BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(3), 2022, Article number: e141005
DOI: 10.24425/bpasts.2022.141005

Effective lattice structures for separable two-dimensional orthogonal wavelet transforms

Dariusz PUCHALA*

Institute of Information Technology, Technical University of Lodz, Poland

Abstract. Discrete two-dimensional orthogonal wavelet transforms find applications in many areas of analysis and processing of digital images. In a typical scenario the separability of two-dimensional wavelet transforms is assumed and all calculations follow the row-column approach using one-dimensional transforms. For the calculation of one-dimensional transforms the lattice structures, which can be characterized by high computational efficiency and non-redundant parametrization, are often used. In this paper we show that the row-column approach can be excessive in the number of multiplications and rotations. Moreover, we propose the novel approach based on natively two-dimensional base operators which allows for significant reduction in the number of elementary operations, i.e., more than twofold reduction in the number of multiplications and fourfold reduction of rotations. The additional computational costs that arise instead include an increase in the number of additions, and introduction of bit-shift operations. It should be noted, that such operations are significantly less demanding in hardware realizations than multiplications and rotations. The performed experimental analysis proves the practical effectiveness of the proposed approach.

Key words: two-dimensional orthogonal discrete wavelet transform; lattice structures; image processing; VLSI circuits.

1. INTRODUCTION

Discrete two-dimensional orthogonal wavelet transforms find wide applications in image compression, image processing, classification of images, image watermarking, industrial informatics, etc. (e.g., see [1–9]). The typical way to calculate two-dimensional wavelet transforms is to assume their separability and take advantage of row-column approach where one-dimensional transformation is applied first to rows (columns) and then to columns (rows) of an input image. Thus for calculation of two-dimensional wavelet transforms we can use one-dimensional computational techniques. In recent years particular interest was laid in the development of adaptive techniques which by parametrization allow to adapt the form of wavelet transform according to the needs of the implemented task (see [10–14]). In such cases it is required to describe wavelet transform with a parametric model that can be characterized by non-redundant number of parameters. In this case the effective tool for calculation of one-dimensional (and also two-dimensional by the means of row-column approach) orthogonal transforms are lattice structures which, not only allow for the reduction of the number of additions and multiplications, but also can be characterized by accurate and non-redundant parametrization (see [15–21]).

Due to their popularity, discrete orthogonal wavelet transforms are hardware implemented in FPGA devices and also with use of CORDIC processors. For example, in papers [22–27] we can find the proposals of hardware implementations of one-dimensional orthogonal wavelet transforms for FPGA units and this task is further extended to the problem of two-dimensional data in papers [28–31]. It should be noted that in many practical solutions the authors propose basic implementations based on convolution approach (see [23, 26–29]). In some papers the simple example of Haar wavelet is being considered in both one- and two-dimensional cases (see [24, 31]). However, the FPGA implementations of discrete orthogonal wavelet transforms which are based on more computationally effective lattice structure and lifting based approach can be also found in papers [22, 25], and [30] respectively. For two-dimensional data the separability of transformations is being assumed and for their calculation the authors use the well-known row-column based approach (see [28, 29, 31]). Finally the studies regarding the possible implementations of filters or discrete wavelet transforms with CORDIC processor can be found in papers [32, 33].

In this paper we propose a novel approach to calculation of two-dimensional separable orthogonal wavelet transforms that takes advantage of natively two-dimensional base operators. The proposed approach allows for more than twofold reduction in the number of multiplications and fourfold reduction in the number of rotations when compared to the row-column approach based on one-dimensional lattice structures. The possible improvement depends on the type of operations needed, since rotations can be implemented by means of addition and multiplication blocks or with use of CORDIC processor. The additional costs that arise is an increase in the number of additions and the introduction of bit-shift operations. It should be noted that additions and bit-shift operations need significantly less resource requirements when implemented as logic circuits (bit-shift operations in most cases can be hardwired). The similar approach, but applied to the lifting scheme, is known in the

*e-mail: dariusz.puchala@p.lodz.pl

Manuscript submitted 2021-10-04, revised 2022-02-17, initially accepted for publication 2022-02-28, published in June 2022.

literature by the name of *Spatial Combinative Lifting Algorithm* (SCLA) [34]. The application of SCLA allowed to obtain more than twofold reduction in the number of multiplications for Cohen-Daubechies-Feauveau 9/7 biorthogonal wavelets both in two- and three-dimensional cases [34, 35].

2. ONE-DIMENSIONAL ORTHOGONAL WAVELET TRANSFORM AND LATTICE STRUCTURE

The one-dimensional discrete wavelet transform (DWT1D) in the practical tasks of digital signal processing and analysis is implemented in most cases as a two-channel bank of filters with the structure depicted in Fig. 1 (see [15–19]). When using the polyphase notation an input signal can be described in the following form $X(z) = X_0(z^2) + z^{-1}X_1(z^2)$, where by $X_0(z) = \sum_{n=0}^{N/2-1} x(n)z^{-n}$ and $X_1(z) = \sum_{n=0}^{N/2-1} x(n+1)z^{-n}$ we understand even and odd indexed samples of input data respectively. Then by analyzing the structure shown in Fig. 1, we can see that both an input signal and its delayed version ($z^{-1}X(z)$) are in the first place decimated at blocks ($\downarrow 2$) resulting in a vector of the form $[X_0(z), z^{-1}X_1(z)]^T$. The following step is the data filtering stage where the obtained vector is transformed at the block of filters described in a polyphase notation by a matrix $E(z)$. This matrix can be defined as follows:

$$E(z) = \begin{bmatrix} H_0(z) & H_1(z) \\ G_0(z) & G_1(z) \end{bmatrix}.$$

Here, with $H_0(z)$ and $G_0(z)$ we describe even indexed coefficients of impulse responses of filters, while $H_1(z)$ and $G_1(z)$ stand for odd indexed ones. It should be noted that both filters $H(z)$ and $G(z)$, in connection with decimators ($\downarrow 2$), form the analysis stage of a two-channel bank of filters. The two components $[V_0(z), V_1(z)]^T$, that we obtain at the output of the analysis stage, represent, respectively, the results of low-pass and high-pass filtering of input signal. The next step is the synthesis stage which is defined using polyphase notation by matrix $F(z)$. It is very important to select filters of both analysis and synthesis stages in a way that allows to satisfy the perfect reconstruction (PR) condition $F(z)E(z) = I$, where I is an identity matrix. Then, with this condition hold, it is possible to restore the input signal at the output of the synthesis stage, i.e., $\hat{X}(z) = X(z)$, with use of upsampling blocks ($\uparrow 2$), and under the assumption that the components obtained at the output of the analysis stage were undisturbed.

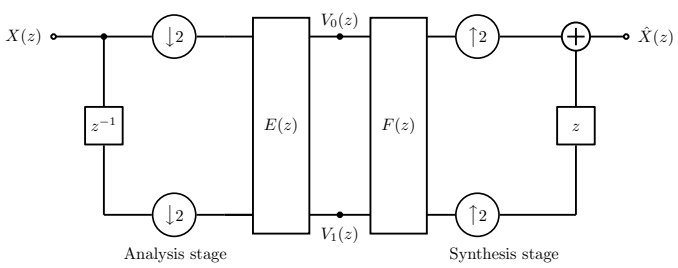

Fig. 1. The structure of a two-stage filter bank in polyphase notation

In the case of a bank of orthogonal filters $E(z)$ is an unitary matrix, which allows: (i) to obtain the synthesis stage filters directly from the following relation $F(z) = E^T(z^{-1})$, (ii) to determine the relationship between filters $H(z)$, $G(z)$ based on the PR condition, where we assume $G_0(z) = -z^{-(\frac{M}{2}-1)}H_1(z^{-1})$, and $G_1(z) = z^{-(\frac{M}{2}-1)}H_0(z^{-1})$, and that the PR condition for orthogonal filters $H_0(z)H_0(z^{-1}) + H_1(z)H_1(z^{-1}) = 1$ is satisfied. With parameter M we describe the size of filters. It should be noted that filter $G(z)$ is a reversed and modulated version of filter $H(z)$, which means that the coefficients of its impulse response can be calculated as $g(m) = -(-1)^m h(M-m-1)$ for $m = 0, 1, \ldots, M-1$. Hence, the whole bank of filters is fully described by a number of M coefficients $h(m)$ of the impulse response of filter $H(z)$.

The operations within a two-channel bank of filters are computationally effective and can be characterized by the linear complexity, i.e., $\mathcal{O}(MN)$, where N is the size of input data vectors. However, it should be noted that this efficiency can be increased even more by using the lattice structures. Lattice structures enable to reduce a number of multiplications and additions by almost two times. Moreover, they allow for non-redundant parametrization understood in the sense of a number of free parameters (see [15, 18]). In order to construct the lattice structure for a two-channel bank of filters it requires to factorize the polyphase matrix $E(z)$ into the product of simple matrices. In accordance to [15, 18] such factorization in its straightforward form (which is required for further considerations) can be described in the following way:

$$E(z) = \left(\prod_{i=1}^{\frac{M}{2}-1} A_{\frac{M}{2}-i} D \Lambda(z^{-1}) \right) A_0, \tag{1}$$

where the matrices used in formula (1) can be defined as:

$$A_i = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i \\ \sin\alpha_i & -\cos\alpha_i \end{bmatrix}, \quad \Lambda(z) = \begin{bmatrix} 1 & 0 \\ 0 & z \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Matrices A_i for $i = 0, 1, \ldots, \frac{M}{2} - 1$ represent rotation/reflection operations and are parametrized by rotation angles α_i. Taking into account the orthogonality of matrices A_i, and also the form of matrices $\Lambda(z)$ and D, it can be easily verified that $E(z)$ is unitary. The formula (1) represents the precise factorization of matrix $E(z)$ with a number of $\frac{M}{2}$ free parameters, which is the smallest possible number[1]. It can be also verified that the number of additions required by the lattice based on formula (1) is almost twofold smaller when compared to the approach based on convolution. However, if we want to reduce the number of multiplications, then the formula (1) must be modified by taking out two multipliers from each A_i matrix and moving them

[1] This is the smallest number of parameters for two-channel bank of orthogonal filters. In case of DWT1D this number equals $\frac{M}{2} - 1$ because one degree of freedom is used by the postulate requiring to ensure the low-pass characteristics of filter $H(z)$ (and high-pass characteristics of filter $G(z)$ at the same time).

forward to the supplementary diagonal matrix standing at the end of the factorization formula (c.f. [18]).

In Fig. 2 we can see an example of lattice structure based on factorization (1) for the size of input data $N = 8$ and the size of filters $M = 6$ (see [36]). It should be noted that the assumed boundary condition is a cyclic repetition of input data, which means that input signal is assumed to be periodical. Symbolically with '$\circ$' we describe base operators implementing operations within A_i matrices. It is straightforward to verify that for the given values of N and M all operations within the lattice structure from Fig. 2 can be described as $v = Ux$, where v is the output of an analysis stage and U is a matrix defined as follows:

$$U = \left(\prod_{i=1}^{\frac{M}{2}-1} U_{\frac{M}{2}-i} R \right) U_0 \tag{2}$$

with R describing cyclic rotation by one element and U_i being the block diagonal matrices with A_i operations on their main diagonals. Hence, the resulting matrix U is orthogonal by definition. Elements v_0 and v_1 of vector v refer to components $V_0(z)$ and $V_1(z)$ respectively.

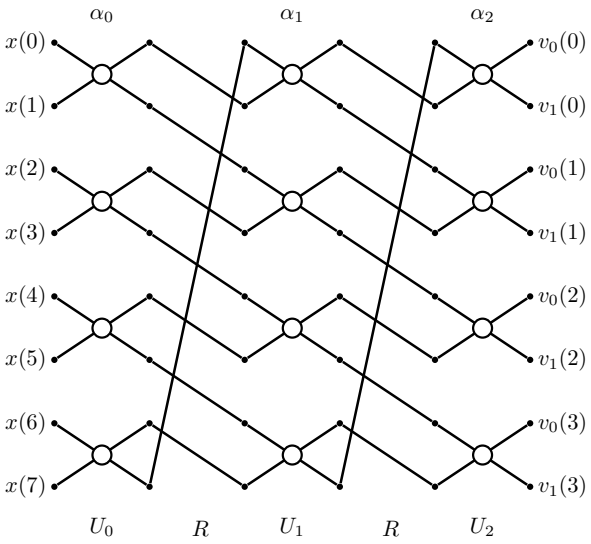

Fig. 2. The lattice structure of an analysis stage for one-dimensional bank of orthogonal filters for $N = 8$ and $M = 6$

3. TWO-DIMENSIONAL SEPARABLE ORTHOGONAL WAVELET TRANSFORMS

Let us assume that matrix X represents two-dimensional input data. Then two-dimensional separable wavelet transform (DWT2D) of matrix X can be calculated as $V = UXU^T$ in a row-column based approach with use of DWT1D described by matrix U. It should be noted that separability of two-dimensional wavelet transform is a basic assumption taken into account in practical applications (c.f. [28–31]). By substituting U with its definition (2) we can write further on:

$$V = \left(U_{\frac{M}{2}-1} R \ldots U_1 R U_0 \right) X \left(U_0 R^T U_1 \ldots R^T U_{\frac{M}{2}-1} \right), \tag{3}$$

since we have $U_i^T = U_i$ (c.f. $A_i^T = A_i$) for $i = 0, 1, \ldots, \frac{M}{2} - 1$. For example with $M = 6$ we obtain:

$$V = U_2 R U_1 R U_0 X U_0 R^T U_1 R^T U_2 .$$

Typically, in case of a row-column based approach first rows (columns) and then columns (rows) of input data are processed with the use of one-dimensional transforms in two consecutive steps. However, if we start to analyze the formula (3) it can be easily observed that the corresponding matrices coming from row and column transforms can be combined together into natively two-dimensional operators. In particular, we mean here the operations realized within block diagonal matrices U_i, where two-dimensional variant of such operations could relate to the $U_i X_i U_i$ product with X_i describing the input data for the i-th stage ($X_0 \equiv X$). It is worth noticing that such arrangement of separable one-dimensional filters is a special case of a four channel filter bank (c.f. [16, 17]).

In Fig. 3 we can see the operators of the i-th stage, i.e., the way they operate on square 2×2 element fragments of matrix X_i (see Fig. 3a), and how they should be interconnected to form natively two-dimensional operators (see Fig. 3b). Such two-dimensional operators A_i^{2D} are applied to 2x2 element fragments of input data and can be characterized by computational complexity strictly proportional to the number of one-dimensional operators used. However, it should be noted that all component operators are parametrized with the same angle of rotation α_i, which gives the possibility for further improvements. Operations RX_iR^T form additional stages consisting of shifting of elements of input matrix X_i. All the described stages form two-dimensional lattice structure for separable orthogonal wavelet transform.

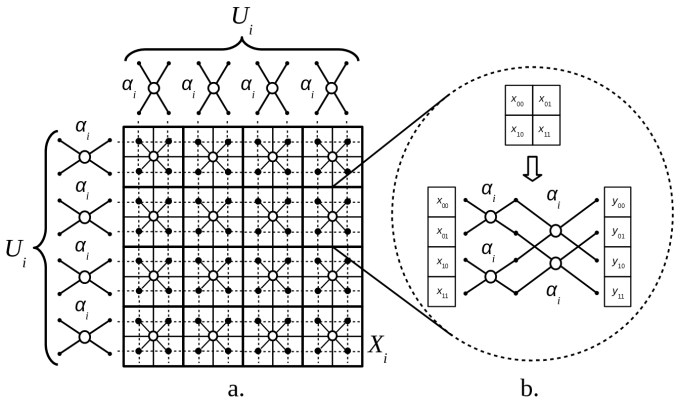

Fig. 3. Two-dimensional separable transform: (a) places of application and interaction between one-dimensional operators, (b) resulting basic two-dimensional base operator A_i^{2D}

In order to construct an inverse DWT2D it is required to reverse the order of stages of two-dimensional lattice structure and to inverse the operations within stages. It should be noted that $U_i^T = U_i$ for $i = 0, 1, \ldots, \frac{M}{2} - 1$ and the only operation that is not symmetrical is a cyclic rotation described by matrix R. Hence, the operation of inverse DWT2D can be described as a

product [34]:

$$\overline{X} = \left(U_0 R^T \ldots U_1 R^T U_{\frac{M}{2}-1}\right) V \left(U_{\frac{M}{2}-1} R U_1 \ldots R U_0\right). \quad (4)$$

The symmetry of U_i matrices results directly from the symmetry of A_i operations. The symmetry of A_i matrices guarantees the symmetry of A_i^{2D}. Further, the symmetry of A_i^{2D} matrices is very practical and it means that precisely the same implementations of those operations can be used to calculate both forward and inverse transforms.

4. OPTIMIZATION OF TWO-DIMENSIONAL BASE OPERATORS

The base operator A_i^{2D} transforms a vector of input data $x = [x_{00}, x_{01}, x_{10}, x_{11}]^T$ according to the formula $y = A_i^{2D} x$, into a vector $y = [y_{00}, y_{01}, y_{10}, y_{11}]^T$. Let $\mathrm{c}_i \equiv \cos(\alpha_i)$ and $\mathrm{s}_i \equiv \sin(\alpha_i)$. Then two-dimensional operator A_i^{2D} in its basic form can be defined as follows:

$$A_i^{2D} = \begin{bmatrix} \mathrm{c}_i & 0 & \mathrm{s}_i & 0 \\ 0 & \mathrm{c}_i & 0 & \mathrm{s}_i \\ \mathrm{s}_i & 0 & -\mathrm{c}_i & 0 \\ 0 & \mathrm{s}_i & 0 & -\mathrm{c}_i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{c}_i & \mathrm{s}_i & 0 & 0 \\ \mathrm{s}_i & -\mathrm{c}_i & 0 & 0 \\ 0 & 0 & \mathrm{c}_i & \mathrm{s}_i \\ 0 & 0 & \mathrm{s}_i & -\mathrm{c}_i \end{bmatrix}.$$

By multiplying the matrices in the above expression, we get:

$$A_i^{2D} = \begin{bmatrix} \mathrm{c}_i^2 & \mathrm{s}_i\mathrm{c}_i & \mathrm{s}_i\mathrm{c}_i & \mathrm{s}_i^2 \\ \mathrm{s}_i\mathrm{c}_i & -\mathrm{c}_i^2 & \mathrm{s}_i^2 & -\mathrm{s}_i\mathrm{c}_i \\ \mathrm{s}_i\mathrm{c}_i & \mathrm{s}_i^2 & -\mathrm{c}_i^2 & -\mathrm{s}_i\mathrm{c}_i \\ \mathrm{s}_i^2 & -\mathrm{s}_i\mathrm{c}_i & -\mathrm{s}_i\mathrm{c}_i & \mathrm{c}_i^2 \end{bmatrix}. \quad (5)$$

In this paper we proposed three variants of optimized factorizations of two-dimensional operators A_i^{2D} that can be characterized by different numbers and types of component operations.

Variant 1. The first variant of proposed factorizations allows to reduce the total number of arithmetical operations. In this case the output of base operation can be described as follows:

$$\begin{bmatrix} y_{00} \\ y_{01} \\ y_{10} \\ y_{11} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_2 \\ -t_1 \end{bmatrix} + \begin{bmatrix} x_{00} \\ -x_{01} \\ -x_{10} \\ x_{11} \end{bmatrix}, \quad (6)$$

where

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} -a_i & b_i \\ b_i & a_i \end{bmatrix} \cdot \begin{bmatrix} x_{00} - x_{11} \\ x_{01} + x_{10} \end{bmatrix},$$

and $a_i = s_i^2$, $b_i = c_i s_i$. This factorization allows to calculate $y = A_i^{2D} x$ with a number of 8 additions and 4 multiplications. The data flow diagram for the first variant of factorization of A_i^{2D} operator is shown in Fig. 4.

Variant 2. In the second variant of the proposed factorizations matrix A_i^{2D} is described as the product of sparce matrices:

$$A_i^{2D} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \overline{c}_i & 0 & \overline{s}_i \\ 0 & 0 & 1 & 0 \\ 0 & -\overline{s}_i & 0 & \overline{c}_i \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (7)$$

where $\overline{c}_i = \cos(\beta_i)$, $\overline{s}_i = \sin(\beta_i)$ and $\beta_i = 2\alpha_i + \pi$. It is simple to verify that in order to calculate $y = A_i^{2D} x$ with this variant of factorization it requires 8 additions, one rotation by an angle β_i, and 4 bit-shift operations (right by one bit) applied to the outputs of the operator. However, bit-shift operations can be potentially moved to the last stage of two-dimensional lattice structure and there implemented in the form of scaling by factor $2^{-M/2}$. It should be also noted that in hardware realizations bit-shift operations are trivial and can be implemented at the level of connections between data bus lines. In Fig. 5 we present the data flow diagram for the variant of factorization described by formula (7).

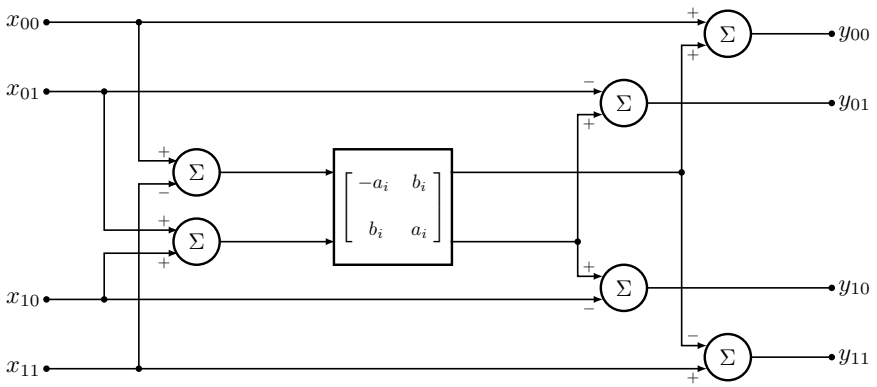

Fig. 4. The first variant of 2D operator based on matrix multiplication by vector multiplication (8 additions and 4 multiplications)

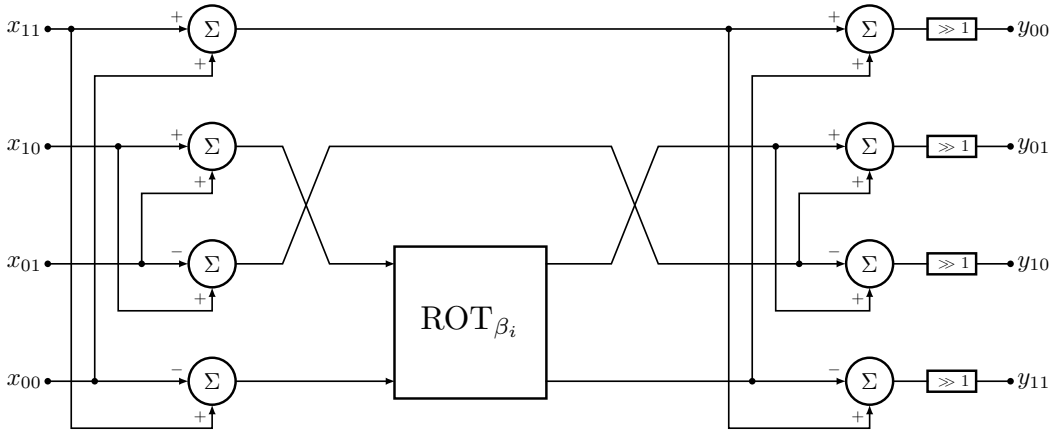

Fig. 5. The proposed variant of 2D operator based on rotation matrix (8 additions, 1 rotation and 4 bit-shift operations)

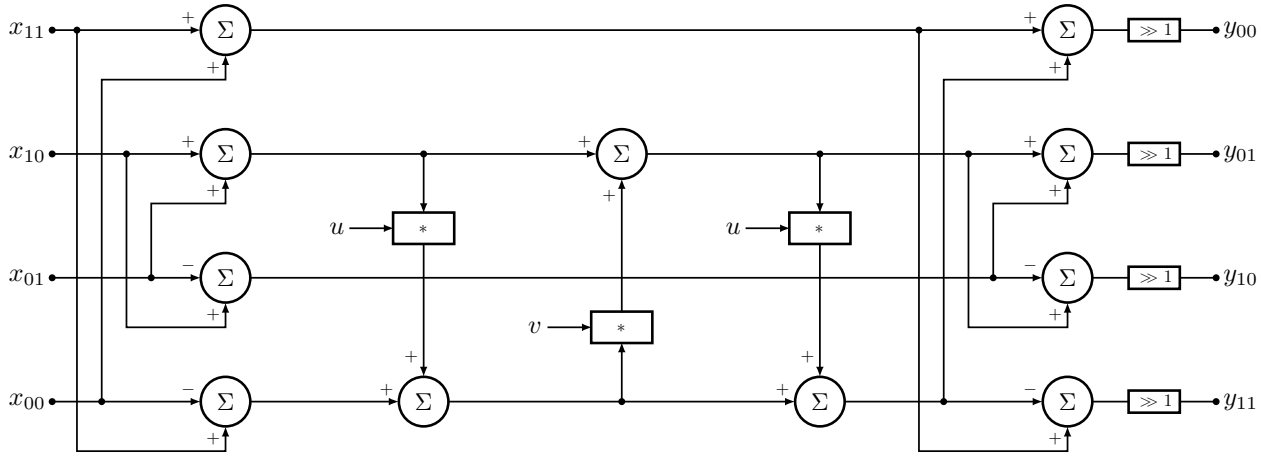

Fig. 6. The proposed variant of 2D operator based on lifting steps (11 additions, 3 multiplications and 4 bit-shift operations)

Variant 3. The last proposed variant of optimized factorization of operator A_i^{2D} is based on lifting steps. It is well know that operation of rotation can be factorized into a product of three lifting steps in the following way:

$$\begin{bmatrix} \bar{c}_i & \bar{s}_i \\ -\bar{s}_i & \bar{c}_i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ u_i & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & v_i \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ u_i & 1 \end{bmatrix}, \tag{8}$$

where $\bar{c}_i \equiv \cos(\beta_i)$ and $\bar{s}_i \equiv \sin(\beta_i)$. The lifting steps coefficients u_i and v_i are defined according to the following equations:

$$u_i = \frac{\cos(\beta_i) - 1}{\sin(\beta_i)}, \qquad v_i = \sin(\beta_i). \tag{9}$$

Taking into account formula (9) in factorization (8) we obtain the third variant of optimized two-dimensional operator with the data flow diagram of the form depicted in Fig. 6. The resulting factorization can be characterized by a number of 11 additions, 3 multiplications and 4 bit-shift operations. The reduction of one multiplication can be advantageous since multiplications are far more demanding in case of hardware realizations than additions. Moreover the explicit feature of lifting steps is the ability to perfectly reconstruct input image even if operations in vertical branches of the structure (here multiplication by coefficients u_i and v_i) are not invertible (e.g., because of the reduced binary representation of the resulting value). It should be noted that this variant may not be implemented directly for $\alpha_i = \frac{\pi}{2}k$ with k being an integer number. However, such cases are trivial and result in an identity or negative identity matrix in equation (8).

5. DISCUSSION OF COMPUTATIONAL COMPLEXITIES

In Table 1 we specify the numbers of arithmetic operations required to calculate two-dimensional separable wavelet transform in row-column approach with aid of various methods: (i) convolution based approach, (ii) lattice structures using A_i^{2D} operator in its basic form (see definition (5)), (iii) lattice structure based on tangent multipliers, (iv)–(vi) lattice structures based on three proposed optimized variants of the base operator. The results are expressed as the numbers of arithmetical operations (addition, multiplication, rotation, bit-shift operation) required for every 4 elements of input data.

By the convolution based method we understand one-dimensional convolution used to calculate two-dimensional wavelet

Table 1
Numbers of operations required for every 4 elements of input data

Computational method used	Additions	Multiplications	Rotation	Bit-shift operations
Convolution based approach	$8(M-1)$	$8M$	0	0
Lattice structure based on basic A_i^{2D} operator	$4M$	$8M$	0	0
Lattice structure based on tangent multipliers	$4M$	$4M+4$	0	0
Lattice structure with optimized operator in variant 1	$4M$	$2M$	0	0
Lattice structure with optimized operator in variant 2	$4M$	0	$\frac{1}{2}M$	$2M$
Lattice structure with optimized operator in variant 3	$5\frac{1}{2}M$	$1\frac{1}{2}M$	0	$2M$

transform in row-column approach. This method can be characterized by the highest total number of arithmetical operations. The second considered method is based on lattice structure using two-dimensional operator A_i^{2D} in its basic form resulting from applying one-dimensional lattice structure in row-column approach. In this case the number of additions can be reduced almost two times but the number of multiplications stays the same when compared to the convolution approach. It is well known that the cosine multipliers (i.e., c_i coefficients) can be moved outside each one-dimensional operator '$\circ$' leading to the following modification of operator A_i^{2D}:

$$\overline{A}_i^{2D} = c_i^2 \begin{bmatrix} 1 & t_i & t_i & t_i^2 \\ t_i & -1 & t_i^2 & -t_i \\ t_i & t_i^2 & -1 & -t_i \\ t_i^2 & -t_i & -t_i & 1 \end{bmatrix}, \tag{10}$$

where $t_i = s_i/c_i = \tan(\alpha_i)$ is a tangent multiplier. The scaling factor c_i^2 can be easily moved to the last stage of the lattice structure and there included into the values of operator coefficients at that stage. We can do the same operation at the following stages for $i = 1, 2, \ldots, \frac{M}{2} - 2$, i.e., except the last one. In this way we can obtain the lattice structure based on tangent multipliers which is characterized by almost twofold smaller number of multiplications when compared to the lattice structure based on operators A_i^{2D}. Of course this method may not be applied to $\alpha_i = \frac{\pi}{2} + k\pi$ for k being an integer number. But this includes trivial cases resulting in A_i^{2D} being an identity matrix.

It should be noted that the computational complexity of two-dimensional lattice structure can be reduced further on if we take into account the proposed variants of optimized base operators. With the first variant the number of multiplications can be reduced more by two times keeping the number of additions precisely at the same level. The number of multiplications can be reduced further more by 25% with the third variant of the operator. Here the number of additions increases and supplementary operations in the form of bit-shifts arise. It is well known, however, that additions are far less resource consuming when it comes to hardware realizations than multiplications and bit-shifts can be implemented as connections between the subsequent computational blocks. If we want to use the CORDIC processors in order to realize rotation operations the second variant would be the appropriate one. It allows to reduce the number of rotations required by the base operator from 4 to 1. Except one rotation it requires only additions (in the same number as lattice structure based on A_i^{2D} operator) and supplementary bit-shift operations.

6. EXPERIMENTAL RESULTS

The last part of research was focused on the comparison of both well-known operators, i.e., basic A_i^{2D} operator and operator with tangent multipliers, and the proposed optimized variants of two-dimensional operators in terms of selected aspects of their implementations in FPGA devices. The mentioned aspects include: number of logic cells required by the implementation, the length of the critical path, i.e., the time of signal propagation from input to the output of the operator along the longest path, and the representation accuracy. The number of logic cells describes a number of building elements required to implement the given variant of the operator and it is a measure of resource utilization in FPGA devices. The length of the critical path determines the time required from the moment the signal appears at the input until the correct response is obtained at the output of the circuit. It also defines the maximum allowable frequency of the clock that can be used to synchronize the calculations. It should be noted that in all of the considered cases the operators are implemented as a single stage of the computational pipeline in a form of a network of wired connections between basic blocks realizing addition and multiplication operations. Finally, the representation accuracy describes the accuracy of calculated results when compared to the results obtained with double floating-point precision. The calculations within operators were performed on the basis of integer arithmetic in a way limiting the accuracy of computations. Specifically, we assume that input data, i.e., elements x_{00}, x_{01}, x_{10} and x_{11}, are coded as signed 8-bit integers, i.e., numbers from -128 to 127, which corresponds to one component of image representation. The output values y_{00}, y_{01}, y_{10} and y_{11} are 12-bit signed integers, where the number of bits results directly from the number of operations realized within operators. It should be noted that in the performed experiment the values of operator parameters (i.e., a_i, b_i, c_i, s_i, $\bar{c}_i$, $\bar{s}_i$, t_i, and u_i, v_i) where coded with 8-bit precision as integer numbers in two's complement code. In order to measure the accuracy of output data representation we carried out a series of tests by changing the value of α_i according to the formula $\alpha_i = i\alpha_\mu$ for $i = 0, 1, \ldots, 359$ with $\alpha_\mu = \operatorname{atan}(2^{-\mu})$

Table 2
The experimental results obtained for Cyclone IV GX device for a single operator

Computational method used	No. of logical cells	Critical path length [ns]	Representation accuracy [dB]		
			$\mu = 0.5$	$\mu = 1.0$	$\mu = 1.5$
Lattice structure based on basic A_i^{2D} operator	2145	15.95	37.12	37.16	37.19
Lattice structure based on tangent multipliers	826	14.67	42.72	42.75	42.41
Lattice structure with optimized operator in variant 1	456	10.75	41.49	41.05	40.79
Lattice structure with optimized operator in variant 2	474	10.52	44.47	44.59	44.48
Lattice structure with optimized operator in variant 3	424	23.65	43.36	43.41	43.94

and $\mu > 0$, and by taking into account only the selected values of the resulting parameters (values in range $[-1, 1)$). For each value of α_i, and the resulting values of remaining parameters, a set of 100 input data vectors was randomly selected and the results obtained at the output of the operator where compared to the results obtained with double floating-point precision. The accuracy of representation was measured as the Signal-to-Noise Ratio (SNR) scaled in decibels. The results obtained with Cyclone IV GX FPGA device for three different values of parameter μ are collected in Table 2.

The implementation of operator A_i^{2D} in its basic form (see (5)) requires the highest number of logic cells. In the remaining cases those numbers are smaller and the optimized operator in variant 3 can be characterized by the smallest utilization of the device resources. Such results are fully expected since multiplications are the most resource consuming operations. In case of the length of the critical path the proposed operator in variant 3 requires the longest time of 23.65 ns to propagate the signal. It is a result of its specific structure composed of a sequence of three addition and three multiplication blocks (see Fig. 6). The smallest time of signal propagation is required by the proposed operator in variant 2, though the result obtained with the first variant is comparable. It is a direct consequence of the small number of multiplications and the compact structures of both operators (c.f. Figs. 4 and 5). The accuracy of output data is also a very important characteristic. In this case the operator in variant 2 (see (7)) can be characterized by the highest accuracy with a result close to 44.5 dB. It should be noted, that the second best results between 43 and 44 dB can be obtained with the proposed operator in variant 3. As so the proposed operator in variant 2 is a good compromise in the sense of the considered metrics. It can be characterized by reasonably small number of logic cells (though the highest among the proposed ones), high accuracy of representation of output data, and the shortest time of signal propagation.

7. CONCLUSIONS

In this paper we propose the novel optimized variants of lattice structures allowing for calculation of two-dimensional separable orthogonal wavelet transforms. The application of natively two-dimensional base operators allowed to obtain a significant reduction in the number of rotations (fourfold) or multiplications (more than twofold) depending on the variant of the base operator. The proposed optimizations were possible due to the redundant number of operations of rotation by the same angle resulting from the direct implementation of separable two-dimensional transforms in a row-column approach with use of one-dimensional transformations. The reduction in the number of multiplications and rotations caused an increase in the number of additions, and introduced additional bit-shift operations. However, the latter operations are much less demanding when it comes to hardware implementations in logic circuits, which allows for an overall improvement of efficiency. The obtained experimental results with Cyclone IV GX FPGA device show the advantage of the proposed operators in practical implementations. They can be characterized by significantly smaller (1.7–1.9 times) utilization of logic cells when compared to the operator using tangent multipliers and even 4.5–5.0 times smaller when compared to the basic two-dimensional operator. Moreover, the selected form of the proposed operators can be characterized by the shortest times of signal propagation (around 10 ns while tangent multipliers based operator required 14.67 ns and the basic operator even 15.95 ns) and allow to obtain good (more than 40 dB) accuracy of output data representation (significantly higher than basic operator but a little smaller in variant 1 than the accuracy obtained with operator using tangent multipliers).

REFERENCES

[1] J. Lin, "Reversible integer-to-integer wavelet filter design for lossless image compression," *IEEE Access*, vol. 8, pp. 89 117–89 129, 2020, doi: 10.1109/ACCESS.2020.2993605.

[2] T. Rahim *et al.*, "Basis pursuit with sparsity averaging for compressive sampling of iris images," *IEEE Access*, vol. 10, pp. 13 728–13 737, 2022.

[3] V. Manokar, R. Balaji, V. Nandhini, S. Shankar, and L. Patnaik, "Wavelet decomposition and classification of diseased fmri brain images using self organized maps," in *6th Inter. Conf. on Advanced Computing and Communication Systems*, 2020, pp. 1342–1348, doi: 10.1109/ICACCS48705.2020.9074422.

[4] X. Tang, L. Zhang, W. Zhang, X. Huang, V. Iosifidis, Z. Liu, M. Zhang, E. Messina, and J. Zhang, "Using machine learning to automate mammogram images analysis," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, pp. 757–764.

[5] J. Stolarek and P. Lipiński, "Improving Watermark Resistance Against Removal Attacks Using Orthogonal Wavelet Adaptation," in *Proc. of SOFSEM 2012: Theory and Practice of Computer Science*, 2012, pp. 588–599.

[6] P. Saravanan, M. Sreekara, and K. Manikantan, "Digital Image Watermarking using Daubechies wavelets," in *Proc. of SPIN 2016 3rd International Conference on Signal Processing and Integrated Networks*, 2016, pp. 57–62.

[7] P. Lipiński, "On domain selection for additive, blind image watermarking," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 60, no. 2, pp. 317–321, 2012.

[8] Z. Jakovljevic, R. Puzovic, and M. Pajic, "Recognition of Planar Segments in Point Cloud Based on Wavelet Transform," *IEEE Trans. Ind. Inf.*, vol. 11, no. 2, pp. 342–352, 2015.

[9] S.H. Shoron *et al.*, "A Watermarking Technique for Biomedical Images Using SMQT, Otsu, and Fuzzy C-Means," *Electronics*, vol. 8, no. 975, pp. 1–9, 2019.

[10] H. J. Wang, T. Chen, and S. Peng, "A novel method for designing adaptive compaction orthogonal wavelet filter banks," in *Proc. of 2003 International Conference on Image Processing*, 2003, pp. I–1041.

[11] M. Flierl, "Adaptive spatial wavelets for motion-compensated orthogonal video transforms," in *Proc. of 16th IEEE International Conference on Image Processing*, 2009, pp. 1045–1048.

[12] A. Sović and D. Seršić, "Robustly adaptive wavelet filter bank using L1 norm," in *Proc. of the International Conference on Systems, Signals and Image Processing*, 2011, pp. 1–4.

[13] J. Karel and R. Peeters, "Orthogonal Matched Wavelets with Vanishing Moments: A Sparsity Design Approach," *Circuits Syst. Sig. Process.*, vol. 37, no. 8, pp. 3487–3514, 2018.

[14] J. Stolarek, "Adaptive synthesis of a wavelet transform using fast neural network," *Bull.Pol. Acad. Sci. Tech. Sci.*, vol. 59, no. 1, pp. 9–13, 2011.

[15] P.P. Vaidyanathan and P.Q. Hoang, "Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 1, pp. 81–94, 1988.

[16] T. D. Tran, "*M*-Channel Linear Phase Perfect reconstruction Filter Bank With Rational Coefficients," *IEEE Trans. Circuits Syst. I. Fundam Theory Appl.*, vol. 49, no. 7, pp. 914–927, 2002.

[17] Y. Tanaka, M. Ikehara, and T.Q. Nguyen, "A Lattice Structure of Biorthogonal Linear-Phase Filter Banks With Higher Order Feasible Building Blocks," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 55, no. 8, pp. 2323–2331, 2008.

[18] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[19] D. Puchala and K. Stokfiszewski, "Highly effective gpu realization of discrete wavelet transform for big-data problems," *Comp. Sci. (ICCS)*, pp. 213–227, 2021.

[20] P. Rieder, K. Gerganoff, J. Götze, and J.A. Nossek, "Parametrization and Implementation of Orthogonal Wavelet Transforms," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Proc.*, vol. 3, 1996, pp. 1515–1518.

[21] P. Rieder, J. Götze, J.S. Nossek, and C.S. Burrus, "Parameterization of orthogonal wavelet transforms and their implementation," *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, vol. 45, no. 2, pp. 217–226, 1998.

[22] J. Jana, S. Tripathi, R.S. Chowdhury, A. Bhattacharya, and J. Bhaumik, "An area efficient vlsi architecture for 1-d and 2-d discrete wavelet transform (dwt) and inverse discrete wavelet transform (idwt)," *Devices Integr. Circuit*, pp. 378–382, 2021.

[23] J. Chilo and T. Lindblad, "Hardware Implementation of 1D Wavelet Transform on an FPGA for Infrasound Signal Classification," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 1, pp. 9–13, 2008.

[24] E.L. Chuma, L.G.P. Meloni, Y. Iano, and L.L.B. Roger, "FPGA implementation of a de-noising using Haar level 5 wavelet transform," in *XXXV Simposio Brasileiro De Telecomunicacoes E Processamento De Sinais*, 2017, pp. 1189–1192.

[25] V. Herrero, J. Cerda, R. Gadea, M. Martinez, and A. Sebastia, "Implementation of 1-D Daubechies Wavelet Transform on FPGA," pp. 1–5, 2008. [Online]. Available: http://www.upv.es/dsd/publica/articulos/paper192.pdf.

[26] C.S. Avinash and J.S.R. Alex, "FPGA Implementation of Discrete Wavelet Transform using Distributed Arithmetic Architecture," in *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials*, 2015, pp. 326–330.

[27] M. Bahoura and H. Ezzaidi, "Real-Time Implementation of Discrete Wavelet Transform on FPGA," in *Proc. of IEEE 10th International Conference On Signal Processing*, 2010, pp. 191–194.

[28] M. León, L. Barba, and C.O. Torres, "Implementation of the 2-D Wavelet Transform into FPGA for Image," *J. Phys. Conf. Ser.*, vol. 274, pp. 1–8, 2011.

[29] M. Katona, A. Pižurica, N. Teslić, V. Kovačević, and W. Philips, "A Real-Time Wavelet-Domain Video Denoising Implementation in FPGA," in *Proc. SPIE: Wavelet Applications in Industrial Processing II*, vol. 5607, 2004, pp. 63–70, doi: 10.1117/12.573783.

[30] G. Dillen, B. Georis, J.D. Legat, and O. Cantineau, "Combined Line-Based Architecture for the 5-3 and 9-7 Wavelet Transform of JPEG2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 9, pp. 944–950, 2003.

[31] S.A. Christe, M. Balaji, and S.S. Kumar, "FPGA Implementation of 2-D Wavelet Transform of Image Using Xilinx System Generator," *Int. J. Appl. Eng. Res.*, vol. 10, no. 29, pp. 22 463–22 466, 2015.

[32] S. Simon, P. Rieder, C. Schimpfle, and J.A. Nossek, "CORDIC-based architectures for the efficient implementation of discrete wavelet transforms," in *Proc. of Inter. Symposium on Circuits and Systems*, 1996, pp. 77–80.

[33] P. Poczekajło, "An overview of the methods of synthesis, realization and implementation of orthogonal 3-d rotation filters and possibilities of further research and development," *Int. J. Electron. Telecommun.*, vol. 67, no. 2, pp. 295–300, 2021.

[34] J. Huang, R. Zhu, and J. Li, "Spatial Combinative Lifting Algorithm-Based Block Wavelet Transform for Image Compression," in *Proc. of the 2nd IEEE Conf. on Industrial Electronics and Applications*, 2007, pp. 2516–2521.

[35] L. Dai, L. Zhang, and X. Zhao, "A Three Dimensional Combinative Lifting Algorithm for Wavelet Transform Using 9/7 Filter," in *Data Compression Conference*, 2008, pp. 510–510.

[36] M. Yatsymirskyy and K. Stokfiszewski, "Effectiveness of lattice factorization of two-channel orthogonal filter banks," in *Proc. of Joint Conference New Trends in Audio & Video and Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, 2012, pp. 275–279.