그림 6 에 나타나 있으며, $[T]_8$과 비교한다면 $[T]_{32}$를 어렵지 않게 얻을 수 있을 것이다. 구해지는 $[S]_{16}$ (그림 5) 은 다시 규칙적 배열을 갖는 $[B]_{16}$ (그림 7) 을 포함하고 있으므로 $[C]_{16}$ (그림 8) 을 또한 구할 수 있게 된다.

$$[S]_{16}=[B]_{16}\cdot[C]_{16} \tag{18}$$

$[T]_{16}$과 같은 형태의 메트릭스는 항상 그림 9, 10과 같은 형태의 메트릭스로 쓸 수 있다. 이렇게 볼때, 고속 알고리즘을 구현을 위해서 N = 8 인 경우에는 4개의 메트릭스, N=16인 경우에는 5 개의 메트릭스로 구성됨을 알 수 있다. N=32의 경우에도 같은 방법으로 구할 수 있었으며 이때에는 6 개의 메트릭스로 구성된다.

일반적인 전개를 위해 기본 메트릭스$[A]_N$과 규칙적인 배열을 갖는 $[B]_N$ 및 $[T]_N$등은 다음 식(20)(21)(22)와 같은 형태의 메트릭스가 된다.

여기서 $[A]_N$은 맨처음 인수 메트릭스로 나타나며 인수 메트릭스들의 배열에서 처음에 위치한다. $[T]_N$ 메트릭스는 $[\phi_{ij}]_N$속에서 $[A]_N$ 인수를 제하고 남는 $[R]_N$ 메트릭스 속의 인수 메트릭스로 존재하나 배열순서로는 마지막 번째에 위치한다.

이 $[T]_N$메트릭스는 반드시 두인수를 가지므로 두개

의 메트릭스인 식(23)(24)로 나눠 표시한다. 또 N값이 커감에 따라서 $[B]_N$, $[F]_N$ 등과 같은 인수 메트릭스가 추가된다. 이러한 진행을 계속하여 남는 메트릭스가 어느 행열에도 2개 이상의 값을 갖지 않을때까지 반복한다.

$[\phi_{ij}]$메트릭스 속에 들어있는 인수 메트릭스의 수 M은 샘플 데이타 수 N가 주어지면 다음 식(19)으로 부터 구할 수 있다.

$$M=\log_2 2N \tag{19}$$

$$[A]_N=\begin{bmatrix}
-C_2 & & & & & & & S_2 & \\
 & -C_3 & & & & & S_3 & & \\
 & & \ddots & & & \iddots & & & \\
 & & -C_{\frac{N}{2}-1} & & & S_{\frac{N}{2}-1} & & & \\
 & & & -C_{\frac{N}{2}} & S_{\frac{N}{2}} & & & & \\
-C_{\frac{N}{2}+1} & & & & & & & & C_{\frac{N}{2}+1} \\
 & & & S_{\frac{N}{2}} & C_{\frac{N}{2}} & & & & \\
 & & S_{\frac{N}{2}-1} & & & C_{\frac{N}{2}-1} & & & \\
 & \iddots & & & & & \ddots & & \\
 & S_2 & & & & & & C_2 & \\
C_1 & & & & & & & & C_1
\end{bmatrix} \tag{20}$$

$$[B]_N=\begin{bmatrix}
-1 & & & & & & & & & & \\
 & -1 & & & 1 & & & & & & \\
 & & \ddots & \iddots & & & & & & & \\
 & & -1 & 1 & & & & & & & \\
 & & & 1 & & & & & & & \\
 & & 1 & & 1 & & & & & & \\
 & \iddots & & & & \ddots & & & & & \\
1 & & & & & & 1 & & & & \\
 & & & & & & & 1 & & -1 & \\
 & & & & & & & \ddots & & 1 & \\
 & & & & & & & -1 & \iddots & & \\
 & & & & & & & 1 & -1 & & \\
 & & & & & & & -1 & & & 1 \\
 & & & & & & 1 & & 1 & & \\
 & & & & & \iddots & & & & \ddots & \\
 & & & & & 1 & & -1 & & 1 & \\
 & & & & & & & & & & 1
\end{bmatrix} \tag{21}$$

$$[T]_N=\begin{bmatrix}
1 & & & & & & & & & & & 1 \\
 & 1 & 1 & & & & & & & -1 & -1 & \\
 & 1 & -1 & & & & & & & 1 & -1 & \\
 & & \ddots & 1 & 1 & & & -1 & -1 & \iddots & & \\
 & & & 1 & -1 & & & 1 & -1 & & & \\
 & & & & & 1 & 1 & & & & & \\
 & & & & & 1 & -1 & & & & & \\
 & & & -1 & -1 & & & -1 & -1 & & & \\
 & & & 1 & -1 & & & -1 & 1 & & & \\
 & -1 & -1 & \iddots & & & & & \ddots & -1 & -1 & \\
 & 1 & -1 & & & & & & & -1 & 1 & \\
1 & & & & & & & & & & & -1
\end{bmatrix} \tag{22}$$

$$[D]_N=\begin{bmatrix}
1 & & & & & & & & & & & 1 \\
 & 1 & & & & & & & & -1 & & \\
 & & 1 & & & & & & & & 1 & \\
 & & & \ddots & & & & & \iddots & & & \\
 & & & & 1 & & & -1 & & & & \\
 & & & & & 1 & & & 1 & & & \\
 & & & & & & 1 & & & & & \\
 & & & & & & & 1 & & & & \\
 & & & & -1 & & & -1 & & & & \\
 & & & \iddots & & & & & \ddots & & & \\
 & -1 & & & & & & & & -1 & & \\
 & & 1 & & & & & & & & -1 & \\
1 & & & & & & & & & & & -1
\end{bmatrix} \tag{23}$$

$$[E]_N=\begin{bmatrix}
1 & & & & & & & & & & \\
 & 1 & 1 & & & & & & & & \\
 & 1 & -1 & \ddots & & & & & & & \\
 & & & 1 & 1 & & & & & & \\
 & & & 1 & -1 & & & & & & \\
 & & & & & 1 & 1 & & & & \\
 & & & & & 1 & -1 & & & & \\
 & & & & & & & 1 & 1 & & \\
 & & & & & & & 1 & -1 & \ddots & \\
 & & & & & & & & & 1 & 1 \\
 & & & & & & & & & 1 & -1 \\
 & & & & & & & & & & & 1
\end{bmatrix} \tag{24}$$

Ⅲ. 신호흐름선도와 비교 고찰

DST 2 의 高速 計算을 하기 위해 Ⅱ장에서 유도한 인

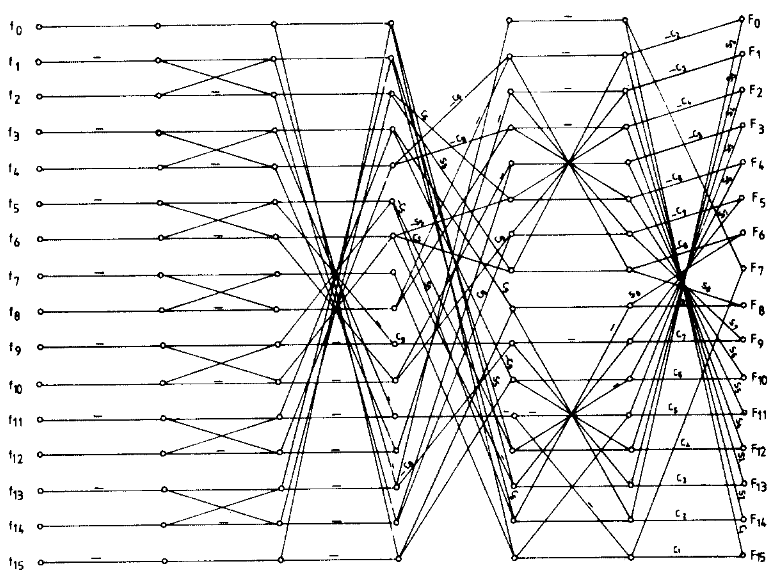

그림11. DST 2 를 위한 고속 신호흐름선도, N=16

Fig. 11 Fast signal flow graph for forward discrete sine transform 2, N=16.

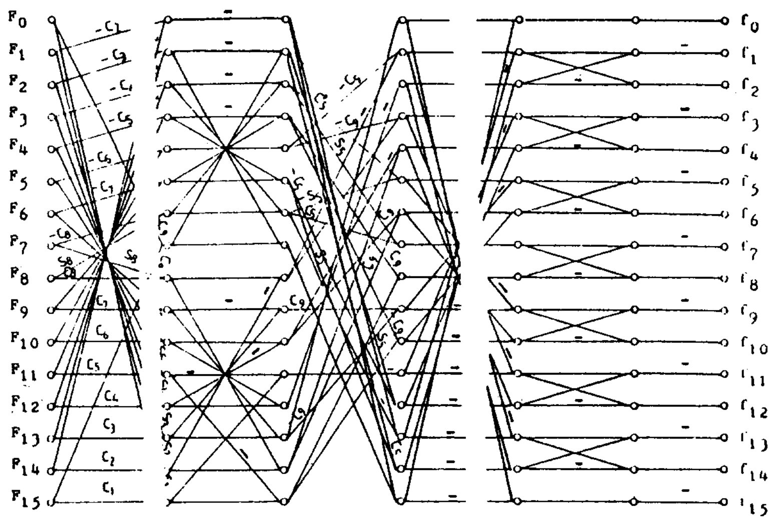

그림12. DST 2 역변환을 위한 고속신호흐름선도, N=16
Fig. 12. Fast signal flow graph for inverse discrete sine transform 2, N=16.

수 메트릭스들로부터 신호흐름선도를 그림11, 12에 나타내었다.

순방향 DST 2 를 위한 고속 신호흐름선도를 나타낸 그림11의 첫단에 우수번째마다 표시된 음의 부호는 우수열 부호교환이 됨을 뜻하며 DST 2 역변환을 위한 고속 신호흐름선도를 나타낸 그림12의 마지막 단에서는 이를 다시 환원시키는 음의 부호가 표시되어 있다.

표 2 는 여러 고속변환 알고리즘들에 대한 연산수들을 비교한 것이다. N=16일때 DST 2 를 직접 연산하면 덧셈수와 곱셈수가 240개와 256개이지만 그림11을 통한 연산은 덧셈 및 곱셈수가 71개와 48개로 된다.

표 2. 여러 변환 알고리즘에 대한 덧셈과 곱셈수
Table 2. Numbers of adders and multipliers to various algorithm for the transform.

N / Transform	8 (7)*		16 (15)*		32 (31)*	
	덧셈수	곱셈수	덧셈수	곱셈수	덧셈수	곱셈수
DCT	56	64	240	256	992	1024
FDCT by Chen & Fralick	26	16	74	44	194	116
CMT for FDCT by Kwak & Rao	34	16	106	56	170	136
DST 2	56	64	240	256	992	1024
FDST 2 by Wang	26	16	74	44	194	116
본논문	26	20	71	48	182	140
DST 1	(42)	(49)	(210)	(225)	(930)	(961)
FDST1 by Yip & Rao	(22)	(8)	(62)	(30)	(166)	(54)

*표는 DST1의 샘플 데이타 수를 표시한 것이다.

이것은 Wang의 방법과 비교할때 곱셈수는 4 개가 많고 덧셈수는 3 개가 적게 나타났다. 또 N=32경우 Wang의 방법보다 덧셈수는 12개 적지만 곱셈수는 많게 나타났다. 그러나 본 연구방법의 알고리즘을 통한 계산 결과는 정의된 DST2와 일치하지만 Wang의 방법을 통한 계산결과는 N=16에서 24개의 오차값을 가지며, N=32에서는 무려 240개의 값들이 오차를 갖는다.

이러한 관계는 DCT의 고속 계산 알고리즘에서 Chen의 FDCT와 Kwak의 CMT에서도 찾아 볼 수 있다.

N=16일때, 정의된 DST 2 의 값들과 Wang의 알고

표 3. 정의된 DST 2 의 값과 Wang의 방법과 다른 값들(N=16)
Table 3. Different values between Wang's method and DST 2 defined(N=16).

$\phi_{i,j}$ 항	정의식과 본 연구방법	Wang의 방법
$\phi_{0,5}$ 와 $\phi_{0,10}$	0.47139	0.09055
$\phi_{0,2}$ 와 $\phi_{0,13}$	0.8819	0.50108
$\phi_{0,1}$ 와 $\phi_{0,14}$	0.9509	0.41835
$\phi_{3,5}$ 와 $\phi_{3,10}$	0.29028	0.58610
$\phi_{3,2}$ 와 $\phi_{3,13}$	0.95069	0.66112
$\phi_{3,1}$ 와 $\phi_{3,14}$	0.47139	0.53046
$\phi_{4,5}$ 외 $\phi_{4,10}$	0.9569	0.77141
$\phi_{4,2}$ 와 $\phi_{4,13}$	0.29028	0.04751
$\phi_{4,1}$ 와 $\phi_{4,14}$	0.8819	0.58859
$\phi_{7,5}$ 외 $\phi_{7,10}$	0.8819	0.9194
$\phi_{7,2}$ 와 $\phi_{7,13}$	0.47139	0.43388
$\phi_{7,1}$ 와 $\phi_{7,14}$	0.2902	0.34331

리즘을 통한 계산결과가 서로 다른 값들을 표3에 나타내었다.

Wang의 방법은 주어지는 N값에 따라 쉽게 알고리즘을 구현할 수 있다는 장점은 있으나 N값이 커감에 따라 오차를 갖는 값들 수도 계속 커가지는 단점을 가지고 있다.

Ⅳ. 結 論

이미지처리등에 사용되는 변환중에서 DST 2의 고속계산 알고리즘을 변환 메트릭스속에 규칙적인 배열을 가진 인수메트릭스들로 부터 계승화시켜 나감으로서 얻을 수 있음을 보였다. 이때 변환메트릭스의 샘플데이타 수가 N인 경우, 인수메트릭스 수는 $\log_2 2N$개가 된다.

N이 8, 16, 32인 例에서 Wang의 방법과 비교할 때 덧셈수는 적으나 곱셈수는 약간 많게 나타났지만 N의 증가에 따라 오차가 커지는 Wang의 방법(N=16에서 24개, N=32에서 240개)보다 본 연구의 방법은 정의된 DST 2를 정확히 구현할 수 있음을 확인하였다. 끝으로 DST2 변환과 역변환에 대한 신호흐름선도를 제시하였다.

參 考 文 献

[1] P.A. Wintz, "Transform picture coding," *Proc. IEEE,* vol.60, pp.809-820, July, 1972.

[2] N. Ahmed and K.R. Rao, *Orthogonal Transform for Digital Signal Processing.* New York: Springer-Verlag, 1975.

[3] J.W. Cooley, P.A. Lewis, and P.D. Welch, "Historical notes on the fast Fourier transform," *Proc. IEEE,* vol.55, pp. 1675-1677, Oct., 1967.

[4] H.C. Andrews, "A high speed algorithm for the computer generation of Fourier transforms," *IEEE Trans. Computers,* vol. C-17, pp. 373-375, April 1968.

[5] D.F. Elliot and K.R. Rao, *Fast Transform: Algorithms, Analysis and Applications.* New York, NY: Academic Press, 1982.

[6] W. K. Pratt, "Generalized Wiener filtering computation techniques," *IEEE Trans.* Computers, C-21, pp.636-641, 1972.

[7] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete cosine transform," IEEE *Trans. Computer,* vol. C-23, pp.90-93, Jan., 1974.

[8] A.K. Jain, "A fast Karhunen-Loeve transform for a class of stochastic processes," *IEEE Trans. Commun.,* vol-24, pp.1023-1029, 1976.

[9] H.B. Kekre and J.K. Solanki, "Comparative performance of various trigonometric unitary transforms for transform image coding," *Int. J. Electron,* vol. 44, pp.305-315, 1978.

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

[11] P. Yip and K.R. Rao, "A fast computational algorithm for the discrete sine transform," *IEEE Trans. Commun.,* vol. COM-28, pp. 304-307, Feb., 1980.

[12] Z. D. Wang, "A fast algorithm for the discrete sine transform implemented by the fast cosine transform," *IEEE Trans. Acoustics Speech, and Signal Processing,* vol. ASSP-30, pp.814-815, Oct., 1982.

[13] P. Yip and K.R. Rao, "On the computation and the effectiveness of the discrete sine transform," *IEEE Trans. Comput. Electron.,* vol. 7, pp.45-55, 1980.

[14] H.S. Kwak, R. Srinivasan, and K.R. Rao, "C-Matrix transform," *IEEE Trans. ASSP.,* vol. ASSP-31, pp.1304-1307, Oct., 1983.

[15] 곽훈성, 신건순, "이산 정현 변환을 위한 고속계산 알고리즘," 대한전자공학회 하계종합학술대회 논문집, vol.7, no.1, pp.307-309, 1984.