Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 163758, 12 pages
doi:10.1155/2010/163758

Research Article

A General Approach for Orthogonal 4-Tap Integer Multiwavelet Transforms

Mingli Jing,[1,2] Hua Huang,[1] Wuling Liu,[3] and Chun Qi[1]

[1] *School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2] *School of Statistics, Xi'an University of Finance and Economics, Xi'an 710061, China*
[3] *School of Information Science and Engineering, Southeast University, Nanjing 211189, China*

Correspondence should be addressed to Mingli Jing, mljing2008@yahoo.cn

Received 12 December 2009; Accepted 11 May 2010

Academic Editor: Angelo Luongo

An algorithm for orthogonal 4-tap integer multiwavelet transforms is proposed. We compute the singular value decomposition (SVD) of block recursive matrices of transform matrix, and then transform matrix can be rewritten in a product of two block diagonal matrices and a permutation matrix. Furthermore, we factorize the block matrix of block diagonal matrices into triangular elementary reversible matrices (TERMs), which map integers to integers by rounding arithmetic. The cost of factorizing block matrix into TERMs does not increase with the increase of the dimension of transform matrix, and the proposed algorithm is in-place calculation and without allocating auxiliary memory. Examples of integer multiwavelet transform using DGHM and CL are given, which verify that the proposed algorithm is an executable algorithm and outperforms the existing algorithm for orthogonal 4-tap integer multiwavelet transform.

1. Introduction

In many applications of image processing, the given data are integer valued. To compress digital image losslessly by means of transformation, the transform must map integers to integers and be perfectly invertible. Calderbank et al. [1] presented two wavelet transforms approaches that map integers to integers, and the proposed lifting scheme has been a popular method for mapping integers to integers. Using lifting scheme, Deever and Hemami [2] presented a projection-based technique for decreasing the first-order entropy of transform coefficients. However, Multiwavelets have several advantages in comparison to scalar wavelets. Such features as short support, orthogonality, symmetry, and vanishing moments are known to be important in signal processing. A scalar wavelet cannot possess all these properties at the time.

Unlike scalar wavelet, little literature is available on integer multiwavelet transform. Cheung et al. [3] presented an integer multiwavelet transform and its associated integer prefilter based on box-and–slope multiscaling system. Van Fleet [4] developed a factorization of 4-tap multiwavelets with multiplicity 2. In particular, the author in [4] derived a factorization of the DGHM multiwavelet and applied DGHM multiwavelet processing to data compression. In order to implement the DGHM integer transform, the author computed scaling value α under certain conditions [4].The scaling value α is not convenient to compute and larger than 1, which turns out to be bad for lossless compression [1].

This paper is to present an algorithm of 4-tap orthogonal multiwavelets that map integers to integers and be perfectly invertible using SVD and TERMs. Unlike [4], the proposed algorithm does not compute scaling value α and calculates in-place and without allocating auxiliary memory.

2. Integer Transform and Multiwavelet

2.1. Integer Transform

Hao and Shi [5] proved that there exists an implementation of integer mapping when a linear transform is invertible and in finite-dimensional space, as follows.

Theorem 2.1. *Matrix A has a TERM factorization of $A = PV_1V_2\cdots V_M D_R$ if and only if $|\det(A)| = 1$, where M is finite, V_k $(k = 1, 2 \ldots M)$ are unit TERMs, P is a permutation matrix, and D_R is a rotator for only one complex number. If all the diagonal elements of a TERM are equal to 1, the TERM will be a unit triangular matrix. If A is a real matrix, then D_R is an identity matrix.*

Unless otherwise stated, we will concentrate on the nonsingular real matrix A, thus reversible integer implementation for general linear transforms $Y = AX$ can be derived as follows [5], where $Y = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$, $X = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$.

Let $L = (l_{i,j})_{n\times n}$, $U = (u_{i,j})_{n\times n}$, and $P_{n\times n}$ be a lower TERM, an upper TERM, and a permutation matrix, respectively. If $A = L$ is a lower TERM, the computational ordering of linear transform $Y = AX$ can be arranged to be top-down:

$$y_i = \begin{cases} l_{i,i}x_i, & i = 1, \\ l_{i,i}x_i + \left[\displaystyle\sum_{k=1}^{i-1} l_{i,k}x_k\right], & 2 \le i \le n. \end{cases} \tag{2.1}$$

Its inverse ordering is reversed:

$$x_i = \begin{cases} \dfrac{y_i}{l_{i,i}}, & i = 1, \\ \left(\dfrac{1}{l_{i,i}}\right)\left(y_i - \left[\displaystyle\sum_{k=1}^{i-1} l_{i,k}x_k\right]\right), & 2 \le i \le n. \end{cases} \tag{2.2}$$

Likewise, if $A = U$ is an upper TERM, the computational ordering of linear transform $Y = AX$ can be arranged to be top-down:

$$y_i = \begin{cases} u_{i,i}x_i, & i = n, \\ u_{i,i}x_i + \left[\displaystyle\sum_{k=i+1}^{n} u_{i,k}x_k\right], & 1 \le i \le n-1. \end{cases} \tag{2.3}$$

Its inverse ordering is reversed:

$$x_i = \begin{cases} \dfrac{y_i}{u_{i,i}}, & i = n, \\ \left(\dfrac{1}{u_{i,i}}\right)\left(y_i - \left[\displaystyle\sum_{k=i+1}^{n} u_{i,k}x_k\right]\right), & 1 \le i \le n-1, \end{cases} \tag{2.4}$$

where [ ] denotes rounding arithmetic. Since permutation matrix $P_{n\times n}$ is an orthogonal matrix, transform and the inverse transform can be denoted by $Y = P_{n\times n}X$ and $X = P_{n\times n}^{\mathrm{T}}Y$, respectively.

2.2. Properties of Transform Matrix

While many researchers have investigated multiwavelet of multiplicity r [6–9], little work has been published on integer multiwavelet transforms. We will discuss integer transform on 4-tap multiwavelet of multiplicity r. Let $\Phi(t) = [\phi_1, \dots \phi_r]^{\mathrm{T}}$, $\Psi(t) = [\psi_1, \dots \psi_r]^{\mathrm{T}} \in L^2(R)^r$ be a multiscaling function and a multiwavelet function, respectively, and then both $\Phi(t)$ and $\Psi(t)$ satisfy the following two-scale relation:

$$\Phi(t) = \sum_{k=0}^{3} P_k \Phi(2t-k), \qquad \Psi(t) = \sum_{k=0}^{3} Q_k \Phi(2t-k), \tag{2.5}$$

where P_k and Q_k are $r \times r$ matrices, respectively. For more details about multiwavelets, see, for example, [7–9]. As is the case in [4], the matrix representation $H_{n\times n}$ of multiwavelet transformation is

$$H_{n\times n} = \begin{bmatrix} H_0 & H_1 & & & \\ & H_0 & H_1 & & \\ & & \ddots & \ddots & \\ & & & H_0 & H_1 \\ H_1 & & & & H_0 \end{bmatrix}, \tag{2.6}$$

where $H_i = \left[\begin{smallmatrix} P_{2i} & P_{2i+1} \\ Q_{2i} & Q_{2i+1} \end{smallmatrix}\right]$, $i = 0, 1, P_j, Q_j, \quad j = 0, 1, 2, 3$, are $r \times r$ matrices, respectively.

In order to find integer multiwavelet transform schemes, inspired by [1], we will prove some remarkable properties of orthogonal matrix H and then present the approach that maps integers to integers.

Property 1. Suppose that $\operatorname{rank}(H_0) = m$, $\operatorname{rank}(H_1) = s$, thus $m + s = 2r$.

Proof. From the condition of orthogonality, $H^{\mathrm{T}}H = HH^{\mathrm{T}} = I_{n\times n}$. It follows that

$$\begin{gathered} H_0H_0^{\mathrm{T}} + H_1H_1^{\mathrm{T}} = I_{2r\times 2r}, \qquad H_1H_0^{\mathrm{T}} = H_0H_1^{\mathrm{T}} = 0, \\ 2r = \operatorname{rank}(I) = \operatorname{rank}\left(H_0H_0^{\mathrm{T}} + H_1H_1^{\mathrm{T}}\right) \le \operatorname{rank}(H_0) + \operatorname{rank}(H_1), \\ 0 = \operatorname{rank}\left(H_0H_1^{\mathrm{T}}\right) \ge \operatorname{rank}(H_0) + \operatorname{rank}(H_1) - 2r. \end{gathered} \tag{2.7}$$

Thus, $\operatorname{rank}(H_1) + \operatorname{rank}(H_0) = 2r$. The proof is completed. □

Let $U_0\Sigma_0V_0^{\mathrm{T}}$ and $U_1\Sigma_1V_1^{\mathrm{T}}$ be the singular value decomposition of H_0,H_1, respectively. By Property 1, since $\operatorname{rank}(H_1) = \operatorname{rank}(\Sigma_1)$, $\operatorname{rank}(H_0) = \operatorname{rank}(\Sigma_0)$, we see that $\operatorname{rank}(\Sigma_0) + \operatorname{rank}(\Sigma_1) = 2r$.

Property 2. If the λ_i are the singular values of H_0 and the vectors u_0^i and v_0^i are the ith left singular vector and ith right singular vector, respectively, and the μ_i are the singular values of H_1 and the vectors u_1^j and v_1^j are the jth left singular vector and ith right singular vector, respectively, then u_0^i and v_0^i are orthogonal to the vectors u_1^j and v_1^j, respectively, $i = 1, 2, \ldots m$, $m = \operatorname{rank}(H_0)$, $j = 1, 2 \ldots s$, $s = \operatorname{rank}(H_1)$.

Proof. From (2.7), we have

$$H_1H_0^{\mathrm{T}} = U_1\Sigma_1V_1^{\mathrm{T}}V_0\Sigma_0U_0^{\mathrm{T}} = 0. \tag{2.8}$$

Since U_0, U_1 are two orthogonal matrices, we have $\Sigma_1V_1^{\mathrm{T}}V_0\Sigma_0 = 0$,

$$\begin{aligned} \Sigma_1V_1^{T}V_0\Sigma_0 &= \Sigma_1\left[v_1^1,\ldots,v_1^{2r}\right]^{T}\left[v_0^1,\ldots,v_0^{2r}\right]\Sigma_0 \\ &= \begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_m & \\ & & & 0 \end{bmatrix} \begin{bmatrix} \left(v_1^1\right)^{\mathrm{T}} \\ \left(v_1^2\right)^{\mathrm{T}} \\ \vdots \\ \left(v_1^{2r}\right)^{\mathrm{T}} \end{bmatrix} \left[v_0^1 \ v_0^2, \ldots, \ v_0^{2r}\right] \begin{bmatrix} \mu_1 & & & \\ & \ddots & & \\ & & \mu_s & \\ & & & 0 \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1\left(v_1^1\right)^{\mathrm{T}} \\ \vdots \\ \lambda_m\left(v_1^m\right)^{\mathrm{T}} \\ 0 \end{bmatrix} \left[\mu_1v_0^1, \ldots, \ \mu_sv_0^s \ \ 0\right] \end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} \lambda_1\mu_1\langle v_1^1, v_0^1\rangle & \cdots & \lambda_1\mu_s\langle v_1^1, v_0^s\rangle & 0 \\ \vdots & & \vdots & \vdots \\ \lambda_m\mu_1\langle v_1^m, v_0^1\rangle & \cdots & \lambda_m\mu_s\langle v_1^m, v_0^s\rangle & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} \\
&= 0.
\end{aligned} \tag{2.9}$$

From (2.9), we have $\langle v_0^i, v_1^j\rangle = 0$. Likewise, $\langle u_0^i, u_1^j\rangle = 0$. The proof is completed. □

By Property 2, we can rewrite the matrices H_0 and H_1 in this case as

$$\begin{aligned}
H_0 &= U_0\Sigma_0 V_0^{\mathrm{T}} \\
&= \left[u_0^1, u_0^2, \ldots, u_0^m, u_1^1, u_1^2, \ldots, u_1^s\right] \\
&\quad \times \operatorname{diag}\left(\lambda_1, \ldots, \lambda_m, \underbrace{0\cdots 0}_{s}\right)\left[v_0^1, v_0^2, \ldots, v_0^m, v_1^1, v_1^2, \ldots, v_1^s\right]^{\mathrm{T}} \\
&= \overline{U}\Sigma_0\overline{V}^{\mathrm{T}},
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
H_1 &= U_1\Sigma_1 V_1^{\mathrm{T}} \\
&= \left[u_0^1, u_0^2, \ldots, u_0^m, u_1^1, u_1^2, \ldots, u_1^s\right] \operatorname{diag}\left(\underbrace{0, \ldots, 0}_{m}, \mu_1, \ldots, \mu_s\right) \\
&\quad \times \left[v_0^1, v_0^2, \ldots, v_0^m, v_1^1, v_1^2, \ldots, v_1^s\right]^{\mathrm{T}} \\
&= \overline{U}\,\overline{\Sigma}_1\overline{V}^{\mathrm{T}},
\end{aligned} \tag{2.11}$$

where $\overline{U} = [u_0^1, u_0^2, \ldots, u_0^m, u_1^1, u_1^2, \ldots, u_1^s]$, $\overline{V} = [v_0^1, v_0^2, \ldots, v_0^m, v_1^1, v_1^2, \ldots, v_1^s]$ $\sum_0 = \operatorname{diag}(\lambda_1, \ldots, \lambda_m, \underbrace{0\cdots 0}_{s})$ $\overline{\sum}_1 = \operatorname{diag}(\underbrace{0\cdots 0}_{m}, \mu_1, \ldots, \mu_s)$. From Property 2, we see that $\overline{U}$ and $\overline{V}$ are two orthogonal matrices.

Property 3. If the λ_i and μ_j are the nonzero singular values of H_0 and H_1, respectively, then the nonzero singular values of H_0 and H_1 are 1.

Proof. From (2.7), we can derive

$$\begin{aligned}
I &= H_0H_0^{\mathrm{T}} + H_1H_1^{\mathrm{T}} \\
&= \overline{U}\Sigma_0\overline{V}^{\mathrm{T}}\overline{V}\Sigma_0\overline{U}^{\mathrm{T}} + \overline{U}\,\overline{\Sigma}_1\overline{V}^{\mathrm{T}}\overline{V}\,\overline{\Sigma}_1\overline{U}^{\mathrm{T}} \\
&= \overline{U}\left(\Sigma_0^2 + \overline{\Sigma}_1^2\right)\overline{U}^{\mathrm{T}}.
\end{aligned} \tag{2.12}$$

Left multiplying (2.12) with $\overline{U}^{-1}$ and right multiplying (2.11) with $(\overline{U}^{\mathrm{T}})^{-1}$, we have $I = \Sigma_0^2 + \overline{\Sigma}_1^2 = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_m^2, \mu_1^2, \ldots, \mu_s^2)$. The proof is completed. □

By Properties 1 and 2, we can rewrite the matrix $H_{n\times n}$ in this case as

$$
\begin{aligned}
H_{n\times n} &= \begin{bmatrix} H_0 & H_1 & & \\ & H_0 & H_1 & \\ & & \ddots & \\ H_1 & & & H_0 \end{bmatrix} = \begin{bmatrix} \overline{U} & & & \\ & \overline{U} & & \\ & & \ddots & \\ & & & \overline{U} \end{bmatrix} \begin{bmatrix} \Sigma_0 & \overline{\Sigma}_1 & & \\ & & \Sigma_0 & \overline{\Sigma}_1 \\ \overline{\Sigma}_1 & & & \Sigma_0 \end{bmatrix} \begin{bmatrix} \overline{V}^{\mathrm{T}} & & & \\ & \overline{V}^{\mathrm{T}} & & \\ & & \ddots & \\ & & & \overline{V}^{\mathrm{T}} \end{bmatrix} \\
&= U\overline{P}V^{\mathrm{T}},
\end{aligned}
\tag{2.13}
$$

where $U_{n\times n}$ and $V_{n\times n}^{\mathrm{T}}$ are block diagonal with the same $2r \times 2r$ orthogonal matrices $\overline{U}$, $\overline{V}^{\mathrm{T}}$ comprising the diagonal, respectively, and $\overline{P}_{n\times n}$ is a permutation matrix. The advantages of the re-representation of transform $H_{n\times n}$ are as follows:

(1) the TERMs factorization of $H_{n\times n}$ directly requires more calculation than $\overline{U}$ and $\overline{V}$, because the TERMs factorization of $H_{n\times n}$ directly involves $o(n^2)$ operations [5], and the TERMs factorization of $\overline{U}(\overline{V})$ involves $o((2r)^2)$ operations; in general, $\overline{U}(\overline{V})$ is 4 by 4 matrices with the case $r = 2$;

(2) integer multiwavelet transform can map integers to integers in-place and without allocating auxiliary memory (see Section 2.1).

3. Multiwavelet Integer Transform

Since $\overline{U}$, $\overline{V}$ are two $2r \times 2r$ orthogonal matrices, we suppose that the input data $\{x_1, x_2, \ldots, x_n\}$ is denoted by $\{[x_1^{1,j}, x_2^{1,j}, \ldots x_{2r}^{1,j}]^{\mathrm{T}}, [x_1^{2,j}, x_2^{2,j}, \ldots x_{2r}^{2,j}]^{\mathrm{T}}, \ldots, [x_1^{L,j}, x_2^{L,j}, \ldots x_{2r}^{L,j}]^{\mathrm{T}}\}$, that is, $\{X^{1,j}, X^{2,j}, \ldots X^{L,j}\}$, where L is a positive integer, the superscript j denotes the jth stages of the transform of $X^{k,j}$, $k = 1, 2, \ldots L$, and k is the kth sequence of input data. When $\{X^{k,j}\}_{k=1}^{L}$ is an initial input data, j equals 0. We can state an algorithm for 4-tap orthogonal integer multiwavelet transform as follows.

(1) Compute the singular value decomposition of H_0, H_1, and find $\overline{U}, \overline{P} = [\Sigma_0|\overline{\Sigma}_1]_{2r\times 4r}$, $\overline{V}^{\mathrm{T}}$.

(2) Find the factorization of $\overline{U}$, $\overline{V}^{\mathrm{T}}$, such as $\overline{U} = P_0V_1V_2,\ldots,V_M$, $V^{\mathrm{T}} = P_1W_1W_2,\ldots,W_M$, respectively.

(3) Combine the data sequence $\{X^{k,j}\}_{k=1}^{L}$ with the matrices $W_M, \ldots W_1, P_1, \overline{P}, V_M, \ldots V_1, P_0$ by corresponding equations in Section 2.1, sequentially, according to transform matrix form (upper, lower, and permutation matrix).

Specially speaking, when j equals $M+1$, $\overline{P}$ is a $2r \times 4r$ matrix, so the ordering of transform can be carried through

$$X^{k,j+1} = \begin{cases} \Sigma_0 X^{k,j} + \overline{\Sigma}_1 X^{k+1,j}, & k = 1,2\ldots(L-1), \\ \Sigma_0 X^{k,j} + \overline{\Sigma}_1 X^{1,j}, & k = L, \end{cases} \tag{3.1}$$

where $X^{k,j}$ is a $2r \times 1$ matrix.

The jth stage of the transform is a product of $X^{k,j}$ and the jth matrix of the matrices $W_M, \ldots W_1, P_1, \overline{P}, V_M, \ldots V_1, P_0$ from left to right by rounding arithmetic (see Section 2.1). Since $X^{k,j}$ is integer valued, $X^{k,j+1}$ is also integer valued. After $j = M + M + 3$ stage of the transform, each stage of this process can map integers to integers, therefore transform results of the initial data sequence $\{X^{k,j}\}_{k=1}^{L}$ are integer valued.

The inverse transform must be the backward running of the forward transformation.

4. Experimental Results

In order to demonstrate the availability of the proposed scheme, DGHM [6] and CL [7] multiwavelets are used as examples. The preprocessing is given by balancing the nonbalanced multiwavelet. The way to achieve this is to find the orthogonal matrix R, such that $R^{\mathrm{T}} P_i R = \overline{P}_i$ and $R^{\mathrm{T}} Q_i R = \overline{Q}_i$ [10]. Since R is an orthogonal matrix, the new matrices $\overline{H}_0 = D^{\mathrm{T}} H_0 D$ and $\overline{H}_1 = D^{\mathrm{T}} H_1 D$ inherit all of the properties of H_0 and H_1 in Section 2.2, where $D = \operatorname{diag}(R, R)$. We can compute the SVD of matrices $\overline{H}_0$ and $\overline{H}_1$, respectively, thus we will have two matrices $\overline{U}$ and $\overline{V}^{\mathrm{T}}$.

For the DGHM case, the corresponding matrices in (2.5) are

$$\begin{gathered} P_0 = \begin{bmatrix} \dfrac{3\sqrt{2}}{10} & \dfrac{4}{5} \\ \dfrac{-1}{20} & \dfrac{-3\sqrt{2}}{20} \end{bmatrix}, \qquad P_1 = \begin{bmatrix} 0 & 0 \\ \dfrac{9}{20} & \dfrac{-3\sqrt{2}}{20} \end{bmatrix}, \\ P_2 = \begin{bmatrix} \dfrac{3\sqrt{2}}{10} & 0 \\ \dfrac{9}{20} & \dfrac{\sqrt{2}}{2} \end{bmatrix}, \qquad P_3 = \begin{bmatrix} 0 & 0 \\ \dfrac{-1}{20} & 0 \end{bmatrix}, \\ Q_0 = \begin{bmatrix} \dfrac{\sqrt{3}}{20} & \dfrac{3\sqrt{6}}{20} \\ 0 & 0 \end{bmatrix}, \qquad Q_1 = \begin{bmatrix} \dfrac{-9\sqrt{3}}{20} & \dfrac{\sqrt{6}}{6} \\ 0 & -\dfrac{\sqrt{3}}{3} \end{bmatrix}, \\ Q_2 = \begin{bmatrix} \dfrac{3\sqrt{3}}{20} & \dfrac{-\sqrt{6}}{20} \\ \dfrac{3\sqrt{6}}{10} & -\dfrac{\sqrt{3}}{5} \end{bmatrix}, \qquad Q_3 = \begin{bmatrix} \dfrac{-\sqrt{3}}{60} & 0 \\ \dfrac{-\sqrt{6}}{30} & 0 \end{bmatrix}. \end{gathered} \tag{4.1}$$

By orthogonal matrix R [11],

$$R = \frac{1}{\sqrt{6}}\begin{bmatrix} 1-\sqrt{2} & 1+\sqrt{2} \\ 1+\sqrt{2} & -1+\sqrt{2} \end{bmatrix}, \tag{4.2}$$

we can find the matrices $\overline{U}$, $\overline{V}^{\mathrm{T}}$, and $\overline{P}$ as follows:

$$\overline{U} = \begin{bmatrix} 0.0157912 & 0.1188553 & -0.989179 & -0.084551 \\ -0.6383476 & -0.5786332 & -0.121839 & 0.4927993 \\ 0.7295663 & -0.6631722 & -0.0805546 & 0.1464466 \\ 0.2449407 & 0.4596299 & -0.013821 & 0.8535535 \end{bmatrix},$$
$$\overline{V}^{\mathrm{T}} = \begin{bmatrix} 0 & 0.3445041 & -0.8684815 & -0.3564502 \\ 0 & -0.0719555 & 0.3541483 & -0.932417 \\ -0.2841407 & -0.8974428 & -0.3325729 & -0.0570605 \\ 0.9587826 & -0.2659623 & -0.0985599 & -0.0169102 \end{bmatrix}, \tag{4.3}$$

$\overline{P} = [\Sigma_0|\overline{\Sigma}_1]$, where $\Sigma_0 = \mathrm{diag}(1,1,1,0)$, $\overline{\Sigma}_1 = \mathrm{diag}(0,0,0,1)$, and then compute the TERMs factorization of $\overline{U}$, $\overline{V}^{\mathrm{T}}$ as follows:

$$\begin{aligned}
\overline{U} &= P_0 V_1 V_2, \dots, V_8 \\
&= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.5042912 & 1 & 0 & 0 \\ -0.0209632 & -0.1149459 & 1 & 0 \\ 0.1175034 & -0.5887366 & 0.2706307 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.2704337 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -0.1464467 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1.3706774 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1.1715728 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

$$\times\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & -2.1827521 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1.8454857 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\times\begin{bmatrix}1 & -0.9089951 & -0.1104144 & 0.2007310\\ 0 & 1 & 1.3487059 & -0.5166513\\ 0 & 0 & 1 & 0.0161923\\ 0 & 0 & 0 & 1\end{bmatrix},$$

$$\overline{V}^{\mathrm{T}} = P_1W_1W_2,\dots,W_8$$

$$=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -0.2533664 & 1 & 0 & 0\\ -0.0209632 & -0.3528807 & 1 & 0\\ 0.0031685 & 0.0737051 & -0.3098048 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\times\begin{bmatrix}1 & 0 & 0 & 0\\ -0.0412174 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -0.067583 & 1\end{bmatrix}\begin{bmatrix}1 & -1.0429893 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & -1.0724815\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\times\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -2.0683494 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & -1.9360232 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\times\begin{bmatrix}1 & -0.2773958 & -0.1027969 & -0.0176372\\ 0 & 1 & 1.4389279 & 0.4693591\\ 0 & 0 & 1 & 0.3798175\\ 0 & 0 & 0 & 1\end{bmatrix}. \tag{4.4}$$

Likewise, for the CL case, the corresponding matrices in (2.5) are

$$P_0=\begin{bmatrix}\dfrac{10-3\sqrt{10}}{40} & \dfrac{5\sqrt{6}-2\sqrt{15}}{40}\\ \dfrac{5\sqrt{6}-3\sqrt{15}}{40} & \dfrac{5-3\sqrt{10}}{40}\end{bmatrix},\qquad P_1=\begin{bmatrix}\dfrac{30+3\sqrt{10}}{40} & \dfrac{5\sqrt{6}-2\sqrt{15}}{40}\\ -\dfrac{5\sqrt{6}+7\sqrt{15}}{40} & \dfrac{15-3\sqrt{10}}{40}\end{bmatrix},$$

$$P_2 = \begin{bmatrix} \dfrac{30+3\sqrt{10}}{40} & -\dfrac{5\sqrt{6}-2\sqrt{15}}{40} \\ \dfrac{5\sqrt{6}+7\sqrt{15}}{40} & \dfrac{15-3\sqrt{10}}{40} \end{bmatrix}, \qquad P_3 = \begin{bmatrix} \dfrac{10-3\sqrt{10}}{40} & -\dfrac{5\sqrt{6}-2\sqrt{15}}{40} \\ -\dfrac{5\sqrt{6}-3\sqrt{15}}{40} & \dfrac{5-3\sqrt{10}}{40} \end{bmatrix},$$

$$Q_0 = \begin{bmatrix} \dfrac{5\sqrt{6}-2\sqrt{15}}{40} & -\dfrac{10-3\sqrt{10}}{40} \\ -\dfrac{5-3\sqrt{10}}{40} & \dfrac{5\sqrt{6}-3\sqrt{15}}{40} \end{bmatrix}, \qquad Q_1 = \begin{bmatrix} -\dfrac{5\sqrt{6}-2\sqrt{15}}{40} & \dfrac{30+3\sqrt{10}}{40} \\ \dfrac{15-3\sqrt{10}}{40} & \dfrac{5\sqrt{6}+7\sqrt{15}}{40} \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} -\dfrac{5\sqrt{6}-2\sqrt{15}}{40} & -\dfrac{30+3\sqrt{10}}{40} \\ -\dfrac{15-3\sqrt{10}}{40} & \dfrac{5\sqrt{6}+7\sqrt{15}}{40} \end{bmatrix}, \qquad Q_3 = \begin{bmatrix} \dfrac{5\sqrt{6}-2\sqrt{15}}{40} & \dfrac{10-3\sqrt{10}}{40} \\ \dfrac{5-3\sqrt{10}}{40} & \dfrac{5\sqrt{6}-3\sqrt{15}}{40} \end{bmatrix}. \tag{4.5}$$

By matrix R [11],

$$R = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{4.6}$$

matrices $\overline{U}$, $\overline{V}^T$, and $\overline{P}$ can be represented as

$$\overline{U} = \begin{bmatrix} -0.7104132 & -0.6924131 & -0.0902343 & 0.087948 \\ 0.087948 & -0.0902343 & -0.6924132 & -0.7104133 \\ 0.0902343 & 0.087948 & -0.7104132 & 0.6924132 \\ 0.6924132 & -0.7104133 & 0.087948 & 0.0902343 \end{bmatrix},$$
$$\overline{V}^{\mathrm{T}} = \begin{bmatrix} 0.1132659 & 0 & -0.9935647 & 0 \\ 0 & -0.1132659 & 0 & -0.9935648 \\ -0.9935647 & -0.0000001 & -0.1132659 & 0.0000001 \\ 0.0000001 & -0.9935647 & 0 & 0.1132659 \end{bmatrix}, \tag{4.7}$$

$\overline{P} = [\Sigma_0|\overline{\Sigma}_1]$, where $\Sigma_0 = \operatorname{diag}(1,1,0,0)$, $\overline{\Sigma}_1 = \operatorname{diag}(0,0,1,1)$, and then compute the TERMs factorization of $\overline{U}$, $\overline{V}^{\mathrm{T}}$ as follows:

$$\overline{U} = P_0 V_1 V_2 \cdots V_8$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3.3822942 & 1 & 0 & 0 \\ -0.1270166 & 0 & 1 & 0 \\ -0.4296076 & 0.1270166 & -1.4329687 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.7104132 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1.7104132 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1.4076315 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1.4076314 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0.0161332 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -0.015877 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.9746625 & 0.1270166 & -0.1237984 \\ 0 & 1 & -1.0161332 & 0.8633704 \\ 0 & 0 & 1 & -0.9746627 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\overline{V}^{\mathrm{T}} = P_1 W_1 W_2, \ldots, W_8$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2.0064769 & 1 & 0 & 0 \\ -0.1139995 & 0 & 1 & 0 \\ -0.2287374 & 0.1139995 & -2.0064769 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.9935647 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1.9935648 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1.006477 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1.0064769 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -0.0128291 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.1139995 & 0 \\ 0 & 1 & -1.012996 & -0.1139994 \\ 0 & 0 & 1 & -0.0000001 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0.0129958 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.8}$$

In order to test the effectiveness of the proposed scheme, the effectiveness for lossless compression is measured using the weighted entropy given by [1]. For comparison, we will focus on 4-tap multiwavelets, such as CL, DGHM, and two scalar wavelets: the Daubechies four-tap orthogonal wavelet D4 and the Daubechies biorthogonal wavelet 9-7. We use seven test images, which are 512×512 pixels in size, and we decompose each image one level with each transform. The weighted entropies are tabulated in Table 1.

These results suggest that our method outperforms the method of [4], because the method of [4] enlarges the dynamic range of the output data by scaling value α. It is also worth noting that the performance of CL is very close to that of 9-7. Since preprocessing of the input data is a crucial point in nonbalanced multiwavelet applications [8], we used order-one balanced multiwavelets by balancing the nonbalanced multiwavelet [10]. The focus of this paper is to develop an algorithm of integer multiwavelet transform rather than construct several balanced multiwavelets by orthogonal matrix R, and then a further improvement in the balanced order is needed.

Table 1: Weighted entropies of transformed images.

	Lena	Boat	Baboon	Man	Couple	Plane	Peppers
DGHM	5.2228	5.3093	6.4926	5.7900	5.6304	4.9220	5.4094
CL	5.1822	5.2550	6.4581	5.7415	5.6082	4.8689	5.3871
DGHM [4]	8.0183	7.8442	8.3686	8.2470	8.0604	7.4840	8.1571
D4 [1]	5.7388	5.7754	6.8012	6.2416	6.0685	5.4416	5.8374
9-7 [1]	5.0731	5.1545	6.3975	5.6822	5.5098	4.7099	5.3542

5. Conclusions

This paper presented a novel algorithm for orthogonal 4-tap integer multiwavelet transforms. Unlike the existing algorithm, transform matrix can be rewritten in a product of two block diagonal matrices and a permutation matrix, and then we factorize block matrix of block diagonal matrices into TERMs, which is lower than the TERMs factorization of $H_{n\times n}$ directly in computational cost. In addition, the proposed algorithm computes in-place and without allocating auxiliary memory. Better results could perhaps be achieved with higher balanced order and also with other multiwavelet bases.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 60972142) and 973 program (Project no. 2010CB327904).

References

[1] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, 1998.

[2] A. T. Deever and S. S. Hemami, "Lossless image compression with projection-based and adaptive reversible integer wavelet transforms," *IEEE Transactions on Image Processing*, vol. 12, no. 5, pp. 489–499, 2003.

[3] K.-W. Cheung, C.-O. Cheung, and L.-M. Po, "Novel multiwavelet-based integer transform for lossless image coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '99)*, vol. 1, pp. 444–447, Kobe, Japan, October 1999.

[4] P. J. Van Fleet, "Multiwavelets and integer transforms," *Journal of Computational Analysis and Applications*, vol. 5, no. 1, pp. 161–178, 2003.

[5] P. Hao and Q. Shi, "Matrix factorizations for reversible integer mapping," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2314–2324, 2001.

[6] G. C. Donovan, J. S. Geronimo, D. P. Hardin, and P. R. Massopust, "Construction of orthogonal wavelets using fractal interpolation functions," *SIAM Journal on Mathematical Analysis*, vol. 27, no. 4, pp. 1158–1192, 1996.

[7] C. K. Chui and J.-A. Lian, "A study of orthonormal multi-wavelets," *Applied Numerical Mathematics*, vol. 20, no. 3, pp. 273–298, 1996.

[8] M. Cotronei, L. B. Montefusco, and L. Puccio, "Multiwavelet analysis and signal processing," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 8, pp. 970–987, 1998.

[9] V. Strela, P. N. Heller, G. Strang, P. Topiwala, and C. Heil, "The application of multiwavelet filterbanks to image processing," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 548–563, 1999.

[10] J. Lebrun and M. Vetterli, "Balanced multiwavelets theory and design," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1119–1125, 1998.

[11] K. Attakitmongcol, D. P. Hardin, and D. Mitchell Wilkes, "Multiwavelet prefilters—part II: optimal orthogonal prefilters," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1476–1487, 2001.