

A Fast Hybrid Jacket-Hadamard Matrix Based Diagonal Block-wise Transform

Lee, M.H.; Khan, M.H.A.; Kim, K.J.; Park, D.

TR2014-002 January 2014

Abstract

In this paper, based on the block (element)-wise inverse Jacket matrix, a unified fast hybrid diagonal block-wise transform (FHDBT) algorithm is proposed. A new fast diagonal block matrix decomposition is made by the matrix product of successively lower order diagonal Jacket matrix and Hadamard matrix. Using a common lower order matrix in the form of $1 \ 1$, a fast recursive structure can be developed in the FHDBT, which is able to convert a newly developed discrete cosine transform (DCT)-II, discrete sine transform (DST)-II, discrete Fourier transform (DFT), and Haar-based wavelet transform (HWT). Since these DCT-II, DST-II, DFT, and HWT are widely used in different areas of applications, the proposed FHDBT can be applied to the heterogeneous system requiring several transforms simultaneously. Comparing with pre-existing DCT-II, DST-II, DFT, and HWT, it is shown that the proposed FHDBT exhibits less the complexity as its matrix size gets larger. The proposed algorithm is also well matched to circulant channel matrix. From the numerical experiments, it is shown that a better performance can be achieved by the use of DCT/DST-II compression scheme compared with the DCT-II only compression method.

Signal Processing: Image Communication

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



Contents lists available at ScienceDirect

Signal Processing: *Image Communication*journal homepage: www.elsevier.com/locate/imageA fast hybrid Jacket–Hadamard matrix based diagonal block-wise transform [☆]Moon Ho Lee ^a, Md. Hashem Ali Khan ^{a,*}, Kyeong Jin Kim ^{b,1}, Daechul Park ^c^a Division Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, South Korea^b Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA^c Department of Information and Communication Engineering, Hannam University, Daejeon 306-791, South Korea

ARTICLE INFO

Article history:

Received 23 April 2013

Received in revised form

12 November 2013

Accepted 12 November 2013

Available online 4 December 2013

Keywords:

Diagonal block (element)-wise inverse Jacket matrix (BIJM)

Sparse matrix decomposition

Successive lower order diagonal sparse matrix

Hadamard matrix

ABSTRACT

In this paper, based on the block (element)-wise inverse Jacket matrix, a unified fast hybrid diagonal block-wise transform (FHDBT) algorithm is proposed. A new fast diagonal block matrix decomposition is made by the matrix product of successively lower order diagonal Jacket matrix and Hadamard matrix. Using a common lower order matrix in the form of $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, a fast recursive structure can be developed in the FHDBT, which is able to convert a newly developed discrete cosine transform (DCT)-II, discrete sine transform (DST)-II, discrete Fourier transform (DFT), and Haar-based wavelet transform (HWT). Since these DCT-II, DST-II, DFT, and HWT are widely used in different areas of applications, the proposed FHDBT can be applied to the heterogeneous system requiring several transforms simultaneously. Comparing with pre-existing DCT-II, DST-II, DFT, and HWT, it is shown that the proposed FHDBT exhibits less the complexity as its matrix size gets larger. The proposed algorithm is also well matched to circulant channel matrix. From the numerical experiments, it is shown that a better performance can be achieved by the use of DCT/DST-II compression scheme compared with the DCT-II only compression method.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The last decade based on orthogonal transform has been seen a quiet revolution in digital video technology such as Moving Picture Experts Group (MPEG)-4, H.264, and high efficiency video coding (HEVC) [1–7]. Digital video is everywhere such as DVD, gaming players, computers and mobile handsets. Nowadays, many of the coexisting heterogeneous systems [7,8] are likely to catch the latest news on the web as on the smart TV and iPhone. Video compression is essential to all these applications. The discrete cosine transform (DCT)-II is popular compression structures for MPEG-4, H.264, and HEVC, and is accepted as the best suboptimal transformation since its performance is very close to that of the statistically optimal Karhunen–Loeve transform (KLT) [1–5]. For practical consideration, the underlying H.264-advanced video coding (AVC) intra mode dictates the transform coding implementation within a block coder with a typical block of size up to 16×16 . However, since a DCT-based block coder suffers from blocking effect, i.e., a disturbing discontinuity at the block boundaries,

[☆] This work was supported by the MEST 2012-002521, NRF, Korea.

* Corresponding author: Tel.: +82 632702463; fax: +82 632704166.

E-mail addresses: moonho@jbnu.ac.kr (M. H. Lee), hashem05ali@jbnu.ac.kr (M. H. A. Khan), kyeong.j.kim@hotmail.com (K. J. Kim), fia4joy@yahoo.co.kr (D. Park).¹ This work was done when he was with Inha University, Incheon, Korea.

Table 1

The comparison of computation complexity of conventional independent the DCT-II, DST-II, DFT, Haar transform and hybrid DCT-II/DST-II/DFT/HWT.

References number	Conventional		Proposed	
	Addition	Multiplication	Addition	Multiplication
W. H. Chen et al [18] DCT-II	$3N/2(\log_2 N - 1) + 2$	$N\log_2 N - (3N/2) + 4$	$N\log_2 N$	$\frac{N}{2}(\log_2 N + 1)$
Z. Wang [13] DST-II	$N(\frac{7}{4}\log_2(N) - 2) + 3$	$N(\frac{3}{4}\log_2(N) - 1) + 3$	$N\log_2 N$	$\frac{N}{2}(\log_2 N + 1)$
Cooley & Tukey [21] DFT	$N\log_2 N$	$(N/2)\log_2 N$	$N\log_2 N$	$(N/2)\log_2 N$
Andrews & Caspari [22] HWT	$2(N-1)$	N	$\sum_{i=1}^{h-1} \frac{N}{2^i}, h = \log_2 N$	$\sum_{i=1}^{h-2} \frac{N}{2^i}, h = \log_2 N^*$

* Addition count = $N/2^{n-1} + N/2^{n-2} + \dots + N/2 = \sum_{i=1}^{n-1} N/2^i$
 Multiplication count = $N/2^{n-2} + N/2^{n-2} + \dots + N/2 = \sum_{i=1}^{n-2} N/2^i$.

Table 2

Computational complexity: DCT-II/DST-II/DFT/HWT.

Matrix size	Conventional				Proposed			
	DCT-II	DST-II	DFT	HWT	DCT-II	DST-II	DFT	HWT
Addition								
$N=4$	8	9	8	6	8	8	8	2
$N=8$	26	29	4	14	24	24	24	6
$N=16$	74	83	64	30	64	64	64	14
$N=32$	194	219	160	62	160	160	160	30
$N=64$	482	547	384	126	384	384	384	62
$N=128$	1154	1315	896	254	896	896	896	126
$N=256$	2690	3075	2048	510	2048	2048	2048	254
Multiplication								
$N=4$	6	5	4		6	6		×
$N=8$	16	3	12	8	16	16	12	4
$N=16$	44	35	32	16	40	40	32	12
$N=32$	116	91	80	32	96	96	80	28
$N=64$	292	227	192	64	224	224	192	60
$N=128$	708	547	448	128	512	512	448	124
$N=256$	1668	1283	1024	256	1152	1152	1024	252

much research efforts have been leveraged to reduce the blocking effect. In [4,7], a first-order Gauss–Markov model was assumed for the images, and then it was shown that the image can be decomposed into a boundary response and a residual process given the closed bound boundary information. The boundary response is an interpolation of the block content from its boundary data, whereas the residual process is the interpolation error. An approach in [4] showed that the KLT of the residual process became discrete sine transform (DST) and DCT when the boundary conditions are available in vertical and horizontal directions [4,6,7].

The discrete signal processing based on the discrete Fourier transform (DFT) is popular in orthogonal frequency division multiplexing (OFDM) wireless mobile communication systems [3] such as 3rd generation partnership project long-term evolution (3GPP-LTE), mobile worldwide interoperability for microwave access (WiMAX), international mobile telecommunications-advanced (IMT-Advanced) as well as wireless local area network (WLAN). In addition, wireless personal area network (WPAN), and broadcasting related applications (digital audio broadcasting (DAB), digital video broadcasting (DVB), digital multimedia broadcasting (DMB)) are based on DFT. Furthermore, the Haar-based wavelet transform (HWT) is also very useful in the joint photographic experts group committee in 2000 (JPEG-2000) standard [2,9]. Thus, different applications require different types of unitary matrices and their decompositions. From this reason, in this paper we will propose a unified hybrid algorithm which can be used in the mentioned several applications in different purposes.

Compared with the conventional individual matrix decompositions, our main contributions are summarized as follows:

- We propose the diagonal sparse matrix factorization for a unified hybrid algorithm based on the properties of the Jacket matrix [10,11] and the decomposition of the sparse matrix. It has been shown that this matrix decomposition is useful in developing the fast algorithms and characters [20]. Individual DCT-II [1–3,6,7,12], DST-II [4,6,7,13], DFT [3,5,14], and HWT [9]

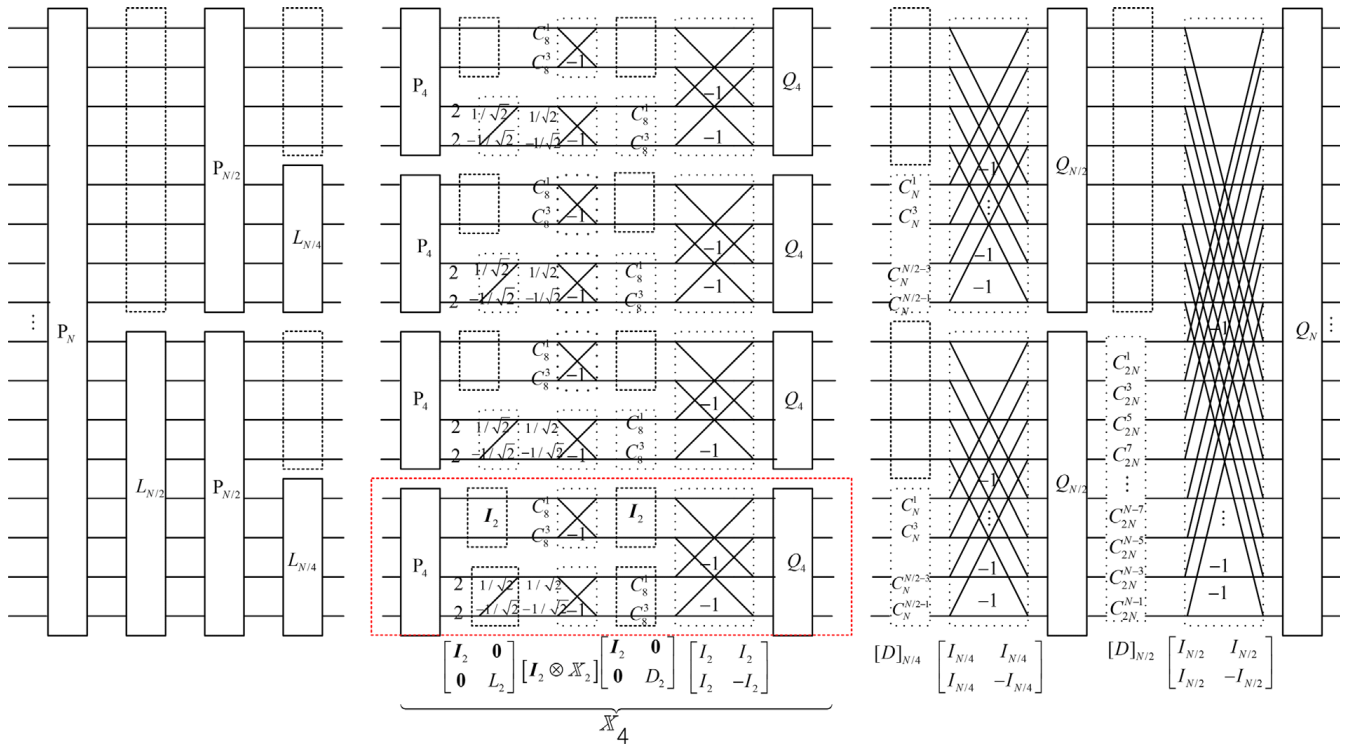


Fig. 1. Regular systematic butterfly data flow of DCT-II.

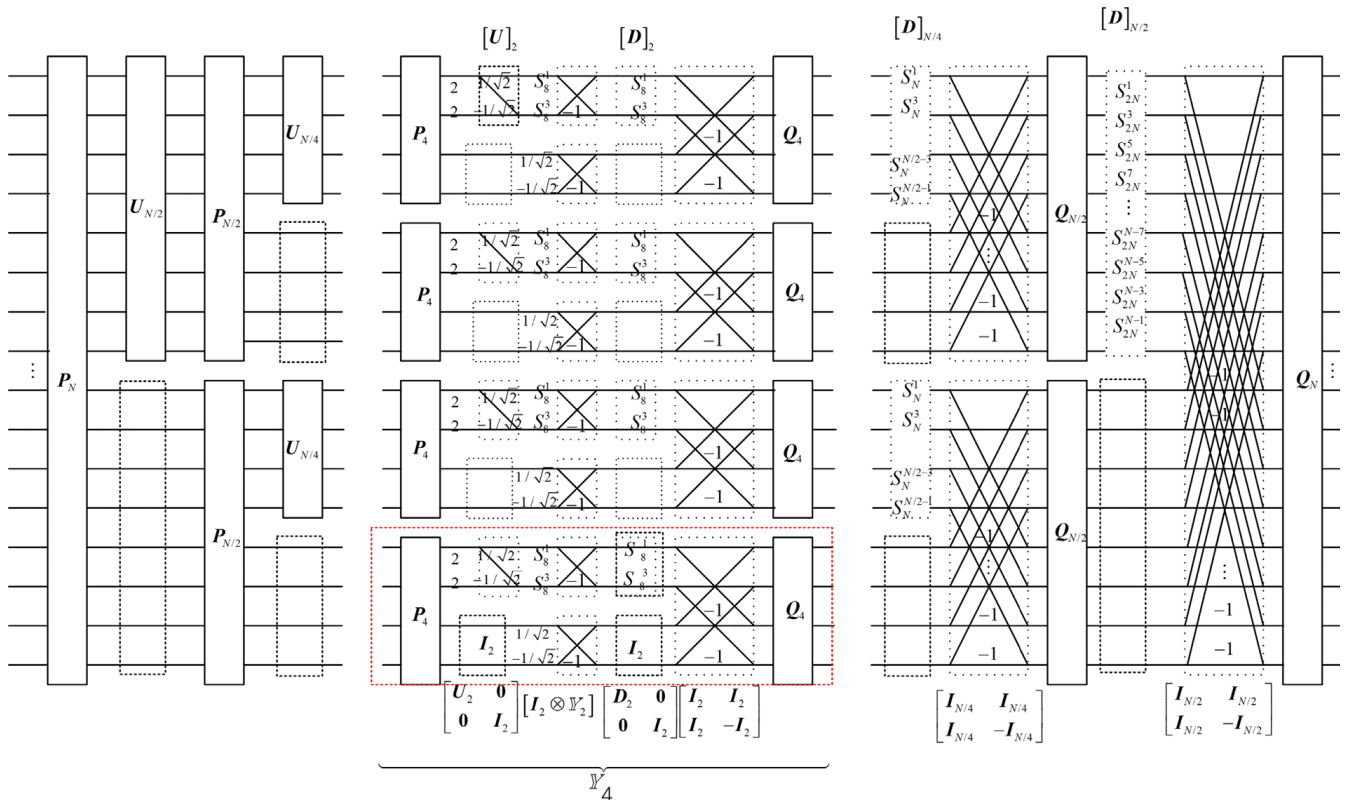


Fig. 2. Regular systematic butterfly data flow of DST-II.

matrices can be decomposed to one orthogonal character matrix and a corresponding special sparse matrix. The inverse of the sparse matrix can be easily obtained from the property of the block (element)-wise inverse Jacket matrix. However, there have been no previous works in the development of the common matrix decomposition supporting these transforms.

- We propose a new unified hybrid algorithm which can be used in the multimedia applications, wireless communication systems, and broadcasting systems at almost the same computational complexity as those of the conventional unitary

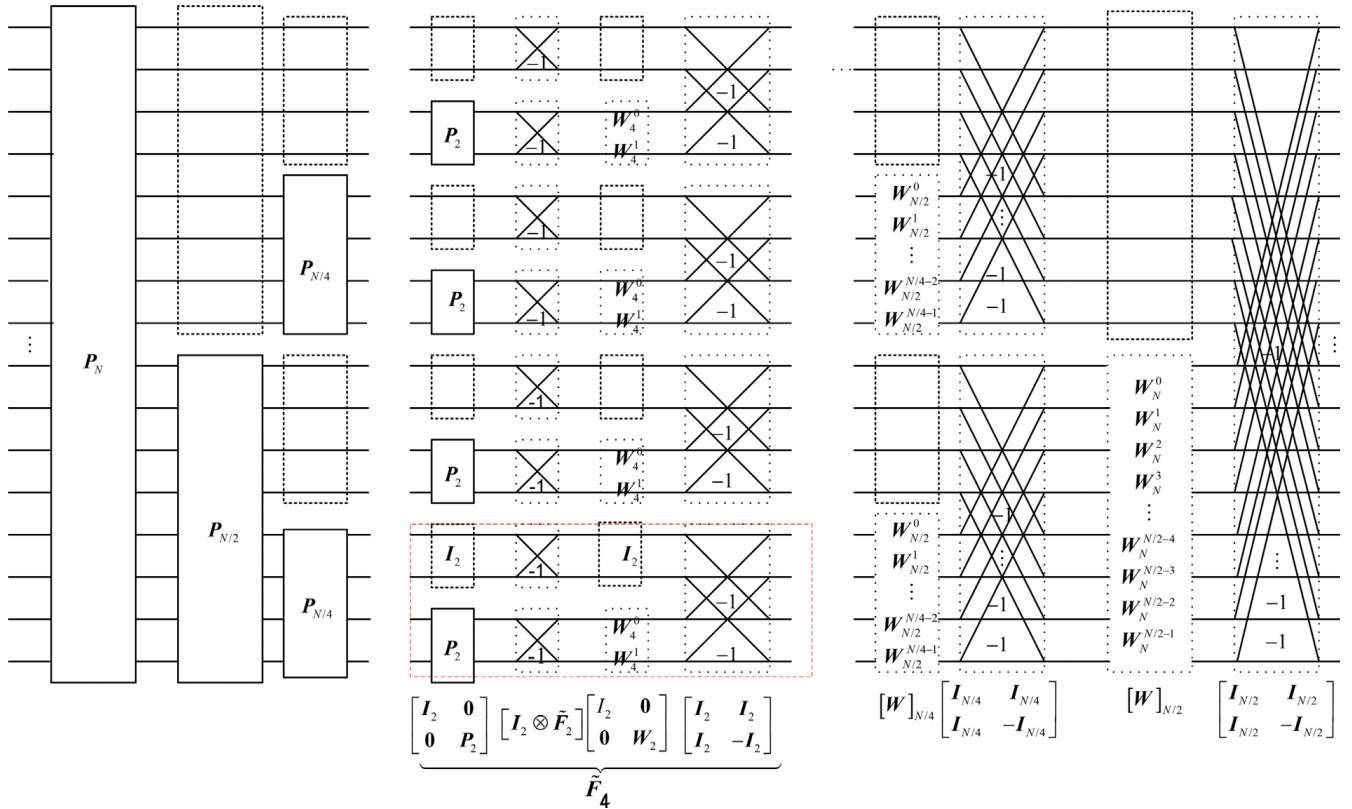


Fig. 3. Butterfly data flow of DFT.

matrix decompositions as summarized in Tables 1 and 2. Compared with the existing unitary matrix decompositions, the proposed hybrid algorithm can be even used to the heterogeneous systems with hybrid multimedia terminals being serviced with different applications. The block (element)-wise diagonal decomposition of DCT-II, DST-II, DFT and HWT as shown in Figs. 1–4 has a similar pattern as Cooley–Tukey’s regular butterfly structures. Moreover, this unified hybrid algorithm can be also applied to the wireless communication terminals requiring a multiuser multiple input–multiple output (MIMO) singular value decomposition (SVD) block diagonalization systems [15,19,20,24] and diagonal channels interference alignment management in macro and femto cell coexisting networks [16]. In [15,16,19,24,25], a block-diagonalized matrix can be applied to the wireless communications MIMO downlink channel [25].

The rest of this paper is organized as follows: in Sections 2 and 3, we present the diagonal block-wise inverse sparse matrix decomposition for the DCT-II and DST-II matrices, respectively. In Sections 4 and 5, we introduce the diagonal element-wise inverse sparse matrix decomposition for the DFT and HWT matrices, respectively. In Section 6, we propose hybrid diagonal block-wise Jacket matrices. The conclusion is given in Section 7.

Notation: The superscript $(\cdot)^T$ denotes transposition; I_N denotes the $N \times N$ identity matrix; 0 denotes an all-zero matrix of appropriate dimensions; $C_l^i \triangleq \cos(i\pi/l)$; $S_l^i \triangleq \sin(i\pi/l)$; $W \triangleq e^{j(2\pi/N)}$; \otimes and \oplus , respectively, denote the Kronecker product and the direct sum.

2. Diagonal block-wise inverse sparse matrix decomposition the DCT-II transform

Definition 1. Let $J_N \triangleq \{a_{ij}\}$ be a matrix, then it is called the Jacket matrix when $J_N^{-1} = (1/N)\{a_{ij}^{-1}\}^T$. That is, the inverse of the Jacket matrix can be determined by its element-wise inverse [10,11,20]. An $N \times N$ row permutation matrix, denoted P_N , is defined by

$$P_N \triangleq \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \quad \text{with } P_2 = I_2 \quad (1a)$$

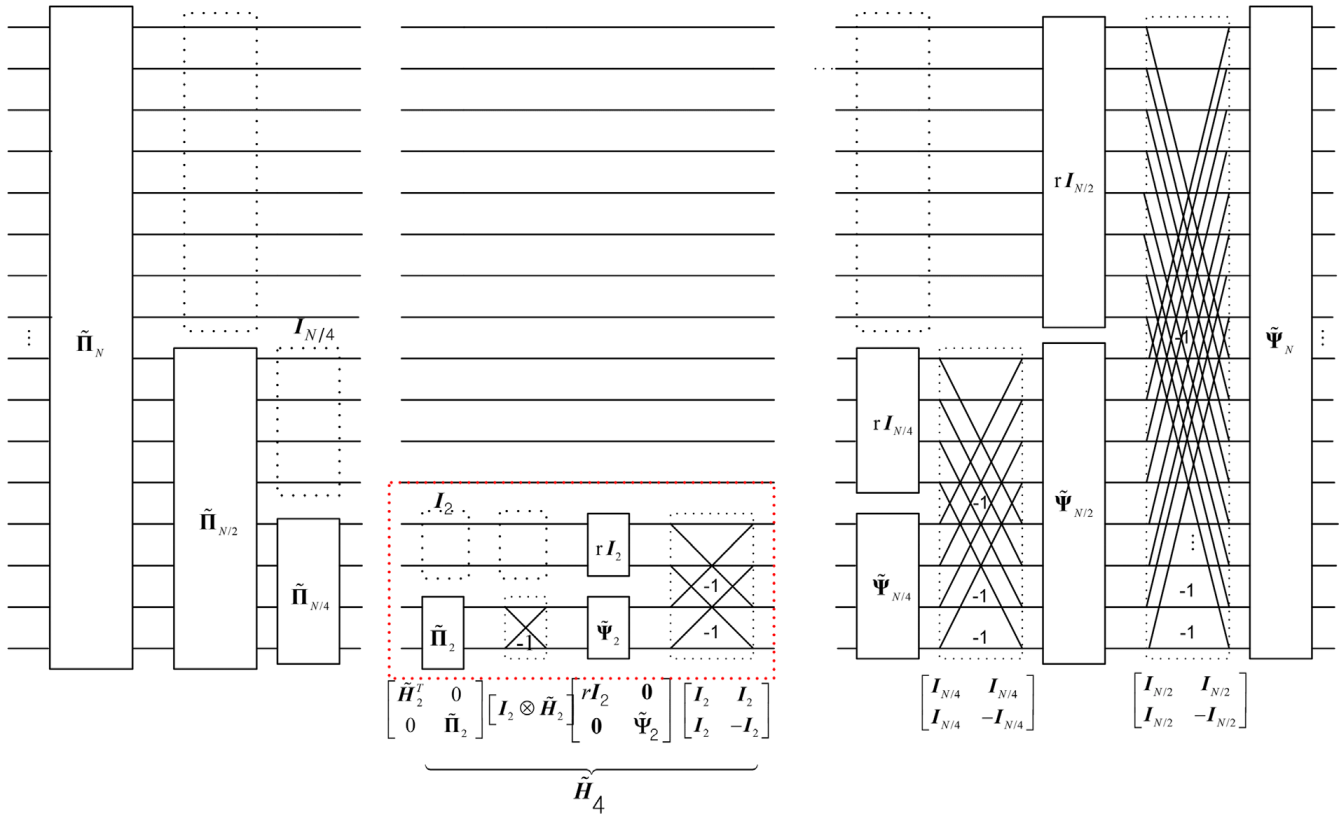


Fig. 4. Butterfly data flow of HWT.

where \mathbf{P}_N elements are determined by the following relationship:

$$\begin{cases} p_{ij} = 1 & \text{if } i = 2j, & 0 \leq j \leq \frac{N}{2} - 1, \\ p_{ij} = 1 & \text{if } i = (2j+1) \bmod N, & \frac{N}{2} \leq j \leq N-1, \\ p_{ij} = 0 & \text{others.} \end{cases} \quad (1b)$$

The block column permutation matrix, denoted by \mathbf{Q}_N , is defined by

$$\mathbf{Q}_N = \mathbf{I}_2 \quad \text{and} \quad \mathbf{Q}_N \triangleq \begin{bmatrix} \mathbf{I}_{N/4} & \mathbf{0}_{N/2} \\ \mathbf{0}_{N/2} & \bar{\mathbf{I}}_{N/4} \end{bmatrix}, \quad N \geq 4 \quad (2)$$

where $\bar{\mathbf{I}}_{N/2}$ denotes a reversed identity matrix. Note that $\mathbf{Q}_N^{-1} = \mathbf{Q}_N$ and $\mathbf{P}_N^{-1} \neq \mathbf{P}_N$, whereas $\mathbf{Q}_N^{-1} = \mathbf{Q}_N^T$ and $\mathbf{P}_N^{-1} = \mathbf{P}_N^T$.

Proposition 1. With the use of the Kronecker product and Hadamard matrices, a higher order block-wise inverse Jacket matrix (BIJM) can be recursively obtained by

$$\mathbf{J}_{2N} = \mathbf{J}_N \otimes \mathbf{H}_2, \quad N \geq 2 \quad (3)$$

then

$$\mathbf{J}_{2N}^{-1} = \frac{1}{N} \mathbf{J}_{2N}^T \quad (4)$$

where the lowest order Hadamard matrix is defined by $\mathbf{H}_2 \triangleq \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

Proof. A proof of this proposition is given in [Appendix A](#).

Note that since the BIJM requires a matrix transposition and then normalization by its size, a class of transforms can be easily inverted as follows:

$$\mathbf{Y}_{2N} = \mathbf{J}_{2N} \mathbf{X}_{2N}$$

and

$$\mathbf{X}_{2N} = \mathbf{J}_{2N}^{-1} \mathbf{Y}_{2N} = \frac{1}{N} \mathbf{J}_{2N}^T \mathbf{Y}_{2N}. \quad (5)$$

Due to a simple operation of the BIJM, we can reduce the complexity order as the matrix size increases. In the following, we shall use this property of the BIJM in developing a hybrid diagonal block-wise transform.

According to [1–4,7], the DCT-II matrix is defined as follows:

$$\mathbf{C}_N = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\phi_0} & C_{4N}^{2k_0\phi_1} & \cdots & C_{4N}^{2k_0\phi_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ C_{4N}^{2k_{N-2}\phi_0} & C_{4N}^{2k_{N-2}\phi_1} & \cdots & C_{4N}^{2k_{N-2}\phi_{N-1}} \end{bmatrix} = \sqrt{\frac{2}{N}} \mathbb{X}_N \quad (6)$$

where $\phi_i = 2i + 1$ and $k_i = i + 1$. We first define a permuted DCT-II matrix $\tilde{\mathbf{C}}_N \triangleq \mathbf{P}_N^{-1} \mathbf{C}_N \mathbf{Q}_N^{-1} = \sqrt{(2/N)} \mathbf{P}_N^{-1} \mathbb{X}_N \mathbf{Q}_N^{-1}$. We can readily show that the matrix \mathbb{X}_N can be constructed recursively as follows:

$$\mathbb{X}_N = \mathbf{P}_N \begin{bmatrix} \mathbb{X}_{N/2} & \mathbb{X}_{N/2} \\ \mathbf{B}_{N/2} & -\mathbf{B}_{N/2} \end{bmatrix} \mathbf{Q}_N = \mathbf{P}_N \begin{bmatrix} \mathbb{X}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N \quad (7)$$

Here, the matrix \mathbf{B}_N in (7) is given as

$$\mathbf{B}_N \triangleq \{B_N(m, n) \triangleq C_{4N}^{f(m, n)}\} \quad (8)$$

where $f(m, 1) = 2m - 1$ and $f(m, n + 1) = f(m, n) + 2f(m, 1)$ for $m, n \in \{1, 2, \dots, N/2\}$. For example, the matrix \mathbf{B}_4 is given by

$$\mathbf{B}_4 = \begin{bmatrix} C_{16}^1 & C_{16}^3 & C_{16}^5 & C_{16}^7 \\ C_{16}^3 & -C_{16}^7 & -C_{16}^1 & -C_{16}^5 \\ C_{16}^5 & -C_{16}^1 & C_{16}^7 & C_{16}^3 \\ C_{16}^7 & -C_{16}^5 & C_{16}^3 & -C_{16}^1 \end{bmatrix} \quad (9)$$

Since $\mathbb{X}_{N/2}^{-1} = (4/N) \mathbb{X}_{N/2}^T$ and $\mathbf{B}_{N/2}^{-1} = (4/N) \mathbf{B}_{N/2}^T$, the matrix decomposition in (7) is the form of the matrix product of diagonal block-wise inverse Jacket and Hadamard matrices. The matrix $\mathbf{B}_{N/2}$ is recursively factorized using Lemma 1.

Lemma 1. The matrix \mathbf{B}_N can be decomposed as

$$\mathbf{B}_N = \mathbf{L}_N \mathbb{X}_N \mathbf{D}_N \quad (10)$$

where a lower triangular matrix \mathbf{L}_N is defined by $\mathbf{L}_N \triangleq \{L_N(m, n)\}$ with elements

$$L_N(m, n) = \begin{cases} \sqrt{2}(-1)^{m-1}, & \forall m \text{ and } n = 1 \\ 2(-1)^{m-1}(-1)^{n-1}, & m \leq n \\ 0, & m > n \end{cases} \quad (11)$$

and a diagonal matrix \mathbf{D}_N is defined by $\mathbf{D}_N \triangleq \text{diag}\{C_{4N}^{\phi_0}, C_{4N}^{\phi_1}, \dots, C_{4N}^{\phi_{N-1}}\}$.

Proof. A proof of this lemma is provided in Appendix B.

Using (10), we first rewrite (7) as

$$\begin{aligned} \mathbb{X}_N &= \mathbf{P}_N \begin{bmatrix} \mathbb{X}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{N/2} \mathbb{X}_{N/2} \mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N \\ &= \mathbf{P}_N \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{N/2} \end{bmatrix} [\mathbf{I}_2 \otimes \mathbb{X}_{N/2}] \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N \end{aligned} \quad (12)$$

which can be evaluated recursively as follows:

$$\mathbb{X}_N = \mathbf{P}_N \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{N/2} \end{bmatrix} \times \left[\mathbf{I}_2 \otimes \left[\cdots \left[\mathbf{I}_2 \otimes \underbrace{\left[\mathbf{P}_4 \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 \end{bmatrix} [\mathbf{I}_2 \otimes \mathbb{X}_2] \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \mathbf{Q}_4 \right]}_{\mathbb{X}_4} \right] \cdots \right] \right] \times \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N \quad (13)$$

Note that in (13) a 2×2 Hadamard matrix is defined by $\mathbb{X}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Also, applying the Kronecker product of \mathbf{I}_2 and $\mathbb{X}_4, \mathbb{X}_8$ can be obtained. Keep applying the Kronecker product of \mathbf{I}_2 and $\mathbb{X}_{N/2}$, the final equivalent form of \mathbb{X}_N is obtained. Thus, the proposed systematic decomposition is based on the Jacket and Hadamard matrices.

In [17], the author proposed a recursive decimation-in-frequency algorithm, where the same decomposition specified in (10) was used. However, due to using a different permutation matrix, a different recursive form was obtained. A different recursive decomposition was proposed in [18]. Four different matrices, such as the first matrix, the last matrix, the odd numbered matrix, and the even number matrix, were proposed. Compared to the decomposition in [18], the proposed

decomposition is seen to be more systematic and requires less numbers of additions and multiplications. We show a complexity comparison among the proposed decomposition and other methods in Tables 1 and 2.

Applying (13), we can readily compute $\mathbf{C}_N = \sqrt{(2/N)} \otimes_N$. The inverse of \mathbf{C}_N can be obtained from the properties of the block-wise Jacket inverse:

$$(\mathbf{C}_N)^{-1} = \sqrt{\frac{N}{2}} (\mathbf{Q}_N)^{-1} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}^{-1} \begin{bmatrix} \otimes_{N/2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{N/2}^{-1} \end{bmatrix} \mathbf{P}_N^{-1} = \sqrt{\frac{N}{2}} \mathbf{Q}_N \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \begin{bmatrix} \otimes_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{N/2}^T \end{bmatrix} \mathbf{P}_N^T. \quad (14)$$

The corresponding butterfly data flow diagram of \mathbf{C}_N is given in Fig. 1.

3. Diagonal block-wise inverse sparse matrix decomposition the DST-II transform

The DST-II matrix [1–4,7] can be expressed as follows:

$$\mathbf{S}_N = \sqrt{\frac{2}{N}} \begin{bmatrix} S_{4N}^{2k_0\phi_0} & S_{4N}^{2k_0\phi_1} & \dots & S_{4N}^{2k_0\phi_{N-1}} \\ S_{4N}^{2k_1\phi_0} & S_{4N}^{2k_1\phi_1} & \dots & S_{4N}^{2k_1\phi_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{4N}^{2k_{N-2}\phi_0} & S_{4N}^{2k_{N-2}\phi_1} & \dots & S_{4N}^{2k_{N-2}\phi_{N-1}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \dots & -\frac{1}{\sqrt{2}} \end{bmatrix} = \sqrt{\frac{2}{N}} \mathbb{Y}_N. \quad (15)$$

Similar to the procedure we have used in the DCT-II matrix, we first define the permuted DST-II matrix, $\tilde{\mathbf{S}}_N$, as follows:

$$\tilde{\mathbf{S}}_N \triangleq \mathbf{P}_N^{-1} \mathbf{S}_N \mathbf{Q}_N^{-1} = \sqrt{\frac{2}{N}} \mathbf{P}_N^{-1} \mathbb{Y}_N \mathbf{Q}_N^{-1}. \quad (16)$$

From (16), we can have a recursive form for \mathbb{Y}_N as

$$\mathbb{Y}_N = \mathbf{P}_N \begin{bmatrix} \mathbf{A}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N \quad (17)$$

where the submatrix \mathbf{A}_N can be calculated by

$$\mathbf{A}_N = \mathbf{U}_N \mathbb{Y}_N \mathbf{D}_N \quad (18)$$

where \mathbf{U}_N and \mathbf{D}_N are, respectively, upper triangular and diagonal matrices. The upper triangular matrix $\mathbf{U}_N \triangleq \{U_N(m, n)\}$ is defined as follows:

$$U_N(m, n) = \begin{cases} \sqrt{2}(-1)^{m-1}, & \forall m \text{ and } n = N \\ 2(-1)^{m-1}(-1)^{n-1}, & m \geq n \\ 0, & m < n \end{cases} \quad (19)$$

whereas the matrix \mathbf{D}_N is defined as before in (10). The derivation of (18) is given in Appendix C. Recursively applying (18) in (17), we can find that

$$\begin{aligned} \mathbb{Y}_N &= \mathbf{P}_N \begin{bmatrix} \mathbf{A}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N = \mathbf{P}_N \begin{bmatrix} \mathbf{U}_{N/2} \mathbb{Y}_{N/2} \mathbf{D}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N. \\ \mathbf{Q}_N &= \mathbf{P}_N \begin{bmatrix} \mathbf{U}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N/2} \end{bmatrix} [\mathbf{I}_2 \otimes \mathbb{Y}_{N/2}] \begin{bmatrix} \mathbf{D}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N. \end{aligned} \quad (20)$$

Further applying (17) to the Kronecker product $[\mathbf{I}_2 \otimes \mathbb{Y}_{N/2}]$, the following general recursive form for the DST-II matrix can be obtained as

$$\mathbb{Y}_N = \sqrt{\frac{2}{N}} \mathbf{P}_N \begin{bmatrix} \mathbf{U}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N/2} \end{bmatrix} \times \left[\mathbf{I}_2 \otimes \left[\dots \left[\mathbf{I}_2 \otimes \underbrace{\left[\mathbf{P}_4 \begin{bmatrix} \mathbf{U}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} [\mathbf{I}_2 \otimes \mathbb{Y}_2] \begin{bmatrix} \mathbf{D}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \mathbf{Q}_4 \right]}_{\mathbb{Y}_4} \right] \dots \right] \right] \times \begin{bmatrix} \mathbf{D}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{Q}_N. \quad (21)$$

Note that if we compare (21) and (13), a similarity can be found in the proposed matrix decompositions. That is, starting from the common lowest order $\mathbb{Y}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, the discrete sine kernel matrix is recursively constructed. Especially,

applying the relationship of $\mathbf{U}_N = \tilde{\mathbf{I}}_N \tilde{\mathbf{L}}_N \tilde{\mathbf{I}}_N$, where

$$\tilde{\mathbf{I}}_N \triangleq \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}$$

denotes the opposite diagonal identity matrix, the butterfly data flow of the DST-II matrix can be obtained from that of the proposed DCT-II decomposition. The butterfly data flow graph of the DST-II matrix is shown in Fig. 2.

Now utilizing the properties of the BIJM, we can first obtain

$$\begin{bmatrix} \mathbf{A}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2} \end{bmatrix}^{-1} = \frac{2}{N} \begin{bmatrix} \mathbf{A}_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2}^T \end{bmatrix} \quad (22)$$

such that the inverse of the matrix \mathbf{S}_N is given by

$$\mathbf{S}_N^{-1} = \sqrt{\frac{2}{N}} \mathbf{Q}_N \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \mathbb{Y}_{N/2}^T \end{bmatrix} \mathbf{P}_N^T. \quad (23)$$

Note that applying again the properties of the BIJM and (18), a recursive form of the inverse DST-II can be easily obtained.

4. Diagonal element-wise inverse sparse matrix decomposition of DFT transform

The DFT is a Fourier representation of a given sequence $\{x(n)\}$, that is,

$$\mathbf{X}(n) = \sum_{m=0}^{N-1} x(m) \mathbf{W}^{nm}, \quad 0 \leq n \leq N-1 \quad (24)$$

where $\mathbf{W} = e^{-j2\pi/N}$. The N -point DFT matrix can be denoted by $\mathbf{F}_N \triangleq \{\mathbf{W}^{nm}\}$. The $N \times N$ Sylvester Hadamard matrix is denoted by \mathbf{H}_N . The Sylvester Hadamard matrix is generated by the successive Kronecker products:

$$\mathbf{H}_N = \mathbf{H}_2 \otimes \mathbf{H}_{N/2} \quad (25)$$

for $N = 4, 8, \dots$. In (25), we have defined $\mathbf{H}_2 \triangleq \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. We decompose a sparse matrix $\mathbf{E}_N \triangleq \mathbf{P}_N \tilde{\mathbf{F}}_N \mathbf{W}_N$ in the following way:

$$\mathbf{F}_N = [\mathbf{P}_N]^T \tilde{\mathbf{F}}_N$$

and

$$\tilde{\mathbf{F}}_N = \begin{bmatrix} \tilde{\mathbf{F}}_{N/2} & \tilde{\mathbf{F}}_{N/2} \\ \mathbf{E}_{N/2} & -\mathbf{E}_{N/2} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{F}}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \quad (26)$$

where $\mathbf{E}_{N/2}$ is further decomposed by Lemma 1 as

$$\mathbf{E}_{N/2} = \mathbf{P}_{N/2} \tilde{\mathbf{F}}_{N/2} \mathbf{W}_{N/2} \quad (27)$$

where \mathbf{W}_N is the diagonal complex unit for the N -point DFT matrix. That is, we have $\mathbf{W}_N = \text{diag}\{W^0, \dots, W^{N-1}\}$. Similar to the development for DCT-II and DST-II, we first rewrite (26) using (27) as

$$\tilde{\mathbf{F}}_N = \begin{bmatrix} \tilde{\mathbf{F}}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{N/2} \tilde{\mathbf{F}}_{N/2} \mathbf{W}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{N/2} \end{bmatrix} [\mathbf{I}_2 \otimes \tilde{\mathbf{F}}_{N/2}] \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}. \quad (28)$$

Note that $[\mathbf{I}_2 \otimes \tilde{\mathbf{F}}_{N/2}]$ in (28) can be recursively decomposed in the following way:

$$\tilde{\mathbf{F}}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{N/2} \end{bmatrix} \times \left[\mathbf{I}_2 \otimes \underbrace{\left[\cdots \mathbf{P}_4 \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 \end{bmatrix} [\mathbf{I}_2 \otimes \tilde{\mathbf{F}}_2] \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \cdots \right]}_{\tilde{\mathbf{F}}_4} \right] \times \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}. \quad (29)$$

It is clear that the form of (29) is the same as that of (13), where we only need to change \mathbf{L}_l to \mathbf{P}_l and \mathbf{D}_l to \mathbf{W}_l for $l \in \{2, 4, 8, \dots, N/2\}$ to convert the DCT-II matrix into the DFT matrix. Consequently, the butterfly data flow of the DFT matrix can be drawn as Fig. 3 using the baseline architecture of DCT-II.

5. Diagonal element-wise inverse sparse matrix decomposition for HWT transform

The discrete Haar wavelet transform (HWT) is expressed by an $N \times N$ matrix \mathbf{H}_N . We can show that $\mathbf{H}_N = \mathbf{H}_N^{-1} = \mathbf{H}_N^T$ and $\mathbf{H}_2 = \begin{bmatrix} r & r \\ r & -r \end{bmatrix}$ with $r = 1/\sqrt{2}$. Let us define following two matrices:

$$\mathbf{\Pi}_N \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{\Psi}_N \triangleq \mathbf{P}_N \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{N/2} \end{bmatrix} \quad \text{for } N \geq 4. \quad (30)$$

Inverses of these defined matrices in (30) are also defined by

$$\mathbf{\Pi}_N^{-1} = \mathbf{\Pi}_N = \tilde{\mathbf{\Pi}}_N \quad \text{and} \quad \tilde{\mathbf{\Psi}}_N \triangleq r\mathbf{\Psi}_N^{-1} = r \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{N/2} \end{bmatrix} \mathbf{P}_N^T. \quad (31)$$

Notice that $\mathbf{\Pi}_N = \tilde{\mathbf{I}}_N$ and $\mathbf{\Pi}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\mathbf{\Psi}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

We can also develop a recursive form for a permuted HWT matrix in the following way, so that the permuted HWT matrix is defined by

$$\begin{aligned} \tilde{\mathbf{H}}_N &= \mathbf{\Pi}_N \mathbf{H}_N \mathbf{\Psi}_N \\ &= r \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}. \end{aligned} \quad (32)$$

Since $\mathbf{H}_N = \mathbf{\Pi}_N^{-1} \tilde{\mathbf{H}}_N \mathbf{\Psi}_N^{-1}$, (32) can be evaluated as

$$\begin{aligned} \tilde{\mathbf{H}}_N &= r \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_{N/2}^{-1} \tilde{\mathbf{H}}_{N/2} \mathbf{\Psi}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} = r \begin{bmatrix} \tilde{\mathbf{H}}_{N/2}^T \tilde{\mathbf{H}}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_{N/2}^{-1} \tilde{\mathbf{H}}_{N/2} \mathbf{\Psi}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \\ &= r \begin{bmatrix} \tilde{\mathbf{H}}_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{H}}_{N/2} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Psi}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\mathbf{H}}_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_{N/2}^{-1} \end{bmatrix} [\mathbf{I}_2 \otimes \tilde{\mathbf{H}}_{N/2}] \begin{bmatrix} r\mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & r\mathbf{\Psi}_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}. \end{aligned} \quad (33)$$

A further recursive factorization in $\tilde{\mathbf{H}}_{N/2}$ leads to a similar factorizations as in (13), (21) and (29). As we keep continuing (33), we can finally obtain the following recursive form for the HWT matrix. Note that $\tilde{\mathbf{H}}_N^T \tilde{\mathbf{H}}_N = \mathbf{I}_N$

$$\tilde{\mathbf{H}}_N = \begin{bmatrix} \tilde{\mathbf{H}}_{N/2}^T & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Pi}}_{N/2} \end{bmatrix} \times \left[\mathbf{I}_2 \otimes \left[\dots \left[\mathbf{I}_2 \otimes \underbrace{\left[\begin{bmatrix} \tilde{\mathbf{H}}_2^T & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Pi}}_2 \end{bmatrix} \left[\mathbf{I}_2 \otimes \tilde{\mathbf{H}}_2 \right] \begin{bmatrix} r\mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Psi}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \right]}_{\tilde{\mathbf{H}}_4} \right] \dots \right] \right] \times \begin{bmatrix} r\mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Psi}}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}. \quad (34)$$

A butterfly data flow corresponding to (34) is shown in Fig. 4.

6. Proposed hybrid architecture for fast computations of DCT-II, DST-II, DFT and HWT matrices

We have derived the recursive formulas for DCT-II, DST-II, DFT, and HWT. The derived results show that DCT-II, DST-II, DFT, and HWT matrices can be unified by using a similar sparse matrix decomposition algorithm, which is based on the block-wise Jacket matrix and diagonal recursive architecture with different characters. The conventional method is only converted from DFT to DCT-II, DST-II, and DWT. However, the proposed method can be universally switching from DCT-II to DST-II, DFT and HWT. Figs. 1–4 exhibit the similar recursive flow diagrams and motivate us to develop universal hybrid architecture via switching mode selection. Moreover, the butterfly data flow graphs have $\log_2 N$ stages. From Fig.1, we can generate Figs. 2 and 3 according to the following proposed ways.

6.1. From DCT-II to DST-II

The N -point DCT-II of $\{x(n)\}$ is given by

$$X_N^{DCT}(m) = c_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{m(2n+1)\pi}{2N} = c_m \sqrt{\frac{2}{N}} C_N \mathbf{x} \quad (35)$$

$$\text{where } m, n = 0, 1, \dots, N-1 \text{ and } c_m = \begin{cases} 1, & m \neq 0 \\ 1/\sqrt{2}, & m = 0. \end{cases}$$

The N -point DST-II of $\{x(n)\}$ is given by

$$X_N^{DST}(m) = s_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \sin \frac{(m+1)(2n+1)\pi}{2N} = s_m \sqrt{\frac{2}{N}} S_N \mathbf{x} \quad (36)$$

$$\text{where } m, n = 0, 1, \dots, N-1 \text{ and } s_m = \begin{cases} 1, & m \neq N-1 \\ 1/\sqrt{2}, & m = N-1. \end{cases}$$

Let C_N and S_N be orthogonal $N \times N$ DCT-II and DST-II matrices, respectively. Also, $\mathbf{x} = [x(0)x(1)\dots x(N-1)]^T$ denotes the column vector for the data sequence $\{x(n)\}$. Substituting $m = N - k - 1, 2, \dots, N$ into (35), we have

$$C_N(N-k-1) = c_{N-k} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)(N-k-1)\pi}{2N}, \quad k = 0, 1, 2, \dots, N-1 \quad (37)$$

Using the following trigonometric identity:

$$\begin{aligned} \cos \left(\frac{(2n+1)\pi}{2} - \frac{(2n+1)(k+1)\pi}{2N} \right) &= \cos \left(\frac{(2n+1)\pi}{2} \right) \cos \left(\frac{(2n+1)(k+1)\pi}{2N} \right) + \sin \left(\frac{(2n+1)\pi}{2} \right) \sin \left(\frac{(2n+1)(k+1)\pi}{2N} \right) \\ &= (-1)^n \sin \left(\frac{(2n+1)(k+1)\pi}{2N} \right) \end{aligned} \quad (38)$$

(37) becomes

$$C_N(N-k-1) = c_{N-k} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} (-1)^n x(n) \sin \frac{(2n+1)(k+1)\pi}{2N} \quad (39)$$

where $C_N(N-k-1)$ represents the reflected version of $C_N(k)$ and this can be achieved by multiplying the reversed identity matrix \bar{I}_N to C_N . Eq. (39) can be represented in a more compact matrix multiplication form [13]:

$$\mathbf{S}_N = \bar{I}_N \mathbf{C}_N \mathbf{M}_N \Leftrightarrow \mathbf{C}_N = \bar{I}_N \mathbf{S}_N \mathbf{M}_N \quad (40)$$

where $\mathbf{M}_N = [\mathbf{M}_2 \otimes \mathbf{I}_{N/2}]$ and $\mathbf{M}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

Then, the DST-II matrix is obtained from the DCT-II matrix. Note that compatibility property exists in the DCT-II and DST-II.

6.2. From DCT-II and DST-II to DFT

We develop a relation between the circular convolution operation in the discrete cosine and sine transform domains. We need to measure a half of the total coefficients. The main advantage of a proposed new relation is that the input sequences to be convolved need not to be symmetrical or asymmetrical. Thus, the transform coefficients can be either symmetric or asymmetric [23].

From (35) and (36), it changes to coefficient for circular convolution (C) format as shown in Fig. 6. Thus, we have the following equations:

$$\begin{aligned} X_N^{DCT-IIC}(m) &\triangleq 2 \sum_{n=0}^{N-1} x(n) \cos \left(\frac{m(2n+1)}{2N} \right), \quad m = 0, 1, \dots, N-1 \text{ and} \\ X_N^{DST-IIC}(m) &\triangleq 2 \sum_{n=0}^{N-1} x(n) \sin \left(\frac{m(2n+1)}{2N} \right), \quad m = 1, \dots, N. \end{aligned} \quad (41)$$

We can rewrite (24) as follows:

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi mn/N}, \quad m = 0, 1, \dots, N-1 \quad (42)$$

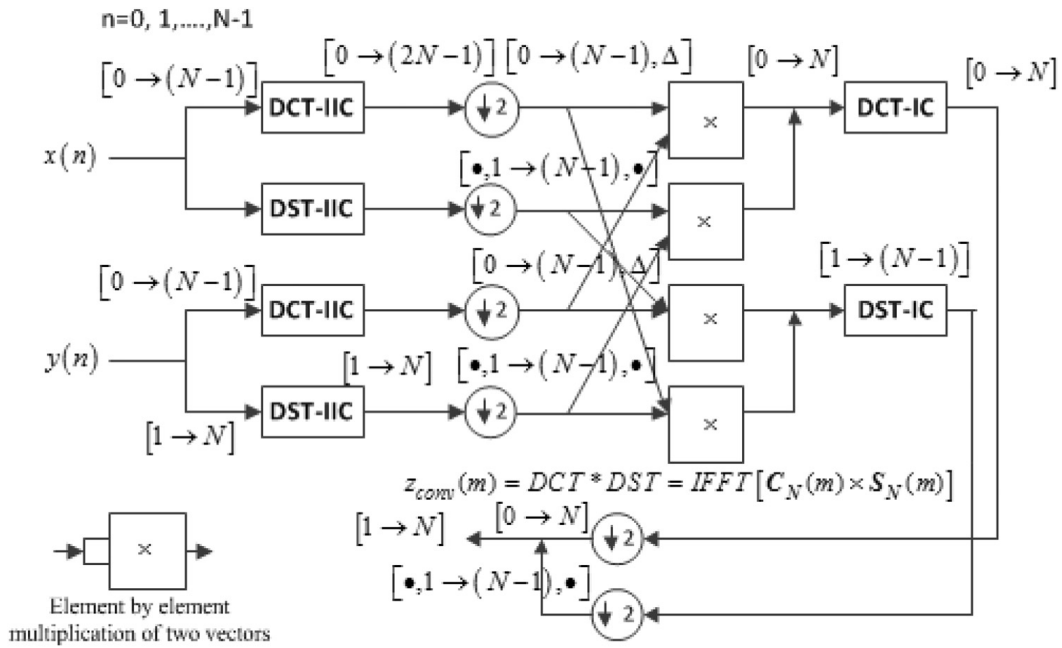


Fig. 6. Circular convolution computes DCTs and DSTs of the FFTs. $(\downarrow 2)$ represents (2:1) decimation. $*$ represents circular convolution. $[0 \rightarrow (N-1), \Delta]$ represents a single zero padding after N DCT/DST coefficients. \bullet represents the place always takes zero as a value and Δ represents an appended zero.

$$= 2 \sum_{n=0}^{N-1} x(n) e^{-j(m(2n+1)\pi/N)} = 2 \sum_{n=0}^{N-1} x(n) \left(\cos \left[\frac{m(2n+1)\pi}{N} \right] - j \sin \left[\frac{m(2n+1)\pi}{N} \right] \right). \quad (43)$$

Comparing the first term of (41) with the first one of (43), it can be seen that $2 \sum_{n=0}^{N-1} x(n) (\cos [(m(2n+1)\pi)/N])$ is decimated and asymmetrically extended of (41) with index $m = 0 : N - 1$. Similarly, $2 \sum_{n=0}^{N-1} x(n) (\sin [(m(2n+1)\pi)/N])$ is decimated and symmetrically extended of (41) with index $m = 1 : N$. From Figs. 5(b) and 6, it is observed that after a proper zero padding of the sequences, a symmetric convolution can be used to perform the linear convolution. The circular convolution of cosine and sine periodic sequences in time and spatial domain is equivalent to the multiplication in the DFT domain. Then, the DFT matrix is obtained from the DCT-II and DST-II matrices.

6.3. From DCT-II to HWT

In this subsection, we first need to change $L_{N/2}$ to $\tilde{H}_{N/2}$ and $D_{N/2}$ to $\tilde{P}_{N/2}$. If a discrete wavelet transform (DWT) is discretely sampled, it becomes the two point discrete Fourier transform, where we allow to use $1/2$ and 1 in the matrix and its inverse. The orthogonal matrix has $1/\sqrt{2}$ in both. Note that the DCT, DST and DFT matrices are based on $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, whereas the HWT matrix is based on $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. We can see that

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{then} \quad \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [I]_2 \quad (44)$$

Since for a 2×2 Haar matrix, it is given that

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} r & r \\ r & -r \end{bmatrix} = \begin{bmatrix} L \\ B \end{bmatrix} \quad (45)$$

where L is low pass and B is high pass band. Moreover, we can have

$$\begin{bmatrix} L \\ B \end{bmatrix} [L^T B^T] = \begin{bmatrix} LL^T & LB^T \\ BL^T & BB^T \end{bmatrix} = [I]_2. \quad (46)$$

Note that the $[I]_2$ matrix in (44) and (46) has been well matched to the proposed Jacket–Hadamard matrix, which has been proved in Appendix A.

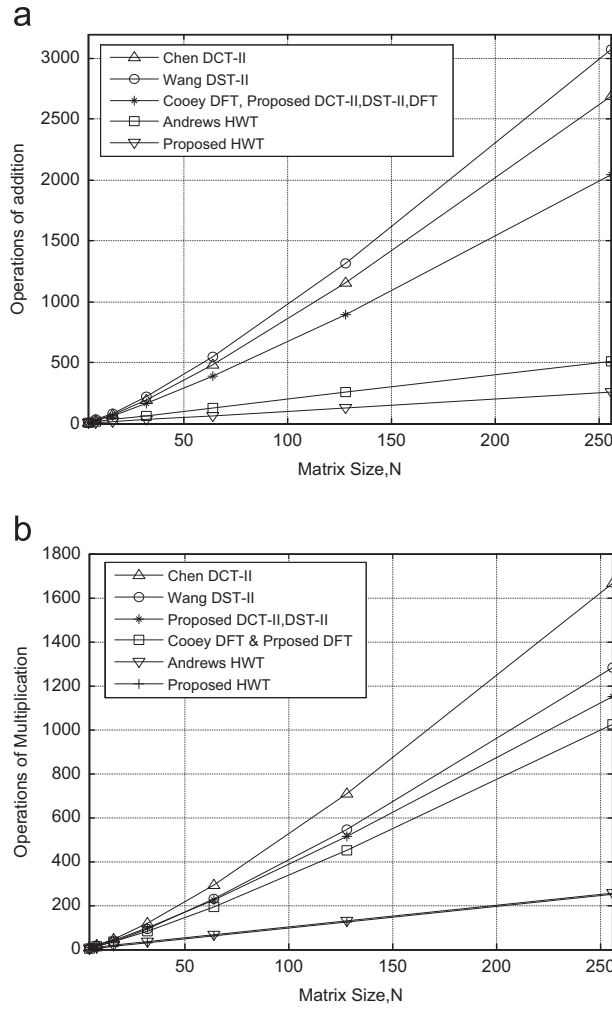


Fig. 7. Comparison plot for computational complexity of conventional methods and the proposed ones. (a) Addition counts and (b) multiplication counts.

Thus, after multiplying the diagonal $N \times N$ matrix, the HWT can be obtained as

$$\mathbf{M}_{HWT} = \mathbf{I}_{N/2} \otimes \begin{bmatrix} r\mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{I}}_{N/2}\tilde{\mathbf{H}}_{N/2} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix} \quad (47)$$

6.4. Circulant channel matrix based on the Hadamard and Haar matrices

This subsection is devoted to a special type of block circulant matrices. A circulant matrix \mathbf{C}_{circ} is fully specified by one vector which appears as the first column of \mathbf{C}_{circ} . Circulant matrices are important because they can be diagonalized by the DFT, and hence linear equations that contain them can be quickly solved using the fast Fourier transform (FFT) [26]. From (44) and (45), we have

$$\mathbf{C}_{circ} = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_0 \end{bmatrix}, \quad (48)$$

where $c_0 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and $c_1 = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}$.

The block circulant matrices are given by

$$\mathbf{C}_{circ} = [\mathbf{V}_2 \otimes \mathbf{I}_2] \begin{bmatrix} A_0 & \\ & A_1 \end{bmatrix} [\mathbf{V}_2 \otimes \mathbf{I}_2]^* \quad (49)$$

where

$$\mathbf{V}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

From (45), \mathbf{V}_2 is the 2×2 Haar matrix and A_0 and A_1 are eigenvalues of circulant matrices

$$\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} = \sqrt{2}[\mathbf{V}_2 \otimes \mathbf{I}_2]^* \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \sqrt{2} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \sqrt{2} \begin{bmatrix} c_0 + c_1 \\ c_0 - c_1 \end{bmatrix} \quad (50)$$

where $A_0 = c_0 + c_1$ and $A_1 = c_0 - c_1$. Finally, the eigenvalues and eigenvectors of circulant matrices are

$$\{2, -2, 2, -2\} \quad \text{and} \quad \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} \right\}.$$

6.5. Unified hybrid fast algorithm

Based on the above conversions from the proposed decomposition of DCT-II, we can form a hybrid fast algorithm that can cover DCT-II, DST-II, DFT, and HWT. The general block diagram of the proposed hybrid fast algorithm is shown in Fig. 5. The common basis function of $[\mathbf{I}]_{N/2^{h-1}} \otimes \begin{bmatrix} \mathbf{I}_{2^{h-1}} & \mathbf{I}_{2^{h-1}} \\ \mathbf{I}_{2^{h-1}} & -\mathbf{I}_{2^{h-1}} \end{bmatrix}$ is multiplied by the sequence with different transforms as like as bracket (((·))). The requiring computational complexity of individual DCT-II, DST-II, DFT, and HWT is summarized in Tables 1 and 2. It can be seen in Fig. 7 that the proposed hybrid algorithm scheme requires a much less computational complexity in addition and multiplication compared to those of the decompositions proposed by [13,18,22]. In addition, compared to these transforms, the proposed hybrid fast algorithm can be efficiently extensible to larger transform sizes due

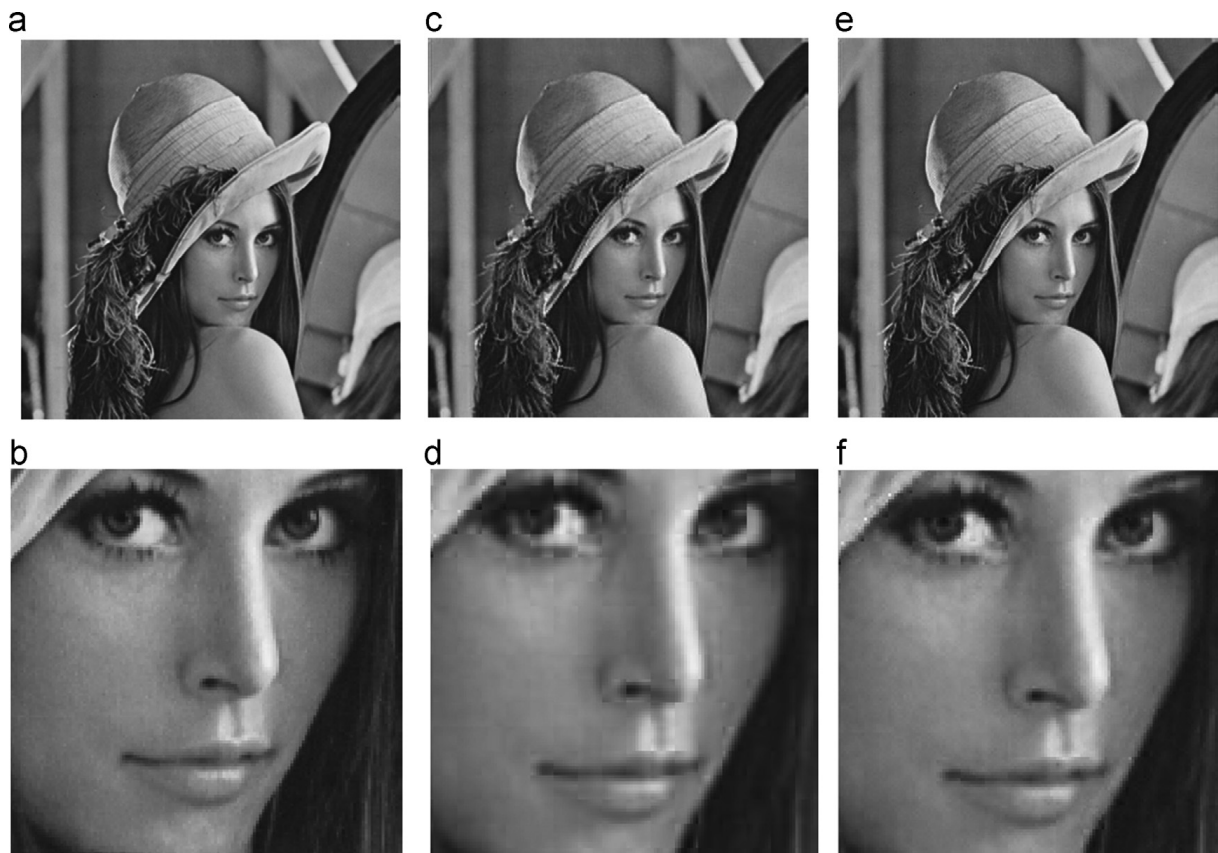


Fig. 8. Image coding results showing DCT-II only and jointly optimized DCT/DST-II compression. (a) Original Lena image, (b) zoomed original Lena image, (c) DCT-II compressed Lena image (PSNR=32.38db), (d) zoomed DCT-II compressed Lena image, (e) DCT/DST-II compressed Lena image (PSNR=35.12db) and (f) zoomed DCT/DST-II compressed Lena image.

to its diagonal block-wise inverse operation of recursive structure. Moreover, the proposed hybrid structure is easily extended to cover different applications. For example, a base station of wireless communication terminal delivers a compressed version of multimedia data via wireless communications network. Either DCT-II or DST-II can be used in compressing various multimedia data since the proposed decomposition is based on block diagonalization it can significantly reduce its complexity due to its simple structure [19,20,24], for various input multimedia sources. The DCT image coding can be easily implemented in the proposed hybrid structure as shown in Fig. 5(b). If the DCT-II is multiplied by $\bar{\mathbf{I}}_N \mathbf{C}_N \mathbf{M}_N$, then we get DST. If the DCT and DST-II are convolved in time and frequency domain and multiplied by $2e^{-j\pi m/N}$, the DFT matrix can be obtained. Finally, if the DCT is multiplied by $\mathbf{I}_{N/2} \otimes \begin{bmatrix} r\mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{I}}_{N/2} \tilde{\mathbf{H}}_{N/2} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}$, then we get HWT. Thus, the proposed hybrid algorithm enables the terminal to adapt to different transforms efficiently.

6.6. Numerical simulations

As shown in [7], the coding performance of DST outperforms DCT at high correlation values (ρ), and it is very close to that of the KLT. Since the basis vectors of DCT maximize their energy distribution at both ends, hence the discontinuity appears at block boundaries due to quantization effects. However, since the basis vectors of DST minimizes their energy distribution at other ends, DST provides smooth transition between neighboring blocks. Therefore, the proposed hybrid transform coding scheme provides a consistent reconstruction and preserves more details, as shown in Fig. 8 with a size of 512×512 and 8 bits quantization.

Now consider an $N \times N$ block of pixels, X , containing x_{ij} , $i, j = 1, 2, \dots, N$. We can write 2-D transformation of the k th block X as $Y_s = T_s Q X_k Q^T$ and $Y_c = T_c X_k$.

Depending on the availability of boundary values (in top-boundary and left-boundary) in images, the hybrid coding scheme accomplishes the 2-D transform of a block pixel in two sequential 1-D transforms separately performed on rows and columns. Therefore, the choice of the 1-D transform for each direction is dependent on the corresponding prediction boundary condition.

- Vertical transform (for each column vector): employ DST if top boundary is used for prediction; otherwise use DCT.
- Horizontal transform (for each row vector): employ DST if left boundary is used for prediction; otherwise use DCT.

What we have observed from numerical experiments is that the combined scheme over DCT-II performs better only in perceptual clarity as well as PSNR. Jointly optimized spatial prediction and block transform (see Fig. 8(e) and (f)) using DCT/DST-II (PSNR 35.12 dB) outperforms the approach that uses only DCT-II (PSNR 32.38 dB). Less blocky artifacts are revealed compared to that of DCT-II (24 dB). Without *a priori* knowledge of boundary condition, DCT-II performs better than any other block transforms coding. The worst result is obtained by using only DST-II.

7. Conclusion

In this paper, we have derived a unified fast hybrid diagonal block-wise transform based on Jacket–Hadamard matrix. The proposed FHDBT have shown that DCT-II, DST-II, DFT, and HWT can be unified by using the diagonal sparse matrix based on the Jacket matrix and recursive structure with some characters changed from DCT-II to DST-II, DFT and HWT. The FHDBT has used the matrix product of recursively lower order diagonal sparse matrix and Hadamard matrix. The resulting signal flow graphs of DCT-II, DST-II, DFT, and HWT have a regular systematic butterfly structure. Therefore, the complexity of the proposed unified hybrid algorithm has been much less as its matrix size gets larger. This butterfly structure has grown by a recursive nature of the fast hybrid Jacket–Hadamard matrix. Based on a systematic butterfly structure, a unified switching system can be devised. The basis of the DCT, DST, and DFT is Hadamard matrix, whereas Haar matrix is the basis for HWT. We have also applied the circulant channel matrix in our proposed method. Thus, the FHDBT can be effectively applied to the heterogeneous transform systems having various matrix dimensions. Jointly optimized DCT and DST-II compression scheme have revealed a better performance (about 3 dB) over the DCT only compression method. This block-wise diagonal matrices can be applied in areas such as MIMO downlink and subchannel decompose for interference alignment.

Acknowledgment

This work was supported by the MEST 2012-002521, NRF, Korea. The first author would like to thank to Professor Gilbert Strang, Department of Mathematics, MIT, for technical discussion.

Appendix A

A proof of Proposition 1

We use mathematical induction to prove Proposition 1. The lowest order BIJM is defined as

$$\mathbf{J}_8 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{C}_2 & \mathbf{C}_2 & -\mathbf{I}_2 \\ \mathbf{I}_2 & \mathbf{C}_2 & -\mathbf{C}_2 & -\mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix} \quad (\text{A.1})$$

where $\mathbf{C}_2 \triangleq (\mathbf{H}_2/\sqrt{2})$. Since

$$\mathbf{J}_8^{-1} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{C}_2^T & \mathbf{C}_2^T & -\mathbf{I}_2 \\ \mathbf{I}_2 & \mathbf{C}_2^T & -\mathbf{C}_2^T & -\mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix} \quad (\text{A.2})$$

Eq. (4) holds for $2N=8$. Now we assume that the BIJM \mathbf{J}_N satisfies (4), i.e., $\mathbf{J}_N \mathbf{J}_N^T = (N/2)\mathbf{I}_N$. Since $\mathbf{J}_{2N} \mathbf{J}_{2N}^T = (\mathbf{J}_N \otimes \mathbf{H}_2)(\mathbf{J}_N \otimes \mathbf{H}_2)^T = (\mathbf{J}_N \mathbf{J}_N^T) \otimes (\mathbf{H}_2 \mathbf{H}_2^T) = (N/2)\mathbf{I}_N \otimes 2\mathbf{I}_2 = N\mathbf{I}_{2N}$, this proposition is proved by mathematical induction that (4) holds for all $2N$. If $N=1$, certainly $\mathbf{J}_2 \mathbf{J}_2^T = \mathbf{I}_2$. This is same as (44) and (46).

Appendix B

A proof of Lemma 1

According to the definition of an $N \times N$ matrix \mathbf{B}_N , \mathbf{B}_N is given as follows:

$$\mathbf{B}_N = \begin{bmatrix} C_{4N}^{\phi_0} & C_{4N}^{\phi_1} & C_{4N}^{\phi_2} & \cdots & C_{4N}^{\phi_{N-1}} \\ C_{4N}^{(2k_0+1)\phi_0} & C_{4N}^{(2k_0+1)\phi_1} & C_{4N}^{(2k_0+1)\phi_2} & \cdots & C_{4N}^{(2k_0+1)\phi_{N-1}} \\ C_{4N}^{(2k_1+1)\phi_0} & C_{4N}^{(2k_1+1)\phi_1} & C_{4N}^{(2k_1+1)\phi_2} & \cdots & C_{4N}^{(2k_1+1)\phi_{N-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{4N}^{(2k_{N-2}+1)\phi_0} & C_{4N}^{(2k_{N-2}+1)\phi_1} & C_{4N}^{(2k_{N-2}+1)\phi_2} & \cdots & C_{4N}^{(2k_{N-2}+1)\phi_{N-1}} \end{bmatrix} \quad (\text{B.1})$$

where $k_i \triangleq i+1$. Since $\cos((2k+1)\phi_m) = 2 \cos(2k\phi_m) \cos(\phi_m) - \cos((2k-1)\phi_m)$, we have

$$C_{4N}^{(2k_i+1)\phi_m} = -C_{4N}^{(2k_i-1)\phi_m} + 2C_{4N}^{(2k_i)\phi_m} C_{4N}^{\phi_m}. \quad (\text{B.2})$$

Using (B.2), \mathbf{B}_N can be decomposed as

$$\mathbf{B}_N = \mathbf{L}_N \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\phi_0} & C_{4N}^{2k_0\phi_1} & C_{4N}^{2k_0\phi_2} & \cdots & C_{4N}^{2k_0\phi_{N-1}} \\ C_{4N}^{2k_1\phi_0} & C_{4N}^{2k_1\phi_1} & C_{4N}^{2k_1\phi_2} & \cdots & C_{4N}^{2k_1\phi_{N-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{4N}^{2k_{N-2}\phi_0} & C_{4N}^{2k_{N-2}\phi_1} & C_{4N}^{2k_{N-2}\phi_2} & \cdots & C_{4N}^{2k_{N-2}\phi_{N-1}} \end{bmatrix} \mathbf{D}_N = \mathbf{L}_N \otimes_N \mathbf{D}_N \quad (\text{B.3})$$

which proves (10) in Lemma 1.

Appendix C

A proof of Eq. (18)

By using the sum and difference formulas for the sine function, we can have

$$\begin{aligned} S_{4N}^{(2k_{N-1}-1)\phi_j} &= C_{4N}^{\phi_j} = S_{4N}^{\phi_{N-j-1}}, \quad S_{4N}^{(2k_i-1)\phi_j} = 2S_{4N}^{2k_i\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_{i+1}-1)\phi_j}, \\ S_{4N}^{(2k_{i+1}-1)\phi_j} &= 2S_{4N}^{2k_i\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_i-1)\phi_j}, \\ S_{4N}^{(2k_0-1)\phi_j} &= (2S_{4N}^{(2k_0-1)\phi_j} - 2S_{4N}^{(2k_1)\phi_j} + \cdots + 2S_{4N}^{(2k_{N-2}-1)\phi_j}) C_{4N}^{\phi_j} = 2S_{4N}^{(2k_0-1)\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_1-1)\phi_j}, \\ S_{4N}^{(2k_1-1)\phi_j} &= (2S_{4N}^{(2k_1-1)\phi_j} - 2S_{4N}^{(2k_1)\phi_j} + \cdots + 2S_{4N}^{(2k_{N-2}+1)\phi_j}) C_{4N}^{\phi_j} = 2S_{4N}^{(2k_1-1)\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_2-1)\phi_j}, \\ &\vdots \\ S_{4N}^{(2k_{N-3}-1)\phi_j} &= (2S_{4N}^{(2k_{N-3}-1)\phi_j} - 2S_{4N}^{(2k_{N-2})\phi_j} + 1) C_{4N}^{\phi_j} = 2S_{4N}^{(2k_{N-3}-1)\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_{N-2}-1)\phi_j}, \end{aligned}$$

$$S_{4N}^{(2k_{N-2}-1)\phi_j} = (2S_{4N}^{(2k_{N-2})\phi_j} - 1)C_{4N}^{\phi_j} = 2S_{4N}^{(2k_{N-2})\phi_j} C_{4N}^{\phi_j} - S_{4N}^{(2k_{N-1}-1)\phi_j} \quad (C.1)$$

where $k_i = i + \Phi_j = 2j + 1, i, j = 0, 1, \dots, N-1$.

By taking (C.1) and the right hand side of (18), we have

$$\mathbf{U}_N \mathbf{Y}_N \mathbf{D}_N = \begin{bmatrix} S_{4N}^{(2k_0-1)\phi_0} & S_{4N}^{(2k_0-1)\phi_1} & \dots & S_{4N}^{(2k_0-1)\phi_{N-1}} \\ S_{4N}^{(2k_1-1)\phi_0} & S_{4N}^{(2k_1-1)\phi_1} & \dots & S_{4N}^{(2k_1-1)\phi_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{4N}^{(2k_{N-1}-1)\phi_0} & S_{4N}^{(2k_{N-1}-1)\phi_1} & \dots & S_{4N}^{(2k_{N-1}-1)\phi_{N-1}} \end{bmatrix}. \quad (C.2)$$

The left hand side of (18) matrix $[\mathbf{A}]_N$ from $[\mathbf{Y}]_N$ can be represented by

$$\mathbf{A}_N = \begin{bmatrix} S_{4N}^{(2k_0-1)\phi_0} & S_{4N}^{(2k_0-1)\phi_1} & \dots & S_{4N}^{(2k_0-1)\phi_{N-1}} \\ S_{4N}^{(2k_1-1)\phi_0} & S_{4N}^{(2k_1-1)\phi_1} & \dots & S_{4N}^{(2k_1-1)\phi_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{4N}^{(2k_{N-1}-1)\phi_0} & S_{4N}^{(2k_{N-1}-1)\phi_1} & \dots & S_{4N}^{(2k_{N-1}-1)\phi_{N-1}} \end{bmatrix}. \quad (C.3)$$

We can obtain (C.2) and (C.3) to be the same and the expression of (18) is correct.

References

- [1] K.R. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, Boston, MA, 1990.
- [2] I.E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed., John Wiley and Sons, Hoboken, NJ, April 2010.
- [3] K.R. Rao, D.N. Kim, J.J. Hwang, *Fast Fourier Transform: Algorithm and Applications*, Springer, New York, NY, 2010.
- [4] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, United States, USA, 1987.
- [5] R. Wang, *Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis*, Cambridge University Press, Cambridge, UK, 2012.
- [6] ITU-T SG16 WP3/JCT-VC, CE 7.5, Performance Analysis of Adaptive DCT/DST Selection, July 2011.
- [7] J. Hai, A. Saxena, V. Melkote, K. Rose, Jointly optimized spatial prediction and block transform for video and image coding, *IEEE Trans. Image Process.* 21 (April (4)) (2012) 1874–1884.
- [8] A. Mellouk, P. Lorenz, A. Boukerche, M.H. Lee, Quality of service based routing algorithms for heterogeneous networks, *IEEE Commun. Mag.* 45 (February (2)) (2007) 65–66.
- [9] G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1996.
- [10] M.H. Lee, A new reverse Jacket transform and its fast algorithm, *IEEE Trans. Circuits Syst. II* 47 (January (1)) (2000) 39–47.
- [11] Z. Chen, M.H. Lee, G. Zeng, Fast cocyclic Jacket transform, *IEEE Trans. Signal Process.* 56 (May) (2008) 2143–2148.
- [12] C.L. Wang, C.Y. Chen, High-throughput VLSI architectures for the 1-D and 2-D discrete cosine transform, *IEEE Trans. Circuits Syst. Video Technol.* 5 (February) (1995) 31–40.
- [13] Z. Wang, Fast Algorithm for the discrete W transform and for the discrete Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.* 32 (August (4)) (1984) 803–816.
- [14] M.H. Lee, High speed multidimensional systolic arrays for discrete Fourier transform, *IEEE Trans. Circuits Syst. II* 39 (December (12)) (1992) 876–879.
- [15] K.J. Kim, Y. Fan, R.A. Iltis, H.V. Poor, M.H. Lee, A reduced feedback precoder for MIMO-OFDM cooperative diversity system, *IEEE Trans. Veh. Technol.* 61 (February) (2012) 584–596.
- [16] U. Jang, K. Cho, W. Ryu, H.J. Lee, Interference management with block diagonalization for macro/femto coexisting networks, *ETRI J.* 34 (June) (2012) 297–307.
- [17] H.S. Hou, A fast recursive algorithm for computing the discrete cosine transform, *IEEE Trans. Acoust. Speech Signal Process.* 35 (October (10)) (1987) 1455–1461.
- [18] W.-H. Chen, C.H. Smith, S.C. Fralick, A fast computational algorithm for the discrete cosine transform, *IEEE Trans. Commun.* 25 (September (9)) (1977) 1004–1009.
- [19] Q.H. Spencer, A. Lee Swindlehurst, M. Haardt, Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels, *IEEE Trans. Signal Process.* 52 (February (2)) (2004) 461–471.
- [20] M.H. Lee, *Jacket Matrices—Construction and Its Application for Fast Cooperative Wireless Signal Processing*, LAP LAMBERT Academic Publishing, Germany, 2012 (November).
- [21] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comput.* 19 (no. 4) (1965) 297–301.
- [22] H.C. Andrews, K.L. Caspari, A generalized technique for spectral analysis, *IEEE Trans. Comput.* 19 (1) (1970) 16–17.
- [23] V.G. Reju, S.N. Koh, I.Y. Soon, Convolution using discrete sine and cosine transforms, *IEEE Signal Process. Lett.* 14 (July (7)) (2007).
- [24] M.H. Lee, M.H.A. Khan, M.A.L. Sarker, Y. Guo, K.J. Kim, A MIMO LTE precoding based on fast diagonal weighted Jacket matrices, in: *Fiber and Integrated Optics*, vol. 31, no. (2), Taylor & Francis, Philadelphia, PA pp. 111–132, March 2012 (Invited Paper).
- [25] M.H.A. Khan, J.u.n. Li, Moon Ho Lee, A block diagonal Jacket matrices for MIMO broadcast channel, in: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Brunel University, UK, June 4–7, 2013.
- [26] J. Gutierrez-Gutierrez, P.M. Crespo, *Block Toeplitz Matrices: Asymptotic Results and Applications*, NOW, the Essence of Knowledge, Boston-Delft, 2012.