SIMPLE ELEMENT INVERSE DCT/DFT HYBRID ARCHITECTURE ALGORITHM

Jia Hou[1,2]*, Moon Ho Lee*[1]*, Dae Chul Park*[3]*, and Kwang Jae Lee*[4]

[1]Institute of Information & Communication, Chonbuk National University, Chonju, 561-756, Korea.
Email: {jiahou, moonho}@chonbuk.ac.kr
[2]School of Electronics & Information, Soochow University, Suzhou, China.
[3]Dept. of Information & Communication Engineering, Hannam University, Daejon, 306-741, Korea.
Email: fia4joy@yahoo.co.kr
[4]Dept. of Infor. & Telecom. Engineering, Hanlyo University, Korea.
Email: kjlee@hanlyo.ac.kr

ABSTRACT

We address a 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.

1. INTRODUCTION

Discrete Cosine Transform (DCT) has found applications in signal classification and representation [1,2,3]. The DCT-II is a popular structure and it is usually accepted as the best suboptimal transformation that its performance is very close to that of the statistically optimal Karhunen-Loeve transform [3,4,5]. Furthermore, the discrete Fourier transform (DFT) is also a popular transformation for signal processing and communication [6,7,8]. To analyze these two different transforms, we now focus on the sparse matrix factorization of their transfer matrices.

Otherwise, the analysis and decomposition of the sparse matrix wad demonstrated as a useful tool to develop the fast computations and character generalization [9,10,11]. Therefore, similar to the method in [9-12], the DCT-II and DFT matrices can be decomposed to one orthogonal character matrix and a special sparse matrix. In this form, the inverse of the sparse matrix is from block-wise inverse or element-wise inverse. Hence, the proposed method is named element inverse sparse matrix decomposition [10,11]. In this paper, we focus on the architecture of the sparse matrix decomposition and propose a hybrid architecture to joint the DCT and DFT together.

2. ELEMENT INVERSE SPARSE MATRIX DECOMPOSITION FOR DCT-II MATRIX

Similar to the definition of Jacket matrix [13,14], the inverse of a N-by-N sparse matrix is only from the element-wise inverse or block-wise inverse, we name it as element inverse sparse matrix.

A typical DCT matrix is the DCT-II case, which is defined by

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

where $k_j = \begin{cases} 1, & j = 1,2,\ldots,N-1 \\ \sqrt{\frac{1}{2}}, & j = 0, N \end{cases}$. In this paper, we will focus on the DCT-II matrix and introduce a simple matrix factorization algorithm. First, the 2-by-2 DCT-II matrix is given by

$$[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}}, \qquad (2)$$

where $1/\sqrt{2}$ can be considered as a special element inverse sparse matrix of order-1, its inverse if $\sqrt{2}$, and $C_l^i = \cos(i\pi / l)$ is the cosine unit for DCT computations. Next, the 4-by-4 DCT-II matrix is formed by

$$[C]_4 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix}. \qquad (3)$$

The row permutation matrix $[\Pr]_N$ is defined by

$$[\Pr]_2 = [I]_2 \text{ and } [\Pr]_N = \lfloor pr_{i,j} \rfloor_N, \; N \geq 4, \qquad (4)$$

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

and $i, j \in \{0,1,..., N-1\}$. Further, we define a reversible permutation matrix $[Pc]_N$ as follows.

$$[Pc]_2 = [I]_2, \text{ and } [Pc]_N = \begin{bmatrix} I_{\frac{N}{4}} & 0 & 0 & 0 \\ 0 & I_{\frac{N}{4}} & 0 & 0 \\ 0 & 0 & 0 & I_{\frac{N}{4}} \\ 0 & 0 & I_{\frac{N}{4}} & 0 \end{bmatrix}, N \ge 4. \quad (5)$$

Thus we can write

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

where $[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$, the equation (2), $()^T$ denotes the transpose of a matrix and $[B]_2 = \begin{bmatrix} C_8^1 & C_8^3 \\ C_8^3 & -C_8^1 \end{bmatrix}$.

Clearly, we have the block-wise inverse sparse matrix as

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

Generally, the permuted DCT-II matrix $[\widetilde{C}]_N$ can be recursively formed by using

$$[\widetilde{C}]_N = [\Pr]_N [C]_N [Pc]_N = \left(\begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} C_{\frac{N}{2}} & 0 \\ 0 & B_{\frac{N}{2}} \end{bmatrix} \right)^T, \quad (8)$$

where $[B]_{N/2}$ can be calculated by

$$[B]_{\frac{N}{2}} = [(C_{2N}^{f(m,n)})_{m,n}]_{\frac{N}{2}}, \begin{cases} f(m,1) = 2m-1, \\ f(m,n+1) = f(m,n) + 2f(m,1), \end{cases} \quad (9)$$

where $m, n \in \{1,2,..., N/2\}$. The inverse form of (8) can be simply computed by

$$([\widetilde{C}]_N)^{-1} = \frac{2}{N} \left(\begin{bmatrix} \left(C_{\frac{N}{2}}\right)^{-1} & 0 \\ 0 & \left(B_{\frac{N}{2}}\right)^{-1} \end{bmatrix} \begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix} \right)^T. \quad (10)$$

Furthermore, the submatrix $[B]_N$ can be represented by

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

where

$[K]_N = \begin{bmatrix} \sqrt{2} & 0 & ... & 0 \\ -\sqrt{2} & 2 & 0 & ... \\ \sqrt{2} & -2 & 2 & ... \\ ... & ... & ... & 2 \end{bmatrix}$, $[D]_N = diag\ [C_{4n}^{\Phi_0},..., C_{4n}^{\Phi_{N-1}}]$, and

$\Phi_i = 2i+1$, $i \in \{0,1,...,N-1\}$.

Proof of (11): The N-by-N DCT-II matrix has the form

$$[C]_N = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & ... & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & ... & C_{4N}^{2k_0\Phi_{N-1}} \\ ... & ... & & ... \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & ... & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix}, \quad (12)$$

where $k_i = i+1$, $i \in \{0,1,2,...\}$. According to (9), the matrix $[B]_N$ from $[C]_{2N}$ can be represented by

$$[B]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & ... & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & ... & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ ... & ... & & ... \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & ... & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix}. \quad (13)$$

Obviously, we have

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

and

$$C_{4N}^{\Phi_m} - 2C_{4N}^{2k_{i-1}\Phi_m} C_{4N}^{\Phi_m} + 2C_{4N}^{2k_i\Phi_m} C_{4N}^{\Phi_m} = C_{4N}^{(2k_i+1)\Phi_m}. \quad (15)$$

By taking (14) and (15) into (11), we have

$$[K]_N [C]_N [D]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & ... & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & ... & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & ... & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ ... & ... & ... & ... \end{bmatrix}. \quad (16)$$

The proof is completed.

Thus the DCT-II matrix can be written by

$$[\widetilde{C}]_N = \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & K_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} C_{\frac{N}{2}} & 0 \\ 0 & C_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & D_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix}. \quad (17)$$

And the general recursive form is given by

$$\begin{aligned} [C]_N = ([\Pr]_N)^{-1} \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & K_{\frac{N}{2}} \end{bmatrix} ... \left[I_{\frac{N}{4}} \otimes ([\Pr]_4)^{-1} \right] \left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \right] \left[I_{\frac{N}{2}} \otimes C_2 \right] \\ \left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix} \right] \left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix} \right] \left[I_{\frac{N}{4}} \otimes ([Pc]_4)^{-1} \right] \\ ... \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & D_{\frac{N}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix} ([Pc]_N)^{-1}. \end{aligned} \quad (18)$$

To simplify (18), we can rewrite it by using

$$[C]_N = ([\widetilde{\Pr}]_N)^{-1} \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & K_{\frac{N}{2}} \end{bmatrix} ... \left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & K_2 \end{bmatrix} \right] \left[I_{\frac{N}{2}} \otimes C_2 \right]$$

$$\left[I_{\frac{N}{4}}\otimes\begin{bmatrix}I_2 & 0\\ 0 & D_2\end{bmatrix}\right]\left[I_{\frac{N}{4}}\otimes\begin{bmatrix}I_2 & I_2\\ I_2 & -I_2\end{bmatrix}\right]\cdots\begin{bmatrix}I_{\frac{N}{2}} & 0\\ 0 & D_{\frac{N}{2}}\end{bmatrix}\begin{bmatrix}I_{\frac{N}{2}} & I_{\frac{N}{2}}\\ I_{\frac{N}{2}} & -I_{\frac{N}{2}}\end{bmatrix}\left([\widetilde{Pc}]_N\right)^{-1}. \qquad (19)$$

The butterfly data flow diagram corresponding to (19) is shown as in Fig.1.

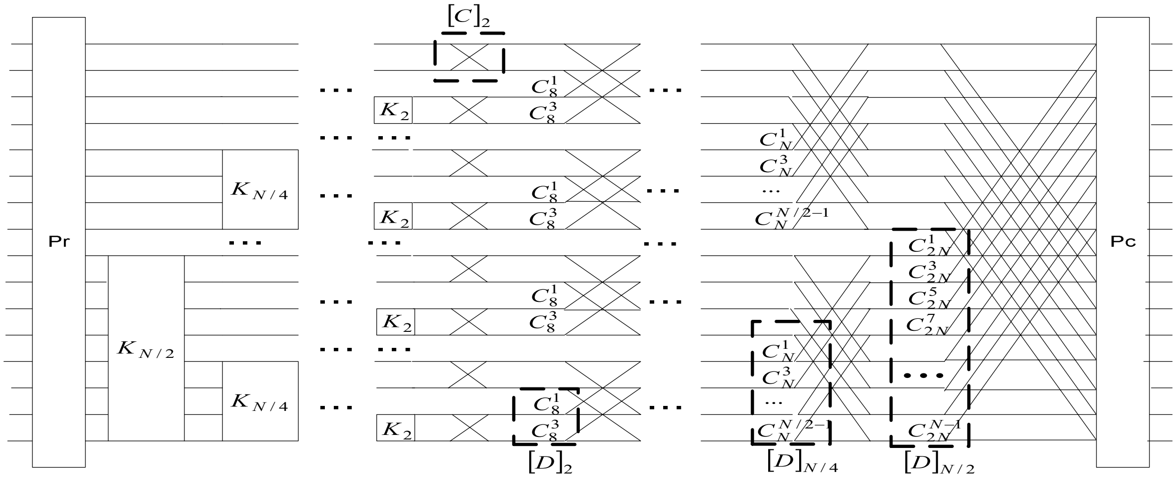

Fig.1 Butterfly data flow diagram of the proposed computation of the N-by-N DCT-II matrix.

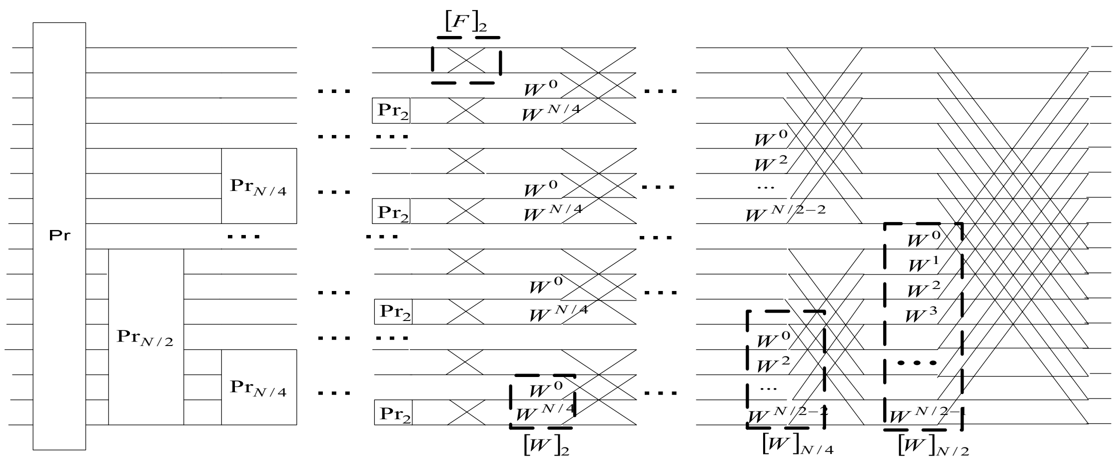

Fig.2 Butterfly data flow diagram of the proposed computation of the N-by-N DFT matrix.

3. ELEMENT INVERSE SPARSE MATRIX DECOMPOSITION FOR DFT MATRIX

The DFT is a Fourier representation of a given sequence $x(m)$, $0 \le m \le N-1$ and it is defined by

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

where $W = e^{-j\frac{2\pi}{N}}$, $j=\sqrt{-1}$. The N-point DFT matrix can be denoted by $[F]_N = [W^{nm}]_N$. Similar to the section 2, we can write a permuted 4-by-4 DFT matrix by using

$$[\widetilde{F}]_4 = [\mathrm{Pr}]_4 [F]_4 = \left(\begin{bmatrix}I_2 & I_2\\ I_2 & -I_2\end{bmatrix}\begin{bmatrix}F_2 & 0\\ 0 & E_2\end{bmatrix}\right)^T, \qquad (21)$$

where $[E]_2 = \begin{bmatrix}1 & -j\\ 1 & j\end{bmatrix}$, and its inverse form can be obtained by

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

Generally, the N-by-N permuted DFT matrix has

$$[\widetilde{F}]_N = [\mathrm{Pr}]_N [F]_N = \left(\begin{bmatrix}I_{\frac{N}{2}} & I_{\frac{N}{2}}\\ I_{\frac{N}{2}} & -I_{\frac{N}{2}}\end{bmatrix}\begin{bmatrix}\widetilde{F}_{\frac{N}{2}} & 0\\ 0 & E_{\frac{N}{2}}\end{bmatrix}\right)^T, \qquad (23)$$

where $[\widetilde{F}]_2 = [F]_2$, and the submatrix $[E]_N$ can be written by

$$[E]_N = [\mathrm{Pr}]_N [\widetilde{F}]_N [W]_N, \qquad (24)$$

where $[W]_N = diag\,[W^0, W^1, ..., W^{N-1}]$, and W is the complex unit for $2N$ -point DFT matrix. Similar to (17), we can rewrite the permuted DFT matrix by using

$$[\widetilde{F}]_N = \begin{bmatrix} \widetilde{F}_{\frac{N}{2}} & 0 \\ 0 & \Pr_{\frac{N}{2}}\widetilde{F}_{\frac{N}{2}}W_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix}$$

$$= \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & \Pr_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} \widetilde{F}_{\frac{N}{2}} & 0 \\ 0 & \widetilde{F}_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & W_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix}. \quad (25)$$

As a result, the general recursive form for DFT matrix can be represented by

$$[F]_N = ([\Pr]_N)^{-1}[\widetilde{F}]_N$$

$$= ([\Pr]_N)^{-1}\begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & \Pr_{\frac{N}{2}} \end{bmatrix}\cdots\left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & \Pr_2 \end{bmatrix}\right]\left[I_{\frac{N}{2}} \otimes F_2\right]$$

$$\left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & 0 \\ 0 & W_2 \end{bmatrix}\right]\left[I_{\frac{N}{4}} \otimes \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}\right]\cdots\begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & W_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix}. \quad (26)$$

Clearly, the form of (26) is the same as that of (19), where we only need to change K_l to $\Pr_l$ and D_l to W_l, with the parameters $l \in \{2,4,8,...,N/2\}$. The butterfly data flow diagram corresponding to (26) is shown as in Fig.2.

4. CONCLUSION

In this paper, we derive the recursive formulas for DCT-II and DFT matrices. The results show that the DCT-II and DFT matrices can be unified by using the same sparse matrix decomposition algorithm and recursive architecture within some characters changed.

As illustrated in Fig.1, and Fig.2, we find that the DFT computation can be from the computation of the DCT matrix by replacing the submatrix $[D]_N$ to $[W]_N$, and the permutation matrix $[\Pr]_N$ to $[K]_N$. As a result, a simple generalized block diagram for DCT/DFT hybrid architecture and its fast algorithm can be shown as in Fig.3. In this figure, we joint DCT and DFT computations into one chip or one kind of processing architecture, where we use one switching box to control the output data flow. This result is useful to develop the united chip for video coding and digital modulations.

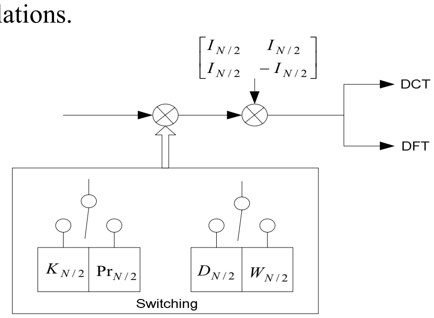

Fig.3. A simple DCT/DFT hybrid architecture.

ACKNOWLEDGEMENT

This work was partial supported by the MIC (Ministry of Information & Communication), under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by IITA, under ITRC supervised by IITA, and International Cooperation Research Program of the Ministry of Science & Technology, and KOTEF, Korea.

REFERENCES

[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] Moon Ho Lee, “Simple systolic arrays for discrete cosine transform,” *Multidimensional system and processing*, no.1, pp.389-398, 1990, Kluwer Academic Publishers.

[4] Moon Ho 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] Moon Ho 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] Daechul Park, Moon Ho Lee, and Euna Choi, “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] Moon Ho 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] Moon Ho 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.

[13] Moon Ho 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.

[14] Moon Ho Lee, B. S. Rajan, and Ju Yong Park, “A Generalized Reverse Jacket Transform,” *IEEE Trans. Circuits Syst. II*, vol.48, no. 7, pp.684-690, July, 2001.