EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Fast parametric reciprocal-orthogonal jacket transforms

Moon Ho Lee[1*], Xiao-Dong Zhang[2] and Xueqin Jiang[3]

Abstract

In this paper, we propose a new construction method for a novel class of parametric reciprocal-orthogonal jacket transform (PROJT) having $\frac{9}{4}N$ parameters for a sequence length $N = 2^{r+1}$ that is a power of two, based on the reciprocal-orthogonal parametric (ROP) transform and block diagonal matrices. It is shown that the inverse transform of the proposed PROJT is conveniently obtained by the reciprocal of each elements of the forward matrix and transpose operation. What is more, an efficient algorithm for the computation of the PROJT has been developed with the aid of the matrix decomposition and Kronecker product. Further, the experiments show that the independent parameters in the proposed PROJT are successfully used as additional secret keys for image encryption, watermarking, and error-correcting codes.

1 Introduction

There are a variety of discrete signal orthogonal transforms [1-3], such as discrete Fourier transform (DFT), discrete Hartley transform (DHT), Walsh-Hadamard transform (WHT), Haar transform, slant transform, and discrete cosine transform (DCT), which have various application in digital signal processing, image compressing video processing, and pattern recognition. A lot of services of these transforms are mainly due to their practical usefulness and the existence of fast and efficient algorithms for their computation. However, since each of the well-known transforms [4-9], for example DFT, DHT, WHT, DCT, etc., is fixed without any parameters, each single transform only deals with its special area of applications. In order to best match the given input signal class or the application, many parametric transforms [10-15] with matrices associating a set of parameters are presented to fit the desirable signal by choosing appropriate parameters. An advantage of parametric transforms is the possibility to implement large families of transforms with a unified software/hardware, which is efficient for every representative of the family and may be tuned to the desired transforms [16].

On the other hand, the DCT and the Karhunen-Loeve transform (KLT) have better compaction performance than the slant transform [10,11]. Both the DCT and KLT have more computational complexity than the slant transform. Therefore, the need arises for slant transform improvement schemes that yield performance comparable to that of the KLT and DCT without incurring their computational complexity [10-20]. Therefore, various generalizations of the WHTs and DFTs have been attempted. Lee [6] proposed a center weight Hadamard transform with matrix of order 4. However, the proposed jacket transform has only three parameters at most. Recently, Bouguezel et al. [13] proposed a new class of reciprocal-orthogonal parametric (ROP) transforms, which have $\frac{3N}{2}$ independent parameters for an input data vector of length N. Further, they showed that the inverse of the ROP transforms and matrices is easily obtained and has fast algorithm. Lee et al. [18] proposed a novel class of element-wise inverse jacket transforms (EIJTs) having $2N - 1$ parameters for an input data vector of length $N = 3 \cdot 2^r$. Moreover, Bouguezel et al. [14] proposed new parametric discrete Fourier transforms, Hartley transforms, and algorithms for fast computation. Ding et al. [15] proposed arbitrary-length Walsh-Jacket transforms. Agaian et al. [11] developed a class of generalized parametric Slant-Hadamard transforms with fast algorithm, whose performance is better than the classical one-transform-based model. Chen et al. [21] proposed a fast cocyclic jacket transform

*Correspondence: moonho@jbnu.ac.kr
[1]Division of Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, Korea
Full list of author information is available at the end of the article

over the complex number field. Moreover, many other transforms based on complex field and finite fields were proposed [19,20,22].

In recent years, enormous parametric transforms corresponding to the existing determined transforms have been developed. It has been shown that parametric transforms can have more flexibility and a wider range of applications compared to its original transform. For example, the independent parameters of the fractional discrete transform are used as an additional secret key for watermarking [5], encryption [4,17], error-correcting codes, etc. From this point of view, parametric transforms with matrices described in a unified form and based on a set of parameters become more and more important.

The main purpose of this paper is to propose a fast parametric reciprocal-orthogonal jacket transform (PROJT) having $\frac{9}{4}N$ parameters for an input data vector $N = 2^{r+1}$, which is reciprocal-orthogonal and has a fast and efficient algorithm with special structure. In addition, this proposed transform has more parameters than some known proposed transforms. A lot of simulations show that the independent parameters in the PROJT are able to be used as additional secret keys for image encryption. The rest of this paper is organized as follows. In Section 2, jacket matrices and some preliminaries are recalled. In Section 3, the PROJT is proposed and developed. In Section 4, an efficient algorithm with special structure is developed for the proposed PROJT having many parameters. Examples and computer simulations are given in Section 5. We draw some conclusions and remarks in Section 6.

2 Preliminaries and notations

In this section, we introduce some definitions and notations. For an $N \times N$ matrix $[J]_N$, the $N \times N$ associated matrix $[J]_N^{RT}$ is obtained from matrix $[J]_N$ by taking the reciprocal of each entry and exchanging its row and column indices. In other words, the (k,i) entry of $[J]_N^{RT}$ is equal to the reciprocal of the element in the (i,k) position in $[J]_N$. We now recall the definition of a jacket matrix which is reciprocal-orthogonal in [7].

Definition 2.1. *An $N \times N$ complex matrix $[J]_N = (j_{i,k})$ is called a jacket matrix, if $[J]_N$ is invertible and the element in the entries (i,k) of its inverse matrix is equal to the product of $\frac{1}{N}$ and the inverse of the element in the entries (k,i) of $[J]_N$. In other words, if*

$$[J]_N = \begin{pmatrix} j_{0,0} & j_{0,1} & \cdots & j_{0,N-1} \\ j_{1,0} & j_{1,1} & \cdots & j_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ j_{N-1,0} & j_{N-1,1} & \cdots & j_{N-1,N-1} \end{pmatrix}, \tag{1}$$

then

$$[J]_N^{-1} = \frac{1}{N}[J]^{RT} = \begin{pmatrix} \frac{1}{j_{0,0}} & \frac{1}{j_{1,0}} & \cdots & \frac{1}{j_{N-1,0}} \\ \frac{1}{j_{0,1}} & \frac{1}{j_{1,1}} & \cdots & \frac{1}{j_{N-1,1}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{j_{0,N-1}} & \frac{1}{j_{1,N-1}} & \cdots & \frac{1}{j_{N-1,N-1}} \end{pmatrix}. \tag{2}$$

For example, the well-known Hadamard matrices are jacket matrices. A 2×2 jacket matrix is as follows:

$$[J]_2 = \begin{pmatrix} a & b \\ c & -\frac{bc}{a} \end{pmatrix}, \tag{3}$$

where a, b, c are nonzero complex numbers, since

$$[J]_2\,[J]_2^{RT} = \begin{pmatrix} a & b \\ c & -\frac{bc}{a} \end{pmatrix} \begin{pmatrix} \frac{1}{a} & \frac{1}{c} \\ \frac{1}{b} & -\frac{a}{bc} \end{pmatrix} = 2[I]_2\,. \tag{4}$$

Clearly, this 2×2 jacket matrix has three parameters. The proposed jacket matrix of order 4 [6]

$$[J]_4 = \begin{pmatrix} b & f & g & h \\ c & -\frac{acf}{b} & \frac{acg}{b} & -\frac{ch}{b} \\ d & \frac{adf}{b} & -\frac{adg}{b} & -\frac{dh}{b} \\ e & -\frac{ef}{b} & -\frac{eg}{b} & \frac{eh}{b} \end{pmatrix} \tag{5}$$

has eight parameters and $[J]_4\,[J]_4^{RT} = 4[I]_4$, which is a generalization of the 4×4 center-weighted Hadamard matrix

$$[CWH]_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{6}$$

with only one nonzero complex parameter and $[CWH]_4\,[CWH]_4^{RT} = 4[I]_4$. Bouguezel et al. [13] proposed the following ROP transform of order 8 that is (7) and $[P]_8\,[P]_8^{RT} = 8[I]_8$.

$$[P]_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -a_{1,-1} & a_{2,-1} & -a_{3,-1} & a_{3,-1} & -a_{2,-1} & a_{1,-1} & -1 \\ 1 & a_{1,-1} & -a_{2,-1} & -a_{3,-1} & a_{3,-1} & a_{2,-1} & -a_{1,-1} & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & a_{1,-1} & a_{2,-1} & a_{3,-1} & -a_{3,-1} & -a_{2,-1} & -a_{1,-1} & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -a_{1,-1} & -a_{2,-1} & a_{3,-1} & -a_{3,-1} & a_{2,-1} & a_{1,-1} & -1 \end{pmatrix}. \tag{7}$$

3 Proposed PROJT with many parameters

Before proposing a desired PROJT, we also need to introduce some notations and definitions. For a given integer $N = 2^{r+1}$, any integer $0 \le n \le N-1$ can be written as

$$n = n_r 2^r + n_r 2^{r-1} + \cdots + n_1 2 + n_0 \tag{8}$$

where n_i is 0 or 1 for $0 \le i \le r+1$. For two integers n and $k = k_r 2^r + k_{r-1} 2^{r-1} + \cdots + k_1 2 + k_0$, we define

$$< k, n >= k_r n_r + k_{r-1} n_{r-1} + \cdots + k_1 n_1 + k_0 n_0.$$

Further, we denote the two sets as follows:

$$S_e \equiv \left\{ n \mid 0 \le n \le N-1,\ \sum_{i=0}^{r} n_i \ \text{ is even} \right\}$$

and

$$S_o \equiv \left\{ n \mid 0 \le n \le N-1, \sum_{i=0}^{r} n_i \ \text{ is odd} \right\}.$$

Moreover, let

$$p(n) = \begin{cases} 0, & \text{if } n \in S_e, \\ 1, & \text{if } n \in S_o, \end{cases}$$

and

$$\widetilde{n} = \begin{cases} 0, & \text{if } n \text{ is even}, \\ 1, & \text{if } n \text{ is odd}. \end{cases}$$

Let

$$V^{(i)} = \begin{pmatrix} V_{0,2i} & V_{0,2i+1} \\ V_{1,2i}, & V_{1,2i+1} \end{pmatrix},$$

for $i = 0, 1, 2 \cdots, N/2 - 1 = 2^r - 1$, where $V_{1,2i+1} = -\frac{V_{1,2i} V_{0,2i+1}}{V_{0,2i}}$. Therefore, we have

$$V_{0,2i} V_{1,2i+1} + V_{0,2i+1} V_{1,2i} = 0, \tag{9}$$

where $i = 0, 1, \cdots, N/2 - 1$. Moreover, it is easy to see that $V^{(0)}, V^{(1)}, \cdots, V^{(2^r-1)}$ are 2×2 jacket matrices, each of which has three independent parameters. Let

$$\alpha = \left(a_{0,0}, a_{0,1}, \cdots, a_{0,N/4-1}, a_{0,N/4}, \cdots, a_{0,N/2-1}\right)$$

and

$$\beta = \left(a_{1,0}, a_{1,1}, \cdots, a_{1,N/4-1}, a_{1,N/4}, \cdots, a_{1,N/2-1}\right)$$

where $a_{1,i} = \frac{a_{0,i}}{a_{0,N/2-i-1}} a_{1,N/2-i-1}$ for $i = N/4, N/4 + 1, \cdots, N/2 - 1$.

Based on the above two vectors Bouguezel et al. [13] constructed a class of ROP transforms having $\frac{3}{4}N$ independent parameters for a sequence length $N/2$. The ROP transform of a complex input data $X(k)$ of order $N/2 = 2^r$ can be stated as follows [13]:

$$Y(n) = \sum_{k=0}^{N/2-1} (-1)^{<k,n>} a_{p(n),k} X(k), \tag{10}$$

where $n = 0, 1, \cdots, N/2 - 1$. Further, they proved that the inverse of the ROP transform by (10) is fastly decoded by

$$X(k) = \frac{1}{N} \sum_{n=0}^{N/2-1} \frac{(-1)^{<k,n>}}{a_{p(n),k}} Y(n), \tag{11}$$

where $k = 0, 1, \cdots, N/2 - 1$.

For two $N \times N$ matrices $[A]_N = (a_{ij})$ and $[B]_N = (b_{ij})$, the Hadamard product $[C]_N = (c_{ij})$ of the two matrices $[A]$ and $[B]$ is defined as $c_{ij} = a_{ij} b_{ij}$ and denoted $[C]_N = [A]_N \circ [B]_N$. For the proposed ROP transform, Bouguezel et al. also gave the method on how to construct the transform matrix associated with the ROP transform. Let $[A]_{N/2}$ be the matrix, whose $k-$th row equals to the $k-$th row of $[A]_{N/2}$, to be α if $p(k) = 0$ and to be β if $p(k) = 1$. Moreover, let $[H]_{N/2} = [H]_2 \otimes [H]_2 \otimes \cdots \otimes [H]_2$ be the Hadamard matrix of order $N/2$. Then the proposed ROP matrix $[P]_{N/2}$ is just $[A]_{N/2} \circ [H]_{N/2}$. Therefore, it is easy to see that $[P]_{N/2}$ is orthogonal and

$$[P]_{N/2} [P]_{N/2}^{RT} = [P]_{N/2}^{RT} [P]_{N/2} = N/2 [I]_{N/2},$$

where $[I]_{N/2}$ is the $N/2 \times N/2$ identity matrix. In other words, we have the following equation:

$$\sum_{s=0}^{N/2-1} \frac{(-1)^{<k,s>} a_{p(k),s}}{(-1)^{<l,s>} a_{p(l),s}} = \begin{cases} N/2 & \text{if } k = l \\ 0 & \text{if } k \ne l \end{cases}, \tag{12}$$

where $k, l = 0, 1, \cdots, N/2 - 1$. With the above notations and symbols, we are ready to propose a novel class PROJT as follows.

Definition 3.1. *A PROJT of a complex sequence $X(n)$ of order $N = 2^{r+1}$ is defined as*

$$Y(n) = \sum_{i=0}^{N-1} (-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\widetilde{n},i} X(i), \tag{13}$$

$n = 0, 1, \cdots, N-1$, and $\lfloor \frac{n}{2} \rfloor$ stands for the largest integer no more than $\frac{n}{2}$.

The PROJT matrix of order $N = 2^{r+1}$ have $\frac{9}{4}N$ independent parameters, since there are $\frac{3}{4}N$ nonzero independent parameters $a_{00}, \cdots, a_{0,N/2-1}, a_{1,0}, \cdots, a_{N/4-1}$ and $\frac{3}{2}N$ independent parameters $V_{00}, \cdots, V_{0,N}, V_{1,0}, \cdots, V_{1,N/2-1}$. It is known that the more independent parameters a transform has, the more it has applications. Note that the PROJT has $\frac{9}{4}N$ parameters while the ROP has $\frac{3}{2}N$ parameters, which implies that the PROJT has more applications in watermarking, encryption, and error-correcting codes than the ROP. Further, it can be shown that the inverse transform of the proposed transform is easily obtained and has an efficient algorithm.

Theorem 3.2. *The inverse transform of the PROJT defined by (13) is given as follows:*

$$X(i) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>}}{a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\widetilde{n},i}} Y(n), \tag{14}$$

where $i = 0, 1, \cdots, N-1$.

Proof. We establish the above theorem by proving that the following equation holds

$$\sum_{i=0}^{N-1} \frac{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{t}{2} \rfloor>} a_{p(\lfloor \frac{t}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{t},i}} = \begin{cases} N & \text{if } t = n \\ 0 & \text{if } t \neq n \end{cases}, \quad (15)$$

where $n, t = 0, 1, \cdots, N-1$.

Case 1: We first consider $t = n$. Then

$$\begin{aligned} &\sum_{i=0}^{N-1} \frac{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{t}{2} \rfloor>} a_{p(\lfloor \frac{t}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{t},i}} \\ &= \sum_{i=0}^{N-1} \frac{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} \\ &= \sum_{i=0}^{N-1} 1 \\ &= N. \end{aligned}$$

Case 2. Next, we consider that $t \neq n$. There are four subcases.

Subcase 2.1 $n = 2k$ and $t = 2l$. Then

$$\begin{aligned} &\sum_{i=0}^{N-1} \frac{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{t}{2} \rfloor>} a_{p(\lfloor \frac{t}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{t},i}} \\ &= \sum_{i=0}^{N-1} (-1)^{<k, \lfloor \frac{i}{2} \rfloor>+<l, \lfloor \frac{i}{2} \rfloor>} \frac{a_{p(k), \lfloor \frac{i}{2} \rfloor} V_{0,i}}{a_{p(l), \lfloor \frac{i}{2} \rfloor} V_{0,i}} \\ &= \sum_{i=0, i=2s}^{N-1} (-1)^{<k, \lfloor \frac{i}{2} \rfloor>+<l, \lfloor \frac{i}{2} \rfloor>} \frac{a_{p(k), \lfloor \frac{i}{2} \rfloor}}{a_{p(l), \lfloor \frac{i}{2} \rfloor}} \\ &\quad + \sum_{i=0, i=2s+1}^{N-1} (-1)^{<k, \lfloor \frac{i}{2} \rfloor>+<l, \lfloor \frac{i}{2} \rfloor>} \frac{a_{p(k), \lfloor \frac{i}{2} \rfloor}}{a_{p(l), \lfloor \frac{i}{2} \rfloor}} \\ &= \sum_{s=0}^{N/2-1} (-1)^{<k,s>+<l,s>} \frac{a_{p(k),s}}{a_{p(l),s}} \\ &\quad + \sum_{s=0}^{N/2-1} (-1)^{<k,s>+<l,s>} \frac{a_{p(k),s}}{a_{p(l),s}} \\ &= 0 + 0 = 0, \end{aligned}$$

where the last equality follows from (12).

Subcase 2.2: $n = 2k$ and $t = 2l + 1$. Then

$$\begin{aligned} &\sum_{i=0}^{N-1} \frac{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{t}{2} \rfloor>} a_{p(\lfloor \frac{t}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{t},i}} \\ &= \sum_{i=0}^{N-1} \frac{(-1)^{<k, \lfloor \frac{i}{2} \rfloor>} a_{p(k), \lfloor \frac{i}{2} \rfloor} V_{0,i}}{(-1)^{<l, \lfloor \frac{i}{2} \rfloor>} a_{p(l), \lfloor \frac{i}{2} \rfloor} V_{1,i}} \\ &= \sum_{i=0, i=2s}^{N-1} \frac{(-1)^{<k, \lfloor \frac{i}{2} \rfloor>} a_{p(k), \lfloor \frac{i}{2} \rfloor} V_{0,i}}{(-1)^{<l, \lfloor \frac{i}{2} \rfloor>} a_{p(l), \lfloor \frac{i}{2} \rfloor} V_{1,i}} \\ &\quad + \sum_{i=0, i=2s+1}^{N-1} \frac{(-1)^{<k, \lfloor \frac{i}{2} \rfloor>} a_{p(k), \lfloor \frac{i}{2} \rfloor} V_{0,i}}{(-1)^{<l, \lfloor \frac{i}{2} \rfloor>} a_{p(l), \lfloor \frac{i}{2} \rfloor} V_{1,i}} \\ &= \sum_{s=0}^{N/2-1} \frac{(-1)^{<k,s>} a_{p(k),s} V_{0,2s}}{(-1)^{<l,s>} a_{p(l),s} V_{1,2s}} \\ &\quad + \sum_{s=0}^{N/2-1} \frac{(-1)^{<k,s>} a_{p(k),s} V_{0,2s+1}}{(-1)^{<l,s>} a_{p(l),s} V_{1,2s+1}} \\ &= \sum_{s=0}^{N/2-1} \frac{(-1)^{<k,s>} a_{p(k),s}}{(-1)^{<l,s>} a_{p(l),s}} \left(\frac{V_{0,2s}}{V_{1,2s}} + \frac{V_{0,2s+1}}{V_{1,2s+1}} \right) \\ &= \sum_{s=0}^{N/2-1} \frac{(-1)^{<k,s>} a_{p(k),s}}{(-1)^{<l,s>} a_{p(l),s}} 0, \end{aligned}$$

where the last equality follows from (9).

Subcase 2.3 $n = 2k + 1$ and $t = 2l$; and **Subcase 2. 4** $n = 2k + 1$ and $t = 2l + 1$ are similar to Subcase 2.2 and Subcase 2.1. Hence we omitted the detail. Therefore,

$$\begin{aligned} &\frac{1}{N} \sum_{n=0}^{N-1} \frac{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>}}{a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} Y(n) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \frac{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>}}{a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} \\ &\quad \times \sum_{t=0}^{N-1} (-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{t}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{t}{2} \rfloor} V_{\tilde{n},t} X(t) \\ &= \frac{1}{N} \sum_{t=0}^{N-1} \sum_{n=0}^{N-1} \frac{(-1)^{<\lfloor \frac{t}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{t}{2} \rfloor} V_{\tilde{n},t}}{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} X(t) \\ &= \frac{1}{N} \sum_{n=0,}^{N-1} \frac{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}}{(-1)^{<\lfloor \frac{n}{2} \rfloor, \lfloor \frac{i}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} X(i) \\ &\quad + \frac{1}{N} \sum_{t=0, t \neq i}^{N-1} \sum_{n=0,}^{N-1} \frac{(-1)^{<\lfloor \frac{t}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{t}{2} \rfloor} V_{\tilde{n},t}}{(-1)^{<\lfloor \frac{i}{2} \rfloor, \lfloor \frac{n}{2} \rfloor>} a_{p(\lfloor \frac{n}{2} \rfloor), \lfloor \frac{i}{2} \rfloor} V_{\tilde{n},i}} X(t) \\ &= \frac{1}{N} \sum_{n=0,}^{N-1} X(i) + \frac{1}{N} \sum_{t=0, t \neq i}^{N-1} 0 \\ &= X(i), \end{aligned}$$

where the second last equality follows from (15) and $i = 0, 1, \cdots, N-1$. Hence we finish our proof. □

From the above theorem, the inverse transform of the proposed PROJTs is easily obtained by the reciprocal of the forward matrix. Next, we give the matrix form of the PROJTs. Moreover, this matrix has a simple structure and interesting properties. Assume that the input sequence is the $N \times 1$ vector $X = (x(0), \cdots, x(N-1))^T$ and the output sequence is the $N \times 1$ vector $Y = (y(0), \cdots, y(N-1))^T$, where T denotes the transpose of a vector or matrix. Then the proposed PROJT and its inverse transform can be presented in terms of the matrix form as follows:

$$Y = [J]_N X$$

and

$$X = [J]_N^{-1} Y = \frac{1}{N}[J]_N^{RT} Y.$$

By Theorem 3.2, the inverse matrix $[J]_N^{-1}$ of the inverse transform can be obtained from the forward matrix with the two reciprocal-orthogonal and transpose operation. The (k, m) entry of $[J]_N^{-1}$ equals to the reciprocal of the (m, k) entry of the forward matrix $[J]_N$ up to a scaling factor $\frac{1}{N}$. Hence the inverse matrix $[J]_N^{-1}$ can be obtained by the following operation from the forward matrix $[J]_N$. First, the matrix $[J]_N^R$ is obtained by taking the reciprocal of each entry of $[J]_N$. Second, the matrix $([J]_N^R)^T$ is obtained by transposing $[J]_N^R$. Last, $[J]_N^{-1}$ is obtained by the product scaling factor $\frac{1}{N}$ and the matrix $\left([J]_N^R\right)^T = [J]_N^{RT}$. In other words,

$$\left([J]_{N\times N}\right)^{-1} = \frac{1}{N}\left(\frac{1}{p_{mk}}\right)_{N\times N}. \tag{16}$$

In order to understand the proposed PROJT, we give some examples to illustrate how to construct it.

Example 1. For $N = 4 = 2^{1+1}$ and $r = 1$. Since the indices of the row $[H]_2$ are 0 and 1, we have $S_e = \{0\}$ and $S_o = \{1\}$. Let $[A]_2$ be the 2×2 matrix

$$[A]_2 = \begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & \frac{a_{0,1}a_{1,0}}{a_{0,0}} \end{pmatrix}.$$

Then the 2×2 forward matrix $[P]_2$ is

$$[P]_2 = [A]_2 \circ [H]_2 = \begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & -\frac{a_{0,1}a_{1,0}}{a_{0,0}} \end{pmatrix}.$$

Let

$$V^{(0)} = \begin{pmatrix} V_{0,0} & V_{0,1} \\ V_{1,0} & -\frac{V_{1,0}V_{0,1}}{V_{0,0}} \end{pmatrix}$$

and

$$V^{(1)} = \begin{pmatrix} V_{0,2} & V_{0,3} \\ V_{1,2} & -\frac{V_{1,2}V_{0,3}}{V_{0,2}} \end{pmatrix}.$$

Then, the forward matrix associated with the PROJT is (17).

$$z[J]_4 = \begin{pmatrix} a_{0,0}V_{0,0} & a_{0,0}V_{0,1} & a_{0,1}V_{0,2} & a_{0,1}V_{0,3} \\ a_{0,0}V_{1,0} & -\frac{a_{0,0}V_{0,1}V_{1,0}}{V_{0,0}} & a_{0,1}V_{1,2} & -\frac{a_{0,1}V_{0,3}V_{1,2}}{V_{0,2}} \\ a_{1,0}V_{0,0} & a_{0,0}V_{0,1} & -\frac{a_{0,1}a_{1,0}V_{0,2}}{a_{0,0}} & -\frac{a_{0,1}a_{1,0}V_{0,3}}{a_{0,0}} \\ a_{1,0}V_{1,0} & -\frac{a_{1,0}V_{0,1}V_{0,1}}{V_{0,0}} & -\frac{a_{0,1}a_{1,0}V_{1,2}}{a_{0,0}} & \frac{a_{0,1}a_{1,0}V_{1,2}V_{0,3}}{a_{0,0}V_{0,2}} \end{pmatrix} \tag{17}$$

Example 2. For $N = 8 = 2^{2+1}$ and $r = 2$. Since $0 = 0 \cdot 2^2 + 0 \cdot 2^1 + 0$, $1 = 0 \cdot 2^2 + 0 \cdot 2 + 1$, $2 = 0 \cdot 2^2 + 1 \cdot 2 + 0$, $3 = 0 \cdot 2^2 + 1 \cdot 2 + 1$, we have $S_e = \{0, 3\}$ and $S_o = \{1, 2\}$. Let $[A]_2$ be the 4×4 matrix obtained by α and β and their indices, i.e.,

$$[A]_4 = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & \frac{a_{0,2}a_{1,1}}{a_{01}} & \frac{a_{0,3}a_{1,0}}{a_{0,0}} \\ a_{1,0} & a_{1,1} & \frac{a_{0,2}a_{1,1}}{a_{01}} & \frac{a_{0,3}a_{1,0}}{a_{0,0}} \\ a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \end{pmatrix}.$$

So the 4×4 forward ROP matrix associated with the ROP transform is

$$[P]_4 = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & -a_{1,1} & \frac{a_{0,2}a_{1,1}}{a_{01}} & -\frac{a_{0,3}a_{1,0}}{a_{0,0}} \\ a_{1,0} & a_{1,1} & -\frac{a_{0,2}a_{1,1}}{a_{01}} & -\frac{a_{0,3}a_{1,0}}{a_{0,0}} \\ a_{0,0} & -a_{0,1} & -a_{0,2} & a_{0,3} \end{pmatrix}.$$

Moreover,

$$V^{(0)} = \begin{pmatrix} V_{0,0} & V_{0,1} \\ V_{1,0} & -\frac{V_{1,0}V_{0,1}}{V_{0,0}} \end{pmatrix},$$

$$V^{(1)} = \begin{pmatrix} V_{0,2} & V_{0,3} \\ V_{1,2} & -\frac{V_{1,2}V_{0,3}}{V_{0,2}} \end{pmatrix},$$

$$V^{(3)} = \begin{pmatrix} V_{0,4} & V_{0,5} \\ V_{1,4} & -\frac{V_{1,4}V_{0,5}}{V_{0,4}} \end{pmatrix},$$

$$V^{(4)} = \begin{pmatrix} V_{0,6} & V_{0,7} \\ V_{1,6} & -\frac{V_{1,6}V_{0,7}}{V_{0,6}} \end{pmatrix}.$$

Hence we obtain the proposed PROJT involving matrix $[J]_8$ (see (18))

$$[J]_8 = \begin{pmatrix} a_{0,0}V_{0,0} & a_{0,0}V_{0,1} & a_{0,1}V_{0,2} & a_{0,1}V_{0,3} & a_{0,2}V_{0,4} & a_{0,2}V_{0,5} & a_{0,3}V_{0,6} & a_{0,3}V_{0,7} \\ a_{0,0}V_{1,0} & -\frac{a_{0,0}V_{0,1}V_{1,0}}{V_{00}} & a_{0,1}V_{1,2} & -\frac{a_{0,1}V_{0,3}V_{1,2}}{V_{0,2}} & a_{0,2}V_{1,4} & -\frac{a_{0,2}V_{0,5}V_{1,4}}{V_{0,4}} & a_{0,3}V_{1,6} & -\frac{a_{0,3}V_{0,7}V_{1,6}}{V_{0,6}} \\ a_{1,0}V_{0,0} & a_{1,0}V_{0,1} & -a_{1,1}V_{0,2} & -a_{1,1}V_{0,3} & \frac{a_{0,2}a_{1,1}V_{0,4}}{a_{0,1}} & \frac{a_{0,2}a_{1,1}V_{0,5}}{a_{0,1}} & -\frac{a_{0,3}a_{1,0}V_{0,6}}{a_{0,0}} & -\frac{a_{0,3}a_{1,0}V_{0,7}}{a_{0,0}} \\ a_{1,0}V_{1,0} & -\frac{a_{1,0}V_{0,1}V_{1,0}}{V_{0,0}} & -a_{1,1}V_{1,2} & \frac{a_{1,1}V_{1,2}V_{0,3}}{V_{0,2}} & \frac{a_{0,2}a_{1,1}V_{1,4}}{a_{0,1}} & -\frac{a_{0,2}a_{1,1}V_{0,5}V_{1,4}}{a_{0,1}V_{0,4}} & -\frac{a_{0,3}a_{1,0}V_{1,6}}{a_{0,0}} & \frac{a_{0,3}a_{1,0}V_{0,7}V_{1,6}}{a_{0,0}V_{0,6}} \\ a_{1,0}V_{0,0} & a_{1,0}V_{0,1} & a_{1,1}V_{0,2} & a_{1,1}V_{0,3} & -\frac{a_{0,2}a_{1,1}V_{0,4}}{a_{0,1}} & -\frac{a_{0,2}a_{1,1}V_{0,5}}{a_{0,1}} & -\frac{a_{0,3}a_{1,0}V_{0,6}}{a_{0,0}} & -\frac{a_{0,3}a_{1,0}V_{0,7}}{a_{0,0}} \\ a_{1,0}V_{1,0} & -\frac{a_{1,0}V_{0,1}V_{1,0}}{V_{0,0}} & a_{1,1}V_{1,2} & -\frac{a_{1,1}V_{1,2}V_{0,3}}{V_{0,2}} & -\frac{a_{0,2}a_{1,1}V_{1,4}}{a_{0,1}} & \frac{a_{0,2}a_{1,1}V_{0,5}V_{1,4}}{a_{0,1}V_{0,4}} & -\frac{a_{0,3}a_{1,0}V_{1,6}}{a_{0,0}} & \frac{a_{0,3}a_{1,0}V_{0,7}V_{1,6}}{a_{0,0}V_{0,6}} \\ a_{0,0}V_{0,0} & a_{0,0}V_{0,1} & -a_{0,1}V_{0,2} & -a_{0,1}V_{0,3} & -a_{0,2}V_{0,4} & -a_{0,2}V_{0,5} & a_{0,3}V_{0,6} & a_{0,3}V_{0,7} \\ a_{0,0}V_{1,0} & -\frac{a_{0,0}V_{0,1}V_{1,0}}{V_{00}} & -a_{0,1}V_{1,2} & \frac{a_{0,1}V_{0,3}V_{1,2}}{V_{0,2}} & -a_{0,2}V_{1,4} & \frac{a_{0,2}V_{0,5}V_{1,4}}{V_{0,4}} & a_{0,3}V_{1,6} & -\frac{a_{0,3}V_{0,7}V_{1,6}}{V_{0,6}} \end{pmatrix} \quad (18)$$

4 Fast and efficient algorithm for the proposed PROJT

In this section, we analyze some properties of the proposed PROJT, which are used to present an efficient algorithm for a fast computation of the proposed PROJT by (13) and (14).

In order to give an algorithm, we may rewrite (13) in another form so that we can analyze the PROJT by splitting the summation in (13) into a sum of two summations and for even and odd indices. For every output vector components we have

$$\begin{aligned} Y(2k) &= \sum_{i=0}^{N-1}(-1)^{<k,\lfloor\frac{i}{2}\rfloor>}a_{p(k),\lfloor\frac{i}{2}\rfloor}V_{0,i}X(i) \\ &= \sum_{i=0,i=2t}^{N-1}(-1)^{<k,\lfloor\frac{i}{2}\rfloor>}a_{p(k),\lfloor\frac{i}{2}\rfloor}V_{0,i}X(i) \\ &\quad + \sum_{i=0,i=2t+1}^{N-1}(-1)^{<k,\lfloor\frac{i}{2}\rfloor>}a_{p(k),\lfloor\frac{i}{2}\rfloor}V_{0,i}X(i) \\ &= \sum_{t=0}^{N/2-1}(-1)^{<k,t>}a_{p(k),t}V_{0,2t}X(2t) \\ &\quad + \sum_{t=0}^{N/2-1}(-1)^{<k,t>}a_{p(k),t}V_{0,2t+1}X(2t+1) \\ &= \sum_{t=0}^{N/2-1}(-1)^{<k,t>}a_{p(k),t}(V_{0,2t}X(2t) \\ &\quad + V_{0,2t+1}X(2t+1)), \end{aligned}$$

where $k = 0, 1, \ldots, N/2-1$. Similarly, for odd output vector components, we have $Y(2k+1) =$

$$\sum_{t=0}^{N/2-1}(-1)^{<k,t>}a_{p(k),t}\left(V_{1,2t}X(2t) + V_{1,2t+1}X(2t+1)\right),$$

where $k = 0, 1, \ldots, N/2-1$. Now let

$$Z^{(0)}(t) = V_{0,2t}X(2t) + V_{0,2t+1}X(2t+1) \quad (19)$$

and

$$Z^{(1)}(t) = V_{1,2t}X(2t) + V_{1,2t+1}X(2t+1). \quad (20)$$

where $t = 0, 1, \ldots, N/2-1$. Now we are able to analyze the fast and efficient algorithm for the proposed ROP transform. Since the set $\{0, 1, \ldots, N/2-1\}$ can be split into two sets S_e and S_o, using the result of the paper [13], we are able to obtain

$$\begin{aligned} Y(2k) &= \sum_{t=0}^{N/2-1}(-1)^{<k,t>}a_{p(k),t}Z^{(0)}(t) \\ &= \sum_{i=0}^{N/4-1}(-1)^{<k,i>}\Big(a_{0,i}Z^{(0)}(i) \\ &\quad + a_{0,N/2-i-1}Z^{(0)}(N/2-i-1)\Big) \end{aligned}$$

for $k \in S_e$ and $1 \le k \le N/2-1$. Similarly, we obtain

$$\begin{aligned} Y(2k) &= \sum_{i=0}^{N/4-1}(-1)^{<k,i>}\Big(a_{1,i}Z^{(0)}(i) \\ &\quad - a_{1,N/2-i-1}Z^{(0)}(N/2-i-1)\Big) \end{aligned}$$

for $k \in S_o$ and $1 \le k \le N/2-1$.

Let

$$\begin{aligned} \begin{pmatrix} f_e^{(0)}(i) \\ f_o^{(0)}(i) \end{pmatrix} &= \begin{pmatrix} a_{0,i} & a_{0,N/2-i-1} \\ a_{1,i} & -a_{1,N/2-i-1} \end{pmatrix} \\ &\quad \times \begin{pmatrix} Z^{(0)}(i) \\ Z^{(1)}(N/2-i-1) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & \frac{a_{1,i}}{a_{0,i}} \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} a_{0,i} & 0 \\ 0 & a_{0,N/2-i-1} \end{pmatrix} \\ &\quad \times \begin{pmatrix} Z^{(0)}(i) \\ Z^{(1)}(N/2-i-1) \end{pmatrix}. \end{aligned}$$

Then,

$$Y(2k) = \begin{cases} \sum_{i=0}^{N/4-1}(-1)^{<k,i>}f_e^{(0)}, & k \in S_e \\ \sum_{i=0}^{N/4-1}(-1)^{<k,i>}f_o^{(0)}, & k \in S_e. \end{cases}$$

Hence all the output points can be obtained from the output sequence of the WHT of order of $N/4$ of the input sequence.

The $N \times N$ proposed matrix with the PROJT can be stated as follows:

$$[W]_N = ([P]_{N/2} \otimes [I]_2)\, diag(V^{(0)}, \ldots, V^{(N/2-1)})$$

where $[I]_2$ is a 2×2 identity matrix and $diag\left(V^{(0)}, \ldots, V^{(N/2-1)}\right)$ is the $N \times N$ block diagonal matrix whose block is 2×2 matrices.

Let $[Q]^{(1)}_{N/2}$ be an $N/2 \times N/2$ permutation matrix whose entries $(0,0), (1, N/2-1), (2,1), (3, N/2-2), (4,2), \ldots, (2i, i), (2i+1, N/2-i-1), \ldots, (N/2-2, N/4-1), (N/2-1, N/2-(N/4-1)-1)$ are 1, the other entries are 0.

Let $[Q]^{(2)}_{N/2}$ be an $N/2 \times N/2$ permutation matrix whose entries $(0,0), (1,2), (2,4), \ldots, (i, 2i), \ldots, (N/4-1, N/2-2), (N/4, 1), (N/4+1, 3), \ldots, (N/4+i, 2i+1), \ldots, (N/4+N/4-1, N/2-1)$ are 1, the other entries are 0.

Let $n = n_r 2^r + n_{r-1}2^{r-1} + \cdots + n_1 2^1 + n_0$, $n = 0, 1, \ldots, N-1, N = 2^{r+1}$. Define

$$S^{(1)}_e = \left\{n, \mid p(n) = 0, 0 \le n \le N/4 - 1\right\}$$

$$S^{(2)}_e = \left\{n, \mid p(n) = 0, N/4 \le n \le N/2 - 1\right\}$$

$$S^{(1)}_o = \left\{n, \mid p(n) = 1, 0 \le n \le N/4 - 1\right\}$$

$$S^{(2)}_o = \left\{n, \mid p(n) = 1, N/4 \le n \le N/2 - 1\right\}.$$

Let $[Q]^{(3)}_{N/2}$ be an $N/2 \times N/2$ permutation matrix whose entries (n, n) for $n \in S^{(1)}_e \bigcup S^{(2)}_o$, $(n, n+N/4)$ for $n \in S^{(1)}_o$ and $(n, n-N/4)$ for $n \in S^{(2)}_e$ are 1, the other entries are 0. Let

$$A^{(i)} = \begin{pmatrix} a_{0,i} & a_{0,N/2-i-1} \\ a_{1,i} & -a_{1,N/2-i-1} \end{pmatrix}, i = 0, 1, \ldots, N/4 - 1$$

where $a_{1,N/2-i-1} = \frac{a_{0,N/2-i-1}a_{1,i}}{a_{0,i}}$.

Then the ROP matrix by the ROP transform [13] can be decomposed to

$$[P]_{N/2} = [Q]^{(3)}_{N/2} \left([I]_2 \otimes H_{N/4}\right) [Q]^{(2)}_{N/2} \times diag\left(A^{(0)}, A^{(1)}, \ldots A^{(N/4-1)}\right) [Q]^{(1)}_{N/2}.$$

Hence, the proposed PROJT is as follows:

$$[W]_N = \left([P]_{N/2} \otimes [I]_2\right) \times diag\left(V^{(0)}, \ldots, V^{(N/2-1)}\right).$$

Therefore, the decomposition of the proposed transform is the following:

$$[W]_N = \left(\left([Q]^{(3)}_{N/2}([I]_2 \otimes H_{N/4})[Q]^{(2)}_{N/2} \times diag\left(A^{(0)}, A^{(1)}, \ldots, A^{(N/4-1)}\right)[Q]^{(1)}_{N/2}\right) \otimes [I]_2\right) \times diag\left(V^{(0)}, \ldots, V^{(N/2-1)}\right).$$

By the mean of this decomposition of the proposed matrix, we are able to get a fast and efficient algorithm.

Example 3. Let $N = 2^{r+1} = 4$. Then

$$[Q]^{(4)}_4 = \begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{pmatrix}.$$

$$[D]^{(4)}_2 = \begin{pmatrix} 1 & 0 \\ 0 & \frac{a_{1,0}}{a_{0,0}} \end{pmatrix}, \; [D]^{(3)}_2 = \begin{pmatrix} a_{0,0} & 0 \\ 0 & a_{0,1} \end{pmatrix}.$$

$$[D]^{(2)}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{V_{1,0}}{V_{0,0}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{V_{1,2}}{V_{0,2}} \end{pmatrix}$$

$$[D]^{(4)}_1 = \begin{pmatrix} V_{0,0} & 0 & 0 & 0 \\ 0 & V_{0,1} & 0 & 0 \\ 0 & 0 & V_{0,2} & 0 \\ 0 & 0 & 0 & V_{0,3} \end{pmatrix}.$$

$[Q]^{(3)}_2 = [Q]^{(2)}_2 = [Q]^{(1)}_2 = [I]_2$, implies that the equation (21)

$$[W]_4 = [Q]^{(4)T}_4 \left([I]_2 \otimes \left([Q]^{(3)}_2 ([I]_2 \otimes H_1)[Q]^{(2)}_2 [D]^{(4)}_2 ([I]_1 \otimes [H]_2)[D]^{(3)}_2 [Q]^{(1)}_2\right)\right)[Q]^{(4)}_4 [D]^{(2)}_4 ([I]_2 \otimes [H]_2)[D]^{(1)}_4 \tag{21}$$

Example 4. Let $N = 2^{r+1} = 8$. Then

$$[Q]^{(1)}_4 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix}$$

$$[Q]^{(2)}_4 = \begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{pmatrix}$$

$$[Q]^{(3)}_4 = \begin{pmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \end{pmatrix}$$

Hence we have the following equation (22):

$$[W]_8 = \left(\left([Q]^{(3)}_4([I]_2 \otimes [H]_2)[Q]^{(2)}_4 \, diag\left(A^{(0)}, A^{(1)}\right)[Q]^{(1)}_4\right) \otimes [I]_2\right) diag\left(V^{(0)}, V^{(1)}, V^{(2)}, V^{(3)}\right) \tag{22}$$

Example 5. Let $N = 16 = 2^{r+1}$. Let

$$[Q]_{16/2}^{(1)} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0 \end{pmatrix}$$

$$[Q]_{16/2}^{(2)} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}$$

$$[Q]_{16/2}^{(3)} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}$$

Hence we have the equation (23).

$$[W]_{16} = \left(\left([Q]_8^{(3)}([I]_2 \otimes [H]_4)[Q]_8^{(2)} diag\left(A^{(0)}, A^{(1)}, A^{(2)}, A^{(3)}\right) [Q]_8^{(1)}\right) \otimes [I]_2\right) diag\left(V^{(0)}, V^{(1)}, V^{(2)}, \ldots, V^{(7)}\right) \tag{23}$$

Moreover, let $[D]_{N/2}^{(4)}$ be an $N/2 \times N/2$ diagonal matrix whose diagonal entries are $1, \frac{a_{1,0}}{a_{0,0}}, 1, \frac{a_{1,1}}{a_{0,1}}, \ldots, 1, \frac{a_{1,i}}{a_{0,i}}, \ldots, 1, \frac{a_{1,N/4-1}}{a_{0,N/4-1}}$. Let $[D]_{N/2}^{(3)}$ be an $N/2 \times N/2$ diagonal matrix whose diagonal entries are $a_{0,0}, a_{0,N/2-1}, a_{0,1}, a_{0,N/2-2}, \ldots, a_{0,i}, a_{0,N/2-i-1}, \ldots, a_{0,N/4-1}, a_{0,N/4}$. Let $[D]_N^{(2)}$ be an $N \times N$ diagonal matrix whose diagonal entries are $1, \frac{V_{1,0}}{V_{0,0}}, 1, \frac{V_{1,2}}{V_{0,2}}, \ldots, 1, \frac{V_{1,2i}}{V_{0,2i}}, \ldots, 1, \frac{V_{1,N-2}}{V_{0,N-2}}$. Let $[D]_N^{(1)}$ be an $N \times N$ diagonal matrix whose diagonal entries are $V_{0,0}, V_{0,1}, V_{0,2}, \ldots, V_{1,N-2}, V_{0,N-1}$.

Let $[Q]_N^{(4)}$ be the $N \times N$ matrix whose entries $(0,0)$, $(1,2), (3,6), \ldots, (i,2i), \ldots, (N/2-1, N-2), (N/2,1)$, $(N/2+1,3), \ldots, (N/2+i, 2i+1), \ldots, (N-1, N-1)$ are 1 and the other entries are 0. Then

$$[P]_{N/2} \otimes [I]_2 = ([Q]_N^{(4)})^T \left([I]_2 \otimes [P]_{N/2}\right) [Q]_N^{(4)}.$$

Hence the fast decomposed factor of the proposed PROJT can be presented as (24).

$$[W]_N = \left([Q]_N^{(4)}\right)^T \left([I]_2 \otimes \left([Q]_{N/2}^{(3)} ([I]_2 \otimes [H]_{N/4})[Q]_{N/2}^{(2)} [D]_{N/2}^{(4)} ([I]_{N/4} \otimes [H]_2)[D]_{N/2}^{(3)} [Q]_{N/2}^{(1)}\right)\right) [Q]_N^{(4)} [D]_N^{(2)} \left([I]_{N/2} \otimes [H]_2\right) [D]_N^{(1)} \tag{24}$$

Moreover, the fast flows of orders 4 and 8 are presented in Figures 1 and 2.

Now we compute the complexity of this decomposed factor of the proposed PROJT. Since the permutation matrices do not need any additions and multiplications, we do not consider permutations in computing the

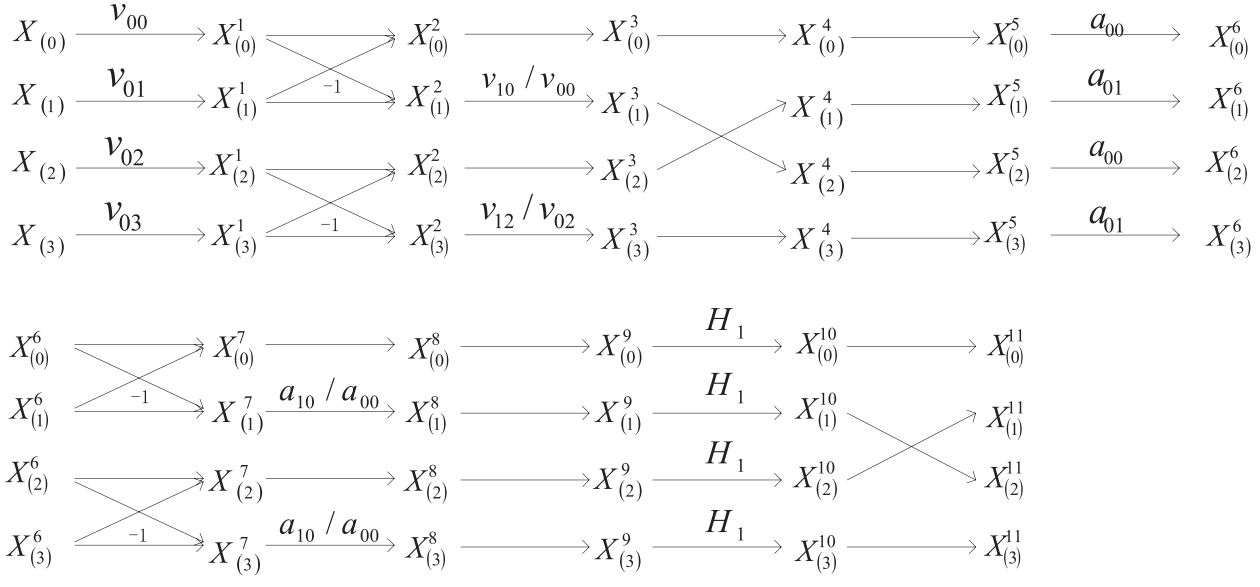

Figure 1 Fast flow of order $N = 2^2$. This figure depicts the fast flow of order $N = 2^2$.

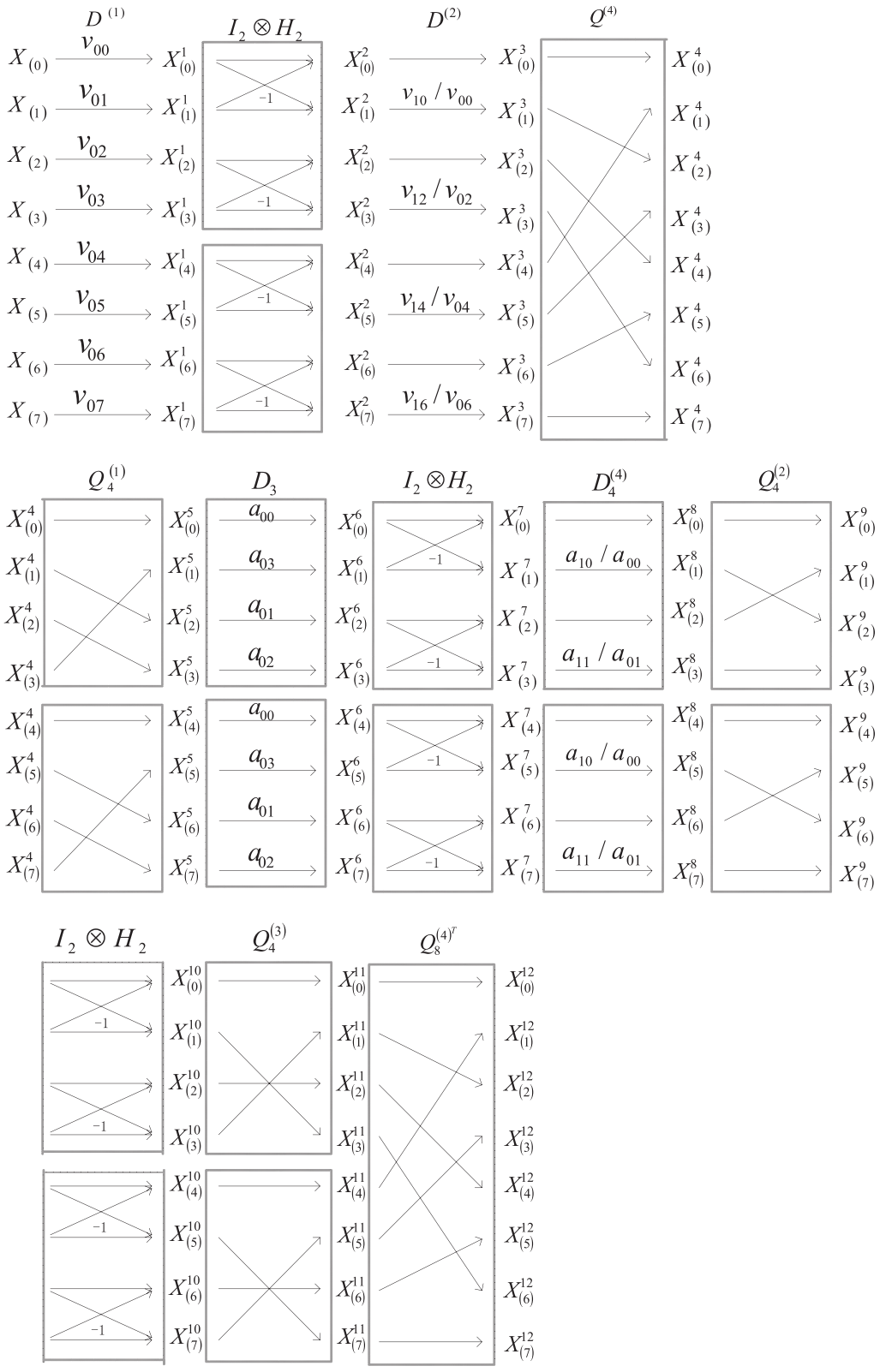

Figure 2 Fast flow of order $N = 2^3$. This figure depicts the fast flow of order $N = 2^3$.

Table 1 Computation complexity of additions and multiplications in conventional DFT [1], reciprocal-orthogonal parametric [10], EIJT [18] $N = 3 \times 2^r$, and proposed PROJT with $N = 2^{r+1}$ matrix size, where $r \geq 2$

	Conventional DFT [1]	ROP [13] $N = 2^{r+1}$	EIJT [18] $N = 3 \times 2^r$	Proposed $N = 2^{r+1}$
Parameters	1 (fixed)	$3N/2$	$2N-1$	$\frac{9}{4}N$
Additions	$(N-1)N$	$N\log_2 N$	$3N\log_2 N$	$N + N\log_2 N$
Multiplications	N^2	$3N/2$	$N\log_2 N$	$3N$

complex of the proposed PROJT. Firstly, for the diagonal matrix $[D]_N^{(1)}$, there are needs for N multiplications. Since $[H]_2$ needs 2 additions, $[I]_{N/2} \otimes [H]_2$ needs $N/2 \times 2 = N$ additions. Note that there are $N/2$ main diagonal elements which may be equal to 1. Then there are needs for $N/2$ multiplications for the the diagonal matrix $[D]_N^{(2)}$. Hence there are needs for $N + N/2$ multiplications and N additions for the matrix $[D]_N^{(2)}([I]_{N/2} \otimes [H]_2)[D]_N^{(1)}$. Since $[D]_{N/2}^{(3)}$ is the $N/2 \times N/2$ diagonal matrix, there are needs for $N/2$ multiplications. However $[D]_{N/2}^{(4)}$ is the $N/2 \times N/2$ diagonal matrix in which there are at least $N/4$ 1's on the main diagonal, so only $N/4$ multiplications are needed. Clearly, $([I]_{N/4} \otimes [H]_2$ needs $2 \times N/4 = N/2$ additions. $[I]_2 \otimes [H]_{N/4}$ needs $2 \times (N/4)\log_2(N/4)$ additions. Therefore, $\left([Q]_{N/2}^{(3)}\ ([I]_2 \otimes [H]_{N/4})\right)[Q]_{N/2}^{(2)}[D]_{N/2}^{(4)} \times \left([I]_{N/4} \otimes [H]_2\right)[D]_{N/2}^{(3)}[Q]_{N/2}^{(1)}$ needs $N/2 + N/4 = 3N/4$ multiplications and $N/4 \times 2 + 2\left(N/4\log_2(N/4) = N/2\log_2 N - N/2\right.$ additions. Hence $\left(\left([Q]_N^{(4)}\right)^T\right.$ $\left([I]_2 \otimes \left([Q]_{N/2}^{(3)}\left([I]_2 \otimes [H]_{N/4}\right)[Q]_{N/2}^{(2)}[D]_{N/2}^{(4)} \times \left([I]_{N/4} \otimes [H]_2\right)[D]_{N/2}^{(3)}[Q]_{N/2}^{(1)}\right)\right)[Q]_N^{(4)}$ needs $3N/2$ multiplications and $N\log_2 N - N$ additions. In sum, the decomposed factor of the proposed PROJT needs $3N/2 + 3N/2 = 3N$ multiplication and $2N + N\log_2 N - N = N\log_N + N$ addition operations. Table 1 shows that the complexity of the proposed PROJT is better than conventional DFT, but a litter less than the ROP. However, the PROJT has a larger number of parameters than the ROP.

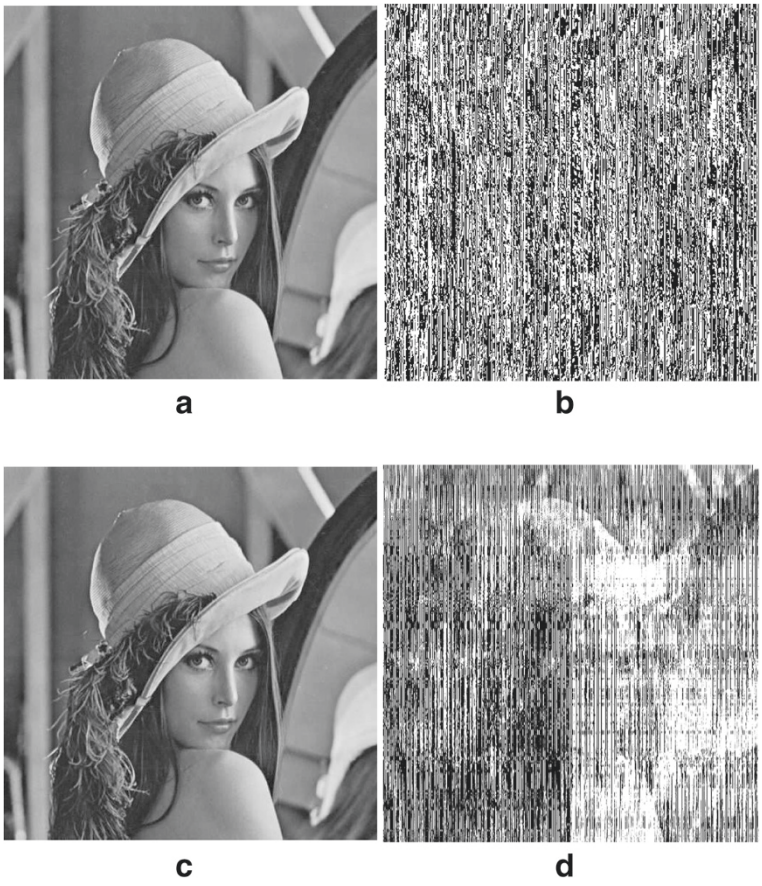

Figure 3 The PROJT encryption and decryption. (a) Original image. **(b)** Encrypted image. **(c)** Decrypted image with the correct parameters. **(d)** Decrypted image with one minimized error.

5 Simulations and discussion

In all of the following computer experiments, we use the proposed PROJT and inverse PROJT for encryptions and decryptions. Let $V_{0,2i}$, $V_{0,2i+1}$, $V_{1,2i}$, $a_{0,i}$ for $i = 0, 1, \ldots, N/2 - 1$ and $a_{1,j}$ for $j = 0, 1, N/4 - 1$ denote the $\left(2 + \frac{1}{4}\right) N$ independent parameters employed in the encryption process. Let $V'_{0,2i}$, $V'_{0,2i+1}$, $V'_{1,2i}$, $a'_{0,i}$ for $i = 0, 1, \ldots, N/2 - 1$ and $a'_{1,j}$ for $j = 0, 1, N/4 - 1$ denote the $\left(2 + \frac{1}{4}\right) N$ parameters employed in the decryption process. Figure 3a shows the 512×512 original image to be encrypted, whose elements are integers in the range 0-255. Figure 3b shows the magnitude image of its encryption output using the proposed PROJT. Then, we use the correct parameter vectors for decryption, and the decrypted output is shown in Figure 3c, which is the same as the original image. To give a decryption example of the previous encrypted image with one single wrong parameter of the $\left(2 + \frac{1}{4}\right) N$ independent parameters, we use

$$\begin{aligned} V'_{0,2i} &= V_{0,2i} + \delta \bmod 255, \\ V'_{0,2i+1} &= V_{0,2i+1}, \\ V'_{1,2i} &= V_{1,2i}, \\ a'_{0,i} &= a_{0,i}, \end{aligned} \tag{25}$$

where $i = 0, 1, \ldots, N/2 - 1$ and

$$a'_{1,j} = a_{1,j}, \tag{26}$$

where $j = 0, 1, N/4 - 1$. Error δ can be any integer uniformly distributed in the range 1-255. Figure 3d shows the decrypted image with the minimized error $\delta = 1$, which shows that the original image is successfully protected.

Therefore, to successfully decrypt the 512×512 ($N = 512$) image, whose elements are integers in the range 0-255, we need to know every parameter of $V_{0,2i}$, $V_{0,2i+1}$, $V_{1,2i}$, $a_{0,i}$ for $i = 0, 1, \ldots, N/2 - 1$ and $a_{1,j}$ for $j = 0, 1, N/4 - 1$. Intuitively, this is also true for $N > 512$. Assume that the errors in $V'_{0,2i}$, $V'_{0,2i+1}$, $V'_{1,2i}$, $a'_{0,i}$ for $i = 0, 1, \ldots, N/2 - 1$ and $a'_{1,j}$ for $j = 0, 1, N/4 - 1$ are uniformly distributed. Then, the probability of a successful decryption without knowing all the $\left(2 + \frac{1}{4}\right) N = 1152$ parameters correctly is $1/256^{1152}$.

6 Conclusions

In this paper, we have proposed a new class of PROJT of order $N = 2^{r+1}$ independent $3N$ parameters. The PROJT is based on the the proposed ROP transform [13], block diagonal matrices, and permutations. On one hand, the critical usefulness of the PROJT generalized the proposed ROP transform, which is a special case of the the PROJT. On the other hand, the PROJT has more parameters than the ROP transform. What is more important, some nice properties are presented, in particular, the inverse transform is fastly obtained by the reciprocal and transpose operations. With the aid of matrix decomposition and Kronecker product approach, a fast and efficient algorithm for computing the proposed PROJT is obtained. In fact, we show that the PROJT has $3N$ multiplications and $N + N \log_2 N$ addition operations. Therefore, the proposed PROJT can be employed in watermaking and encryption where the independent parameters can be used as an additional secret key.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
The authors would like to thank the reviewers for their insightful suggestions and comments for improving the manuscript.
This work was supported by the MEST 2012-002521, National Research Foundation, Korea. It was also supported partly by the National Nature Science Foundation of China (11271256, 61201249) and the DHU Distinguished Young Professor Program (14D210402).

Author details
[1]Division of Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, Korea. [2]Department of Mathematics, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China. [3]School of Information Science and Technology, Donghua University, Shanghai 201620, China.

Received: 14 March 2014 Accepted: 10 September 2014
Published: 29 September 2014

References

1. N Ahmed, KR Rao, *Orthogonal Transforms for Digital Signal Processing*. (Springer, New York, 1975)
2. RK Yarlagadda, JE Hershey, *Hadamard Matrix Analysis and Synthesis With Applications to Communications Signal/Image Processing*. (Kluwer Academic Publishers, Norwell, MA, 1997)
3. KJ Horadam, *Hadamard Matrices and Their Applications*. (Princeton University Press, Princeton, NJ, 2006)
4. S-C Pei, W-L Hsue, The multiple-parameter discrete fractional Fourier transform. IEEE Signal Process. Lett. **13**(6), 329–332 (2006)
5. C-C Tseng, Eigenvalues and eigenvectors of generalized DFT, generalized DHT, DCT-IV and DST-IV matrices. IEEE Signal Process. **50**(4), 866–877 (2002)
6. MH Lee, The center weighted Hadamard transform. IEEE Trans. Circuits Syst. **36**(9), 1247–1249 (1989)
7. MH Lee, A new reverse jacket transform and its fast algorithm. IEEE Trans. Circ. Syst. II. **47**(1), 39–47 (2000)
8. JJ Ding, SC Pei, PH Wu, Jacket Haar, Transform, in *Circuits and Systems (ISCAS) 2011 IEEE International Symposium*, (15-18 May 2011), pp. 1520–1523
9. MH Lee, X-D Zhang, Fast block center weighted Hadamard transform. IEEE Trans. Circuits Syst. I: Reg. Papers. **54**(12), 2741–2745 (2007)
10. S Agaian, K Tourshan, JP Noonan, Parametric Slant-Hadamard transforms with applications. IEEE Signal Process. Lett. **9**(11), 375–377 (2002)
11. S Agaian, K Tourshan, JP Noonan, Generalized parametric Slant-Hadamard transforms with applications. Signal Process. Lett. **84**(8), 1299–1306 (2004)
12. S Bouguezel, A reciprocal-orthogonal parametric transform and its fast algorithm. IEEE Signal Process. Lett. **19**(11), 769–772 (2012)
13. S Bouguezel, MO Ahmad, MNS Swamy, A new class of reciprocal-orthogonal parametric transforms. IEEE Trans. Circuits Syst. I, Reg. Papers. **56**(4), 795–804 (2009)
14. S Bouguezel, MO Ahmad, MNS Swamy, New parametric discrete Fourier and Hartley transforms, and algorithms for fast computation. IEEE Trans. Circuits Syst. I, Reg. Papers. **58**(3), 562–575 (2011)
15. JJ Ding, S-C Pei, P-H Wu, Arbitrary-length Walsh-Jacket transforms, in *Proceedings of 2011 APSIPA Annual Summit and Conference* (Xi'an, China, 18-21 October 2011), pp. 1–10

16. S Minasyan, J Astola, D Guevorkian, An image compression scheme based on parametric, Haar-like transform, in *Circuits and Systems 2005. ISCAS 2005. IEEE International Symposium*, (23-26 May 2005), pp. 2088–2091
17. JM Vilardy, JE Calderon, CO Torres, L Mattos, Digital images phase encryption using fractional Fourier transform, in *Electronics, Robotics and Automotive Mechanics Conference*, vol. 1, (26-29 Sept 2006), pp. 15–18
18. MH Lee, X-D Zhang, W Song, X-G Xia, Fast reciprocal jacket transform with many parameters. IEEE Trans. Circuits Syst. I, Reg. Papers. **59**(7), 1472–1481 (2012)
19. MH Lee, A new reverse, Jacket transform based on Hadamard matrix, in *IEEE International Symposium on Information Theory* (Sorrento, Italy, 25-30 June 2000), p. 471
20. MH Lee, YL Borissov, On Jacket transforms over finite fields, in *IEEE International Symposium on Information Theory* (Seoul, Korea, 28 June 2009-3 July 2009), pp. 2803–2807
21. Z Chen, MH Lee, G Zeng, Fast cocylic jacket transform. IEEE Trans. Signal Process. **56**(5), 2143–2148 (2008)
22. MH Lee, *Jacket Matrices: Construction and its Applications for Fast Cooperative Wireless Signal Processing*. (Lambert Academic Publishing, Lambert, Germany, 2012)

doi:10.1186/1687-6180-2014-149
Cite this article as: Lee *et al.*: Fast parametric reciprocal-orthogonal jacket transforms. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:149.