Science in China Ser. F Information Sciences 2004 Vol.47 No.1 1—16

Block TERM factorization of block matrices

SHE Yiyuan & HAO Pengwei

Center for Information Science, Peking University, Beijing 100871, China (email: {sheyy, phao}@cis.pku.edu.cn)

Received July 2, 2003

Abstract Reversible integer mapping (or integer transform) is a useful way to realize lossless coding, and this technique has been used for multi-component image compression in the new international image compression standard JPEG 2000. For any nonsingular linear transform of finite dimension, its integer transform can be implemented by factorizing the transform matrix into 3 triangular elementary reversible matrices (TERMs) or a series of single-row elementary reversible matrices (SERMs). To speed up and parallelize integer transforms, we study block TERM and SERM factorizations in this paper. First, to guarantee flexible scaling manners, the classical determinant (*det*) is generalized to a matrix function, **DET**, which is shown to have many important properties analogous to those of *det*. Then based on **DET**, a generic block TERM factorization, **BLUS**, is presented for any nonsingular block matrix. Our conclusions can cover the early optimal point factorizations and provide an efficient way to implement integer transforms for large matrices.

Keywords: integer mapping, lossless coding, parallel computing, determinant, block matrix factorization.

DOI: 10.1360/ 03yf0499

Due to the limitation of computational precision and storage capacity, transforms used in lossless data compression should be equivalently integer-reversible. Reversible integer transform (or integer mapping) is such a type of transform that maps integers to integers and realizes perfect reconstruction (PR). People started to work in this area long ago, and their early work, such as S transform[1], TS transform[2], S+P transform[3], and color space transforms[4], suggested a promising future of reversible integer mapping in image compression, region-of-interest (ROI) coding, progressive transmission, and unified lossy/lossless compression systems. However, to construct such integer transforms, they used to resort to some special skills.

Bruekers and van den Enden's work on perfect inversion (PI) and perfect reconstruction (PR) lends great insight into the problem of reversible transformation: (i) the ladder network is an elementary unit of PI, and the forward and the backward transforms based on it have 'symmetry', whereas the choices of specific linear and nonlinear operators are by no means essential; (ii) from the viewpoint of matrix structure, PR depends on the possibility of decomposing a transform matrix into a few vector matrices, where a row vector matrix is actually a unit SERM as defined in [10]; (iii) a unit triangular matrix is the product of a series of vector matrices, and furthermore, besides a permutation, there always exists a vector matrix factorization for any matrix with determinant 1.

Yet at that time their work did not arouse much attention from the researchers in this area. As a matter of fact, not until lifting scheme (LS)[6] was proposed for constructing the second generation wavelets did people try to break away from various specific transforms and roundings to build generic integer wavelet transforms. By integrating the nonlinear quantization into the LS, Dewitte and Cornelis[7] first constructed a few simple forward/backward integer transforms. Then Calderbank et al.[8] systematically presented the concept of integer wavelet transforms (IWT) based on the simplified ladder structure of LS. Afterwards, research in this area is booming and the techniques are widely adopted in applications.

For finite dimensional signals, the transform matrix can be simplified from a polyphase matrix consisting of Laurent polynomials[8] to a constant matrix of finite dimension. By matrix factorization, Hao and Shi first considered reversible integer implementations for any invertible linear transforms in a finite dimensional space[9], and recently obtained an optimal factorization of minimum number of matrices[10]. This technique[11] has been included in the new international image compression standard JPEG 2000.

The rest of the paper is organized as follows. In Section 1, we summarize Hao and Shi's matrix factorization theory for reversible integer transform, and address its efficiency problem caused by recursiveness. Section 2 introduces some notations and symbols. In Section 3, by using an auxiliary function $\boldsymbol{W}^{(n)}$, we define a new determinant function, **DET**, as a block generalization of the classical determinant (*det*). Then, from **DET** Section 4 obtains a general **BLUS** block TERM factorization which has a very flexible scaling manner. To better apply the **BLUS** factorization in practice, permutation preprocessings are discussed in Section 5 to guarantee the factorizability for any nonsingular transform matrix. Section 6 demonstrates the factorization with some examples.

1 Hao and Shi's matrix factorization theory

The basic factors are called elementary reversible matrices (ERMs), including triangular ERMs (TERMs) and single-row ERMs (SERMs). A TERM is a special triangular matrix of which the diagonal elements belong to the unit group of an integral domain. For instance, they are ± 1 and $\pm i$ on the set $\{a+bi|a,b\in Z\}$, the so-called integer factors in [10]. Obviously, a unit TERM is a unit triangular matrix. Given A = $(a_{i,j})$, an upper TERM of size N with its diagonal entries $j_1, \cdots, j_N$, the forward integer transform for $\boldsymbol{y} = \boldsymbol{Ax}$ is computed as follows:

$$\begin{cases} y_m = j_m x_m + \left\lfloor \sum_{n=m+1}^{N} a_{m,n} x_n \right\rfloor, & 1 \leqslant m \leqslant N-1, \\ y_N = j_N x_N, & \end{cases}$$

while its inverse should be executed in a recursive way like backward substitution:

$$x_N = y_N / j_N\,; \quad x_m = \left(y_m - \left\lfloor \sum_{n=m+1}^{N} a_{m,n} x_n \right\rfloor \right) \Big/ j_m\,, \quad m = N-1, \cdots, 1,$$

where '$\lfloor\ \rfloor$' can be rounding-up, rounding-down, chopping or any other rounding arithmetic. The computation is analogous for a lower TERM, except that the computational ordering of the inverse should be downward. We easily see the following characteristics in the above transform computations: (i) mapping integers to integers; (ii) perfect reconstruction; (iii) in-place computation. All these are attractive for lossless data compression.

Another type of ERM is single-row ERM, or SERM. A SERM is a matrix with integer factors on the diagonal and only one row possibly nonzero. It can be converted into a simple TERM by row and column permutations. Note that all components can be independently reconstructed in the inverse integer transform of a unit SERM.

Given an $N \times N$ nonsingular matrix $\boldsymbol{A}$, the main conclusion of ref. [10] is $\boldsymbol{A} = \boldsymbol{PLDUS}_0$, where $\boldsymbol{L}$, $\boldsymbol{U}$ are respectively unit lower/upper triangular matrices, $\boldsymbol{D} = diag(1, \cdots 1, \det(\boldsymbol{P}^T \boldsymbol{A}))$, and $\boldsymbol{S}_0$ is a unit SERM associated with the last row. Particularly, if $\det(\boldsymbol{A})$ is an integer factor, we obtain a 3-TERM factorization $\boldsymbol{A} = \boldsymbol{PLUS}_0$, and the corresponding SERM factorization $\boldsymbol{A} = \boldsymbol{PS}_N \cdots \boldsymbol{S}_1 \boldsymbol{S}_0$. To implement reversible integer transform for any nonsingular transform matrix, the scaling modification is usually necessary: we are rather free to choose any row(s) or column(s) to perform the scaling, as long as the final determinant is an integer factor.

As we have pointed out, the algorithm of the inverse TERM integer transform is recursive, though it evades the workload of evaluating an inverse matrix. For instance, an inverse upper triangular transform should be executed stepwise from bottom to top, which makes the computing rather painful for large matrices. As for the SERM integer transform, actually, only one component can be changed through a unit SERM as in a single lifting step; thus it is still sequential in practice. Motivated by the observation that the potential parallelism of SERM can be trivially generalized to unit block SERMs $\boldsymbol{S}_i = \boldsymbol{I} + \boldsymbol{e}_i \boldsymbol{s}_i^T$, where $\boldsymbol{e}_i$ is an elementary block matrix with the i-th block $\boldsymbol{I}$ and others $\boldsymbol{0}$, and $\boldsymbol{s}_i$ is a block matrix with the i-th block $\boldsymbol{0}$, we study block TERM and SERM factorizations in this paper. In contrast to point SERM factorizations, block SERM factorizations boost the degree of parallelism and make it possible that the factorization and transforms are carried out at the block level. Such block approaches are appropriate for efficient integer implementation of large matrices, let alone those with natural block structures originated from underlying physical backgrounds.

Generalizing the point factorizations to block factorizations is not so straightforward due to the difficulty of the scaling modification and the possibility that some crucial blocks may not have full rank in factorization. In the following discussions, we shall use a matrix function, **DET**, as a generalization of the classical *det* to study block triangular matrix factorizations.

2 Symbols and notations

Throughout the rest of the paper, we use the following notations and symbols:

- To distinguish from a block matrix, an un-partitioned matrix is also called an element matrix.
- Unless otherwise specified, we suppose all blocks in a block matrix are of the same size, and use $\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n1} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}$ to denote a general block matrix with n rows and n columns of blocks, where $\boldsymbol{A}_{i,j}$ is the 'block' at the i-th row and j-th column, a square element matrix in this paper. n is also called the order of this block matrix. Denote by $\boldsymbol{A}_n(i_1, i_2, \cdots, i_m; j_1, j_2, \cdots, j_m)$ the block submatrix $\begin{bmatrix} \boldsymbol{A}_{i_1 j_1} & \cdots & \boldsymbol{A}_{i_1 j_m} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{i_m j_1} & \cdots & \boldsymbol{A}_{i_m j_m} \end{bmatrix}$, where $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$ and $1 \leqslant j_1 < j_2 < \cdots < j_m \leqslant n$.
- Denote by $(i) \cdot \boldsymbol{X} + (j) \rightarrow (j)'$ the block elementary column operation of adding the i-th column of blocks right-multiplied by $\boldsymbol{X}$ to the j-th column of blocks; and $\boldsymbol{Y} \cdot (i) + (j) \rightarrow (j)'$ the block elementary row operation of adding the i-th row of blocks left-multiplied by $\boldsymbol{Y}$ to the j-th row of blocks, where $i \neq j$.

3 DET matrix function

In this section, we shall extend the classical scalar determinant function to block matrices, such that we get a mapping from block matrices to matrices.

Definition 1. Given a block matrix $\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n1} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}$, we refer to $\boldsymbol{B}_k = \boldsymbol{A}(1,\cdots,k;\ 1,\cdots k-1,n)$ as its k-th quasi leading principal block submatrix for $k = 1,2,\cdots,n$.

3.1 Auxiliary function $\boldsymbol{W}^{(n)}$

Definition 2. Given a block matrix $\boldsymbol{A}_n$, $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is recursively defined as follows:

(i) For $n=1$, $\boldsymbol{W}^{(1)}(\boldsymbol{A}_1) := \boldsymbol{A}_{11}$;

(ii) For $n=2$, if $\boldsymbol{A}_{12}$ is invertible, then $\boldsymbol{W}^{(2)}(\boldsymbol{A}_2):=\boldsymbol{A}_{22}\boldsymbol{A}_{12}^{-1}\boldsymbol{A}_{11}-\boldsymbol{A}_{21}$;

(iii) For $n\geqslant 3$, if $\boldsymbol{W}^{(n-1)}(\boldsymbol{A}_n(1,\cdots,n-1;1\cdots,n-2,n))$ exists and is invertible, then $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is defined by

$$\boldsymbol{W}^{(n)}(A_n):=\boldsymbol{D}_{n,n-1}^{(n)}-\boldsymbol{D}_{n,n-2}^{(n)}(\boldsymbol{D}_{n-1,n-2}^{(n)})^{-1}\boldsymbol{D}_{n-1,n-1}^{(n)},$$

where $\boldsymbol{D}_{i,j}^{(n)}$ is a notation for $\boldsymbol{W}^{(n-1)}(\boldsymbol{A}_n(1,\cdots n-2,i;\ 1,\cdots n-3,j,n))$ with $n-1\leqslant i\leqslant n$ and $n-2\leqslant j\leqslant n-1$.

Suppose $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is defined. Then

$$\boldsymbol{W}^{(1)}(\boldsymbol{A}_1)=\boldsymbol{W}^{(1)}(\boldsymbol{A}_{11})=\boldsymbol{A}_{11},$$

$$\boldsymbol{W}^{(2)}(\boldsymbol{A}_2)=\boldsymbol{W}^{(2)}\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix}=\boldsymbol{A}_{22}\boldsymbol{A}_{12}^{-1}\boldsymbol{A}_{11}-\boldsymbol{A}_{21},$$

$$\begin{aligned}\boldsymbol{W}^{(3)}(\boldsymbol{A}_3)=\boldsymbol{W}^{(3)}&\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{A}_{13}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23}\\ \boldsymbol{A}_{31} & \boldsymbol{A}_{32} & \boldsymbol{A}_{33}\end{bmatrix}\\ &=\boldsymbol{D}_{3,2}^{(3)}-\boldsymbol{D}_{3,1}^{(3)}(\boldsymbol{D}_{2,1}^{(3)})^{-1}\boldsymbol{D}_{2,2}^{(3)}\\ &=(\boldsymbol{A}_{33}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{12}-\boldsymbol{A}_{32})\\ &\quad-(\boldsymbol{A}_{33}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11}-\boldsymbol{A}_{31})(\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11}-\boldsymbol{A}_{21})^{-1}(\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{12}-\boldsymbol{A}_{22}),\end{aligned}$$

3.2 Properties of $\boldsymbol{W}^{(n)}$

Property 1. Suppose $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is defined. If $\boldsymbol{A}_{11}$ is invertible and $\boldsymbol{A}_{21}=\cdots=\boldsymbol{A}_{n,1}$ $=\boldsymbol{0}$, then $\boldsymbol{W}^{(n-1)}(\boldsymbol{A}_n(2,\cdots,n;2,\cdots n))$ exists and $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)=\boldsymbol{W}^{(n-1)}(\boldsymbol{A}_n(2,\cdots,n;2,\cdots n))$, $\forall n\geqslant 3$.

Proof. First prove the case n = 3. It is easy to know that $\boldsymbol{D}_{2,1}^{(3)}=\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11}$, $\boldsymbol{D}_{3,1}^{(3)}=\boldsymbol{A}_{33}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11}$, from which it follows that $\boldsymbol{A}_{23}$ is invertible and $\boldsymbol{W}^{(2)}(\boldsymbol{A}_3(2,3;2,3))$ is defined. Hence

$$\begin{aligned}\boldsymbol{W}^{(3)}(\boldsymbol{A}_3)=\boldsymbol{D}_{3,2}^{(3)}-\boldsymbol{D}_{3,1}^{(3)}(\boldsymbol{D}_{2,1}^{(3)})^{-1}\boldsymbol{D}_{2,2}^{(3)}\\ =\boldsymbol{D}_{3,2}^{(3)}-\boldsymbol{A}_{3,3}\boldsymbol{A}_{1,3}^{-1}\boldsymbol{A}_{1,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,3}\boldsymbol{A}_{2,3}^{-1}\boldsymbol{D}_{2,2}^{(3)}\\ =\boldsymbol{A}_{3,3}\boldsymbol{A}_{2,3}^{-1}\boldsymbol{A}_{2,2}-\boldsymbol{A}_{3,2}.\end{aligned}$$

Then by induction it is easy to show that for any $n>3$,

$$W^{(n-1)}(A_n(1,\cdots,n-2,i;1,\cdots,n-3,j,n)) = W^{(n-2)}(A_n(2,\cdots,n-2;2,\cdots,n-3,j,n)),$$

where $n-1 \leqslant i \leqslant n$, $n-2 \leqslant j \leqslant n-1$.

Property 2. $W^{(n)}$ has left linearity in the last row, and right linearity in the (n–1)-th column of blocks, that is,

$$W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \cdots & \cdots & \cdots \\ A_{n-1,1} & \cdots & A_{n-1,n} \\ XA_{n,1}+A'_{n,1} & \cdots & XA_{n,n}+A'_{n,n} \end{bmatrix} = X\cdot W^{(n)}(A_n) + W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \cdots & \cdots & \cdots \\ A_{n-1,1} & \cdots & A_{n-1,n} \\ A'_{n,1} & \cdots & A'_{n,n} \end{bmatrix},$$

$$W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A_{1,n-1}Y+A'_{1,n-1} & A_{1,n} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n,1} & \cdots & A_{n,n-1}Y+A'_{n,n-1} & A_{n,n} \end{bmatrix} = W^{(n)}(A_n)\cdot Y + W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A'_{1,n-1} & A_{1,n} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n,1} & \cdots & A'_{n,1} & A_{n,n} \end{bmatrix},$$

where $n>1$.

It follows that

$$W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \cdots & \cdots & \cdots \\ XA_{i,1} & \cdots & XA_{i,n} \\ \cdots & \cdots & \cdots \\ A_{n,1} & \cdots & A_{n,n} \end{bmatrix} = W^{(n)}(A_n),\ \ 1 \leqslant i \leqslant n-1,$$

$$W^{(n)}\begin{bmatrix} A_{1,1} & \cdots & A_{1,j}Y & \cdots & A_{1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{n,1} & \cdots & A_{n,j}Y & \cdots & A_{n,n} \end{bmatrix} = W^{(n)}(A_n),\ \ j \neq n-1,$$

as long as X and Y are invertible. The above properties lead to the following propositions.

Proposition 1. If $W^{(n)}(A_n)$ is defined and $A_{n,i} = 0$ (or $A_{i,n-1} = 0$) for $i = 1,\cdots,n$, then $W^{(n)}(A_n) = 0$.

Proposition 2. If $W^{(n)}(A_n)$ is defined and there exists a $k(\neq n)$ such that $A_{n,i} = A_{k,i}$ hold for all $i = 1,2,\cdots,n$, then $W^{(n)}(A_n) = 0$. Similar property also holds for the (n–1)-th column of blocks.

Property 3. Suppose that $W^{(n)}(A_n)$ is defined. Then $W^{(n)}$ remains unchanged under the block elementary row operation of $X\cdot(i)+(j)\to(j)'$, where $i<j$.

Proof. First prove the case $j = n$ by induction. Then use it to prove $j = n-1$. Finally prove the case $j \leqslant n-2$ by induction.

Property 4. Suppose that $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is defined. Then $\boldsymbol{W}^{(n)}$ remains unchanged under the block elementary column operation of $(i)\cdot \boldsymbol{Y} + (j) \to (j)'$, where $i = n$ and $j < n$.

3.3 Definition of **DET**

Now we give the definition of **DET** based on $\boldsymbol{W}^{(n)}$.

Definition 3. Given a block matrix $\boldsymbol{A}_{\mathrm{n}}$, $\mathbf{DET}(\boldsymbol{A}_n)$ is recursively defined as follows:

(i) For $n = 1$, $\mathbf{DET}(\boldsymbol{A}_1) := \boldsymbol{A}_{11}$;

(ii) For $n \geqslant 2$, if $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is defined, then $\mathbf{DET}(\boldsymbol{A}_n)$ exists and is defined by

$$\mathbf{DET}(\boldsymbol{A}_n) := \boldsymbol{W}^{(n)}(\boldsymbol{A}_n)\cdot \mathbf{DET}(\boldsymbol{B}_{n-1}) = \boldsymbol{W}^{(n)}(\boldsymbol{A}_n)\cdot \mathbf{DET}(\boldsymbol{A}_n(1,\cdots,n-1;1,\cdots,n-2,n)).$$

From the definition, if $\mathbf{DET}(\boldsymbol{A}_n)$ exists, we immediately know that **DET** also exists for all its quasi leading principal block submatrix $\boldsymbol{B}_k$, where $k = 1,\cdots,n$; and from the computation of $\mathbf{DET}(\boldsymbol{A}_n)$ we get all $\mathbf{DET}(\boldsymbol{B}_k)$, too. Suppose that $\mathbf{DET}(\boldsymbol{A}_n)$ is defined, then we have

$$\mathbf{DET}(\boldsymbol{A}_1) = \mathbf{DET}(\boldsymbol{A}_{11}) = \boldsymbol{A}_{11},$$

$$\mathbf{DET}(\boldsymbol{A}_2) = \mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} = (\boldsymbol{A}_{22}\boldsymbol{A}_{12}^{-1}\boldsymbol{A}_{11} - \boldsymbol{A}_{21})\cdot \boldsymbol{A}_{12},$$

$$\begin{aligned}\mathbf{DET}(\boldsymbol{A}_3) = \mathbf{DET}&\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{A}_{13} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23} \\ \boldsymbol{A}_{31} & \boldsymbol{A}_{32} & \boldsymbol{A}_{33} \end{bmatrix} \\ &= (\boldsymbol{D}_{3,2}^{(3)} - \boldsymbol{D}_{3,1}^{(3)}(\boldsymbol{D}_{2,1}^{(3)})^{-1}\boldsymbol{D}_{2,2}^{(3)})\cdot \boldsymbol{D}_{2,1}^{(3)}\cdot \boldsymbol{A}_{13} \\ &= [(\boldsymbol{A}_{33}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{12} - \boldsymbol{A}_{32}) \\ &\quad - (\boldsymbol{A}_{33}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11} - \boldsymbol{A}_{31})(\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11} - \boldsymbol{A}_{21})^{-1}(\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{12} - \boldsymbol{A}_{22})]\cdot \\ &\quad (\boldsymbol{A}_{23}\boldsymbol{A}_{13}^{-1}\boldsymbol{A}_{11} - \boldsymbol{A}_{21})\boldsymbol{A}_{13},\end{aligned}$$

…

3.4 Properties of **DET**

In this subsection, we shall show that **DET** has some important properties analogous to those of *det*. All proofs are simple and thus omitted.

Property 1. Suppose $\mathbf{DET}(\boldsymbol{A}_n)$ is defined. If $\boldsymbol{A}_{11} = \boldsymbol{I}$ and $\boldsymbol{A}_{21} = \cdots = \boldsymbol{A}_{n,1} = \boldsymbol{0}$, then $\mathbf{DET}(\boldsymbol{A}_n) = \mathbf{DET}(\boldsymbol{A}_n(2,\cdots,n;2,\cdots n))$, $\forall n \geqslant 2$.

Property 2. **DET** has left linearity in the last row of blocks, that is, for $n > 1$,

$$\mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n-1,1} & \cdots & \boldsymbol{A}_{n-1,n} \\ \boldsymbol{X}\boldsymbol{A}_{n,1} + \boldsymbol{A}'_{n,1} & \cdots & \boldsymbol{X}\boldsymbol{A}_{n,n} + \boldsymbol{A}'_{n,n} \end{bmatrix} = \boldsymbol{X} \cdot \mathbf{DET}(\boldsymbol{A}_n) + \mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n-1,1} & \cdots & \boldsymbol{A}_{n-1,n} \\ \boldsymbol{A}'_{n,1} & \cdots & \boldsymbol{A}'_{n,n} \end{bmatrix}.$$

As for the last column of blocks, if $\boldsymbol{X}$ is invertible, then

$$\mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,n-1} & \boldsymbol{A}_{1,n}\boldsymbol{X} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n,1} & \cdots & \boldsymbol{A}_{n,n-1} & \boldsymbol{A}_{n,n}\boldsymbol{X} \end{bmatrix} = \mathbf{DET}(\boldsymbol{A}_n) \cdot \boldsymbol{X}.$$

And for other rows or columns of blocks, if $\boldsymbol{X}$ is invertible, then

$$\mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{X}\boldsymbol{A}_{i,1} & \cdots & \boldsymbol{X}\boldsymbol{A}_{i,n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n,1} & \cdots & \boldsymbol{A}_{n,n} \end{bmatrix} = \mathbf{DET}\begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,i}\boldsymbol{X} & \cdots & \boldsymbol{A}_{1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n,1} & \cdots & \boldsymbol{A}_{n,i}\boldsymbol{X} & \cdots & \boldsymbol{A}_{n,n} \end{bmatrix}$$

$$= \boldsymbol{W}^{(n)}(\boldsymbol{B}_n)\cdots\boldsymbol{W}^{(i+1)}(\boldsymbol{B}_{i+1}) \cdot \boldsymbol{X} \cdot \mathbf{DET}(\boldsymbol{B}_i) = \mathbf{DET}(\boldsymbol{A}_n) \cdot \boldsymbol{Y},$$

where $\boldsymbol{Y} = \mathbf{DET}(\boldsymbol{B}_i)^{-1} \cdot \boldsymbol{X} \cdot \mathbf{DET}(\boldsymbol{B}_i)$, $1 \leqslant i \leqslant n-1$. (Actually, $\boldsymbol{X}$ is not necessarily invertible for the (n–1)-th column of blocks.)

In summary, we get

$$\mathbf{DET}\left(\begin{bmatrix} \boldsymbol{X}_1 & & \\ & \ddots & \\ & & \boldsymbol{X}_n \end{bmatrix}\begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n1} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}\right) = \boldsymbol{X}_n \cdot \boldsymbol{W}^{(n)}(\boldsymbol{A}_n)\cdots\boldsymbol{X}_1 \cdot \boldsymbol{W}^{(1)}(\boldsymbol{A}_1),$$

$$\mathbf{DET}\left(\begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1n} \\ \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n1} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}\begin{bmatrix} \boldsymbol{Y}_1 & & \\ & \ddots & \\ & & \boldsymbol{Y}_n \end{bmatrix}\right) = \boldsymbol{W}^{(n)}(\boldsymbol{A}_n) \cdot \boldsymbol{Y}_{n-1}\cdots\boldsymbol{W}^{(2)}(\boldsymbol{A}_2) \cdot \boldsymbol{Y}_1 \cdot \boldsymbol{W}^{(1)}(\boldsymbol{A}_1) \cdot \boldsymbol{Y}_n,$$

where $\boldsymbol{X}_i$ $(1 \leqslant i \leqslant n-1)$ and $\boldsymbol{Y}_j$ $(1 \leqslant j \leqslant n,\ j \neq n-1)$ should be invertible. These two nice scaling properties of **DET** are an important guarantee of the flexibility and practicability of scaling modification.

Property 3. Suppose that $\mathbf{DET}(\boldsymbol{A}_n)$ is defined. Then for $i<j$, **DET** remains unchanged under the block elementary row operation of $\boldsymbol{X}\cdot(i)+(j)\rightarrow(j)'$.

Property 4. Suppose that $\mathbf{DET}(\boldsymbol{A}_n)$ is defined. Then for $i=n$ and $j<n$, **DET** remains unchanged under the block elementary column operation of $(i)\cdot\boldsymbol{Y}+(j)\rightarrow(j)'$.

4 BLUS TERM factorization

In this section, we discuss how to factorize a block matrix into a product of the least number of block TERMs. First, we give a block factorization algorithm of $\boldsymbol{A}_n$ as follows:

Algorithm BLUS

Let $\boldsymbol{A}_n^{(i)}$ (or simply $\boldsymbol{A}^{(i)}$) denote the block matrix obtained in the i-th iteration, and $\boldsymbol{A}_n^{(n)}$ the final one. $\boldsymbol{A}_n^{(1)}=\boldsymbol{A}_n$.

For i = 1 to n–1

If $\boldsymbol{A}_{i,n}^{(i)}$ is invertible, perform the following operations; else quit.

(i.1) Apply block row operation $(n)\cdot\boldsymbol{X}_i+(i)\rightarrow(i)$, such that $\boldsymbol{A}_{i,i}^{(i)}$ becomes $\boldsymbol{I}$;

(i.2) Carry out the block forward elimination of the i-th column of blocks.

End

Theorem 1. If $\mathbf{DET}(\boldsymbol{A}_n)$ is defined, then Algorithm **BLUS** can be executed step by step until finally it yields a factorization: $\boldsymbol{A}_n=\boldsymbol{LDUS}$, where $\boldsymbol{D}=diag(\boldsymbol{I},\cdots,\boldsymbol{I},\mathbf{DET}(\boldsymbol{A}))$, $\boldsymbol{L}$ and $\boldsymbol{U}$ are block unit lower triangular matrix and upper triangular matrix respectively, and $\boldsymbol{S}$ is a block unit SERM associated with the last row of blocks. Clearly, the factorization can also be formulated as $\boldsymbol{A}_n=\boldsymbol{D}\cdot\boldsymbol{LUS}$ or $\boldsymbol{A}_n=\boldsymbol{LUS}\cdot\boldsymbol{D}$.

Proof. Once $\boldsymbol{A}_{i,n}^{(i)}$ is guaranteed to be invertible in each step (i.1), the algorithm runs without premature termination. By **DET**'s Property 3 and Property 4, we know that **DET** is unchanged after operations (i.1) through (i.2), that is, $\mathbf{DET}(\boldsymbol{A}_n^{(i)})=\mathbf{DET}(\boldsymbol{A}_n)$

for $i=1,2,\cdots,n$. From **DET**'s Property 1, $\mathbf{DET}(\boldsymbol{A}_n^{(i)}(i,\cdots,n;i,\cdots,n))$ exists and $\mathbf{DET}(\boldsymbol{A}_n^{(i)})=\mathbf{DET}(\boldsymbol{A}_n^{(i)}(i,\cdots,n;i,\cdots,n))$. Hence Algorithm **BLUS** can be executed till it stops normally, and $\boldsymbol{A}_{n,n}^{(n)}=\mathbf{DET}(\boldsymbol{A}_n)$. Moreover, from the whole process $\boldsymbol{L}\boldsymbol{A}_n\boldsymbol{S}=\boldsymbol{D}\boldsymbol{U}$, it is easy to obtain the final factorization formula. The proof is now complete.

Corollary 1. If $\mathbf{DET}(\boldsymbol{A}_n)$ is a diagonal matrix with its diagonal consisting of integer factors only, then $\boldsymbol{A}_n$ has a block TERM factorization $\boldsymbol{A}_n=\boldsymbol{LUS}$.

Similar to the element case (see Theorem 4 of ref. [10]), $\boldsymbol{LU}$ can be further factorized into n block unit SERMs $\boldsymbol{S}_n\boldsymbol{S}_{n-1}\cdots\boldsymbol{S}_1$. Thus we have obtained a block SERM factorization for $\boldsymbol{A}_n$ as desired for the parallel computation of its integer transform.

Corollary 2 (Compatibility). If $\mathbf{DET}(\boldsymbol{A}_n)$ exists, then $\det(\mathbf{DET}(\boldsymbol{A}_n))=\det(\boldsymbol{A}_n)$.

This corollary can also be used conversely to calculate the determinant of a large matrix.

Corollary 3 (Degeneration). Let $\boldsymbol{A}_n$ denote the same matrix of $\boldsymbol{A}=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$, taken as a block matrix with 1-by-1 blocks. If $\mathbf{DET}(\boldsymbol{A}_n)$ exists, then $\mathbf{DET}(\boldsymbol{A}_n)=\det(\boldsymbol{A})$.

From the above two corollaries, we see the close relationship between **DET** and *det*: **DET** is a generalization of *det*, and *det* can be regarded as a special case of **DET**. Furthermore, using the permutation techniques introduced in the next section, we shall show that our block factorization is compatible with the point TERM factorizations of ref. [10].

5 Block TERM factorization for any nonsingular matrix

The existence of **DET** is one of the most important problems of applying the block TERM and SERM factorizations in practice.

Lemma 1. $\boldsymbol{W}^{(n)}(\boldsymbol{A}_n)$ is invertible if and only if $\mathbf{DET}(\boldsymbol{A}_n)$ is invertible.

Proof. First prove its necessity by induction. Then use it to prove the sufficiency.

Theorem 2. For block matrix $\boldsymbol{A}_n$ ($n\geqslant 2$), the necessary and sufficient condition for the existence of $\mathbf{DET}(\boldsymbol{A}_n)$ is that its 1st through (n–1)-th quasi leading principal block submatrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_{n-1}$ are all invertible.

Proof. From Lemma 1 and Corollary 2, its necessity can be easily proved by induction. Now we prove its sufficiency by induction on n. The case $n = 2$ is trivial. Suppose that the theorem holds for block matrices of order $n-1$ or less. It is easy to know that $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots \boldsymbol{B}_{n-2}$ are just quasi leading principal block submatrices of $\boldsymbol{B}_{n-1}$. By Corollary 2, we have $\det(\mathbf{DET}(\boldsymbol{B}_{n-1})) = \det(\boldsymbol{B}_{n-1})$, from which it follows that $\mathbf{DET}(\boldsymbol{B}_{n-1})$ is invertible, and thus $\boldsymbol{W}^{(n-1)}(\boldsymbol{B}_{n-1})$ is invertible. Then by Definition 3 we easily know that $\mathbf{DET}(\boldsymbol{A}_n)$ is defined.

Denote by $(\boldsymbol{A}_n)_e$ the original element matrix corresponding to the block matrix $\boldsymbol{A}_n$, and denote by $(\boldsymbol{A})_{p(\boldsymbol{A}_n)}$ the block matrix obtained by partitioning the element matrix $\boldsymbol{A}$ conformably with $\boldsymbol{A}_n$.

Proposition 3. Given nonsingular $\boldsymbol{A}$ and $\boldsymbol{A}_n$ satisfying $\boldsymbol{A} = (\boldsymbol{A}_n)_e$, there exists an element row permutation matrix $\boldsymbol{P}$, such that for the new block matrix $\boldsymbol{A}'_n = (\boldsymbol{PA})_{p(\boldsymbol{A}_n)}$, $\mathbf{DET}(\boldsymbol{A}'_n)$ is defined and invertible. An analogous conclusion holds for column permutations.

Proof. Set $\boldsymbol{E}_1 = \boldsymbol{A}_n$, $\boldsymbol{H}_1 = \begin{bmatrix} \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,n-2} & \boldsymbol{A}_{1,n} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A}_{n,1} & \cdots & \boldsymbol{A}_{n,n-2} & \boldsymbol{A}_{n,n} \end{bmatrix}$ and $\boldsymbol{G}_1 = (\boldsymbol{H}_1)_e$. Since $\boldsymbol{A}_n$ is invertible, $\boldsymbol{G}_1$ has full column rank. Let c_1 be the number of its columns. Then there must exist a permutation $\boldsymbol{P}_1$ such that the submatrix corresponding to the first c_1 rows of $\boldsymbol{G}'_1 = \boldsymbol{P}_1\boldsymbol{G}_1$ is invertible, which means the $(n-1)$-th quasi leading principal block submatrix of $(\boldsymbol{P}_1\boldsymbol{A})_{p(\boldsymbol{A}_n)}$, denoted by $\boldsymbol{E}_2$, is invertible. Similarly, $\boldsymbol{P}_2$ can be obtained from analogous processing of $\boldsymbol{E}_2$, such that the $(n-2)$-th leading principal block submatrix of $(\boldsymbol{P}_2(\boldsymbol{P}_1\boldsymbol{A}))_{p(\boldsymbol{A}_n)}$ is invertible. Moreover, it is easy to see that its $(n-1)$-th leading principal block submatrix still remains invertible. (Strictly speaking, to act on $\boldsymbol{A}$, $\boldsymbol{P}_2$ should be expanded as $\begin{bmatrix} \boldsymbol{P}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$.) Proceeding in this way, we finally get $\boldsymbol{P} = \boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_1$ such that all the quasi leading principal block submatrices of $\boldsymbol{A}'_n = (\boldsymbol{PA})_{p(\boldsymbol{A}_n)}$ are invertible. Hence $\mathbf{DET}(\boldsymbol{A}'_n)$ is defined by Theorem 2 and is invertible by Corollary 2.

From the proposition, we can obtain the factorizations $\boldsymbol{A}_n = \boldsymbol{PLDUS}$ and $\boldsymbol{A}_n = \boldsymbol{LDUSP}$ as long as $\boldsymbol{A}_n$ is invertible, where $\boldsymbol{P}$ is permutation matrix at the element level, and the last term of $\boldsymbol{D}$ is $\mathbf{DET}(\boldsymbol{P}^T \boldsymbol{A}_n)$ or $\mathbf{DET}(\boldsymbol{A}_n \boldsymbol{P}^T)$. Thus when all blocks are of size one-by-one, our factorization is just the element TERM factorization given by ref [10].

To deal with the term $\boldsymbol{D}$, on the one hand, we may scale $\boldsymbol{A}_n$ by Property 2 to normalize **DET**, and thus obtain a block unit triangular factorization afterwards; on the other hand, the last block can be repeatedly partitioned and factorized till the block size is reduced to a moderate level to apply the element matrix TERM factorization at last. It is worth mentioning that here, we are rather free to choose any row(s) or column(s) of blocks (as in the element case) to make the scaling, thanks to **DET**'s convenient scaling property. To satisfy the requirement of equally-sized blocks, it is sometimes necessary to expand the original transform matrix with $\boldsymbol{I}$ along the diagonal. Finally, although we can directly prove the factorization ***PLDUS***, **DET** provides us the possibility to consider the factorization residue before the whole process, and a very flexible scaling manner analogous to ref. [10], which is useful in adjusting the dynamic ranges of the transform coefficients.

6 Examples

We demonstrate our factorizations by a real transform – 9×9 DCT.

Example. 9-by-9 DCT.

$$\boldsymbol{A}=\begin{bmatrix} 0.3333 & 0.3333 & 0.3333 & 0.3333 & 0.3333 & 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.4642 & 0.4082 & 0.3030 & 0.1612 & 0.0000 & -0.1612 & -0.3030 & -0.4082 & -0.4642 \\ 0.4430 & 0.2357 & -0.0819 & -0.3611 & -0.4714 & -0.3611 & -0.0819 & 0.2357 & 0.4430 \\ 0.4082 & 0.0000 & -0.4082 & -0.4082 & -0.0000 & 0.4082 & 0.4082 & 0.0000 & -0.4082 \\ 0.3611 & -0.2357 & -0.4430 & 0.0819 & 0.4714 & 0.0819 & -0.4430 & -0.2357 & 0.3611 \\ 0.3030 & -0.4082 & -0.1612 & 0.4642 & 0.0000 & -0.4642 & 0.1612 & 0.4082 & -0.3030 \\ 0.2357 & -0.4714 & 0.2357 & 0.2357 & -0.4714 & 0.2357 & 0.2357 & -0.4714 & 0.2357 \\ 0.1612 & -0.4082 & 0.4642 & -0.3030 & -0.0000 & 0.3030 & -0.4642 & 0.4082 & -0.1612 \\ 0.0819 & -0.2357 & 0.3611 & -0.4430 & 0.4714 & -0.4430 & 0.3611 & -0.2357 & 0.0819 \end{bmatrix}.$$

Without loss of generality, partition the transform matrix into three blocks of size 3-by-3. We can choose one of the following ways to perform the factorization.

(i) Scale before the factorization.

From Proposition 3 and **DET**'s definition, we get $\boldsymbol{A}'_n = \boldsymbol{P}^T\boldsymbol{A}_n$, and

$$\mathbf{DET}(\boldsymbol{A}'_n)=\begin{bmatrix} 1.0279 & 0.7223 & 0.0170 \\ 0.6231 & -0.6022 & -0.4636 \\ 0.1745 & -0.1260 & 0.8250 \end{bmatrix},$$

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

At the same time we get all **DET**s for $\boldsymbol{B}'_k$, the quasi leading principal block sub-matrices of $\boldsymbol{A}'_n$:

$$\mathbf{DET}(\boldsymbol{B}'_1) = \begin{bmatrix} -0.4642 & 0.4082 & -0.1612 \\ -0.3030 & -0.4082 & -0.4642 \\ -0.4430 & -0.2357 & 0.3611 \end{bmatrix},$$

$$\mathbf{DET}(\boldsymbol{B}'_2) = \begin{bmatrix} -0.1749 & -0.6086 & 0.1902 \\ 0.2114 & -0.2245 & -0.6075 \\ 0.7795 & 0.0621 & 0.2059 \end{bmatrix}.$$

We simply choose the last row of blocks to make the scaling, getting $\boldsymbol{A}''_n$ such that $\mathbf{DET}(\boldsymbol{A}''_n) = \boldsymbol{I}$. Then use **BLUS** algorithm to obtain a block unit TERM factorization: $\boldsymbol{A}''_n = \boldsymbol{LUS}_0$, where

$$\boldsymbol{L} = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & & \\ \begin{bmatrix} 0.4047 & -0.1341 & -1.6488 \\ -0.6640 & -0.7095 & 1.1264 \\ 0.7010 & -0.1996 & 0.4091 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \\ \begin{bmatrix} 0.8616 & -0.4221 & -0.2200 \\ 0.1861 & -0.0297 & -1.7043 \\ -0.6688 & -0.5769 & 0.4268 \end{bmatrix} & \begin{bmatrix} 0.5125 & -0.3029 & -0.6593 \\ 1.9924 & 0.3798 & -0.5392 \\ -0.5579 & 2.1764 & -0.5790 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix},$$

$$\boldsymbol{U}=\begin{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}-0.6747&0.2574&-0.0448\\0.7987&1.2071&-0.9397\\0.8973&-0.9645&-0.3025\end{bmatrix}\begin{bmatrix}-0.4642&0.4082&-0.1612\\-0.3030&-0.4082&-0.4642\\-0.4430&-0.2357&0.3611\end{bmatrix}\\ \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}-0.1749&-0.6086&0.1902\\0.2114&-0.2245&-0.6075\\0.7795&0.0621&0.2059\end{bmatrix}\\ \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\end{bmatrix},$$

$$\boldsymbol{S}_0=\begin{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\\ \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\\ \begin{bmatrix}0.2728&0.8655&0.8138\\-1.6400&0.1963&1.1839\\0.2642&0.5371&-2.2249\end{bmatrix}\begin{bmatrix}0.6874&-0.6345&-1.0269\\1.5270&-0.1200&-0.5295\\-0.4183&3.1198&-0.5409\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\end{bmatrix}.$$

(ii) Scale after the factorization.

From Proposition 3, we have $\boldsymbol{A}_n = \boldsymbol{PDLUS}_0$, where $\boldsymbol{P}$, $\boldsymbol{L}$, $\boldsymbol{U}$, and $\boldsymbol{S}_0$ are all the same as above, while

$$\boldsymbol{D}=diag\left(\boldsymbol{I},\boldsymbol{I},\begin{bmatrix}1.0279&0.7223&0.0170\\0.6231&-0.6022&-0.4636\\0.1745&-0.1260&0.8250\end{bmatrix}\right).$$

The last term of $\boldsymbol{D}$ can be factorized further or be used to scale the last row of blocks. It is easy to verify that $\boldsymbol{D}^{-1}\boldsymbol{P}^T\cdot\boldsymbol{A}_n=\boldsymbol{A}_n''$.

Anyway, $\boldsymbol{LU}$ can be further factorized into a product of block SERMs: $\boldsymbol{LU} = \boldsymbol{S}_3\boldsymbol{S}_2\boldsymbol{S}_1$,

$$\begin{bmatrix} \boldsymbol{s}_1^T \\ \boldsymbol{s}_2^T \\ \boldsymbol{s}_3^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -0.6747 & 0.2574 & -0.0448 \\ 0.7987 & 1.2071 & -0.9397 \\ 0.8973 & -0.9645 & -0.3025 \end{bmatrix} & \begin{bmatrix} -0.4642 & 0.4082 & -0.1612 \\ -0.3030 & -0.4082 & -0.4642 \\ -0.4430 & -0.2357 & 0.3611 \end{bmatrix} \\ \begin{bmatrix} 0.4047 & -0.1341 & -1.6488 \\ -0.6640 & -0.7095 & 1.1264 \\ 0.7010 & -0.1996 & 0.4091 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -0.1749 & -0.6086 & 0.1902 \\ 0.2114 & -0.2245 & -0.6075 \\ 0.7795 & 0.0621 & 0.2059 \end{bmatrix} \\ \begin{bmatrix} 0.9153 & -0.6999 & 1.2358 \\ 0.0099 & 0.3992 & 1.3735 \\ 1.4080 & 0.7767 & -2.7077 \end{bmatrix} & \begin{bmatrix} 0.5125 & -0.3029 & -0.6593 \\ 1.9924 & 0.3798 & -0.5392 \\ -0.5579 & 2.1764 & -0.5790 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{bmatrix},$$

where $\boldsymbol{S}_i = \boldsymbol{I} + \boldsymbol{e}_i \boldsymbol{s}_i^T$ (see Section 1 for the meanings of $\boldsymbol{S}_i, \boldsymbol{e}_i, \boldsymbol{s}_i$). The corresponding block unit SERM factorization is $\boldsymbol{A}_n'' = \boldsymbol{S}_3 \boldsymbol{S}_2 \boldsymbol{S}_1 \boldsymbol{S}_0$.

This SERM factorization can be used for perfect reversible integer transform. For example, given an input signal [34 250 49 136 223 233 36 67 153]^T, the naïve DCT transformed value (after rounded to nearest integer) is [−58 8 90 −14 −103 394 −168 23 −66]^T. Similarly, we get the direct reconstruction vector [35 249 50 136 223 233 37 67 152]^T, which is not the exact original data. Information is lost in this process even if the DCT transform matrix itself is invertible. By contrast, our integer transform guarantees perfect reconstruction in its implementation; and with [−58 9 91 −15 −103 394 −168 22 −66]^T in the transform domain, the transform error is almost neglectable. Note that the number of SERMs in our new factorization relies on the number of blocks, but not the order of the original element matrix, which may achieve fewer SERMs in the end. Moreover, in our new SERM integer transform, all components can be completely parallel restored and BLAS can be used to speed up the block operations.

7 Conclusions

We summarize the conclusions obtained in this paper as follows:

(i) We define a 'determinant matrix function' **DET** mapping from block matrices to matrices, and obtain its important properties;

(ii) **DET** is a generalization of *det*, and *det* can be regarded as a special case of **DET**. When the blocks are of size 1-by-1, **DET** is just the value of *det*;

(iii) **DET**($\boldsymbol{A}_n$) exists if and only if all the quasi leading principal submatrices of $\boldsymbol{A}_n$ are invertible;

(iv) We presented **BLUS** block factorizations for any nonsingular matrix. Generally, any given nonsingular block matrix can be factorized into 3 block unit triangular matrices, besides some necessary permutations and the scaling operation. These conclusions cover the early discussions of optimal point matrix factorizations for integer transform.

Our block TERM and SERM factorizations can be used to realize perfect reconstruction (PR) in signal transform, and have attractive properties (like block computation and parallel reconstruction) advantageous to computation optimization and parallel design. It is worth mentioning that although we assume all blocks in a block matrix are of the same size, the **BLUS** factorization can be generalized to any nonsingular block matrix with its diagonal blocks increasing in size from top to bottom, if we give up the requirement of scaling flexibility. Finally, how to eliminate **DET** by permutations or other means merits further exploration.

Acknowledgements This work was supported by NKBRSF China (grant No. G1998030606) and the foundation for the authors of the National Excellent Doctoral Dissertation, China (grant No. 200038).

References

1. Blume, H., Frand, A., Reversible and irreversible image data compression using the S-transform and Lempel-Ziv coding, Proceedings of SPIE, 1989, 1091: 2—18.
2. Zandi, A., Allen, J., Schwartz, E. et al., CREW: Compression with reversible embedded wavelets, in: James, A. S., Martin, C. eds. Proceedings of IEEE Data Compression Conference, Snowbird: IEEE Computer Society Press, 1995: 212—221.
3. Said, A., Pearlman, W. A., An image multiresolution representation for lossless and lossy compression, IEEE Transactions on Image Processing, 1996 5(9): 1303—1310.
4. Gormish, M. J., Schwartz, E. L., Keith, A. F. et al., Lossless and nearly lossless compression for high quality images, Proceedings of SPIE, 1997, 3025: 62—70.
5. Bruekers, F. A. M. L., van den Enden, A. W. M., New networks for perfect inversion and perfect reconstruction, IEEE J. on Selected Areas in Communications, 1992, 10(1): 130—137.
6. Sweldens, W., The lifting scheme: A custom-design construction of biorthogonal wavelets, J. of Applied and Computational Harmonic Analysis, 1996, 3(2): 186—200.
7. Dewitte, S., Cornelis, J., Lossless integer wavelet transform, IEEE Signal Processing Letters, 1997, 4(6): 158—160.
8. Calderbank, A. R., Daubechies, I., Sweldens, W.. et al., Wavelet transform that map integers to integers, J. of Applied and Computational Harmonic Analysis, 1998, 5(3): 332—369.
9. Hao, P., Shi, Q., Invertible linear transforms implemented by integer mapping, Science in China, Series E (in Chinese), 2000, 30(2): 132—141.
10. Hao, P., Shi, Q., Matrix factorizations for reversible integer mapping, IEEE Trans. Signal Processing, 2001, 49(10): 2314—2324.
11. Hao, P., Shi, Q., Proposal of reversible integer implementation for multiple component transforms, ISO/IEC JTC1/SC29/WG1N 1720, Arles, France, July 3-7, 2000.
12. Qiu, W., Advanced Algebra (in Chinese), Beijing: Higher Education Press, 1996.