2-D non-separable integer implementation of paraunitary filter bank based on the quaternionic multiplier block-lifting structure

Eugene V. Rybenkov*, Nick A. Petrovsky†
Department of computer engineering,
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: { *rybenkov, †nick.petrovsky }@bsuir.by

***Abstract*—This paper presents a novel technique of factorization for 2-D non-separable quaternionic paraunitary filter banks (2-D NSQ-PUFB) based on the integer-to-integer invertible quaternionic multipliers. Two-dimensional factorization structures called "16in-16out" and "64in-64out" respectively for 4-channel and 8-channel Q-PUFB based on the proposed technique are shown. Comparison of the energy compaction level between the 2-D separable Q-PUFB based on the 1D Q-PUFB (8×24 Q-PUFB one-dimensional coding gain is $CG_{1D} = 9.38\,\mathrm{dB}$) and 2D non-separable Q-PUFB (8×24 2D-NSQ-PUFB, multi-dimensional coding gain is $CG_{MD} = 17.15\,\mathrm{dB}$) for the Barbara image shows that the 2-D non-separable Q-PUFB generates a higher percentage of small-value coefficients, hence creates a significant increase in the number of zero trees. This holds the key to our coder's superior performance.**

***Index Terms*—quaternionic paraunitary filter banks, two-dimensional, non-separable transform**

I. Introduction

Quaternions, four-dimensional hypercomplex numbers, have recently found application in image processing [1]–[3]. In particular, they are used as a basis for novel transforms, whose elementary operation is quaternion multiplication in which one of operands is a constant with unit magnitude. In this context, the authors have recently [4] presented a new concept of a quaternionic building block applicable to many existing structures of linear phase paraunitary filter banks (Q-PUFB) and transforms, especially to the 4- and 8-channel ones, commonly used in imaging applications. The main results were: structurally guaranteed perfect reconstruction (up to scaling) under a rough coefficient quantization, reduced memory requirements, and good suitability for FPGA and VLSI implementations, mitigating the disadvantage of increased computational complexity non-critical [5].

One-dimensional linear phase PUFB's can be applied to the construction of multidimensional separable systems. 2-D signals (images) are separately transformed along vertical and horizontal directions. However, multidimensional signals are generally non-separable, and this approach does not exploit their characteristics effectively. 2-D non-separable filter banks (FBs) perform more efficiently for image coding than separable FBs, because non-separable FBs may have better frequency characteristics [6], [7]. Suzuki et al. proposed a lattice structure of the 2-D non-separable perfect reconstruction FBs and showed their efficiency for lossy-to-lossless image coding [8].

Taking into account the advantages of the Q-PUFB in Section II, the aim of this contribution is to show a novel technique of factorization for 2-D non-separable quaternionic paraunitary filter banks (2-D-NSQ-PUFB) described in Section III, in Section IV we investigate structural transformations which facilitate finite-precision implementation of block lifting-based quaternion multipliers and evaluate their performance in Section V.

II. Summary of the Q-PUFB

As shown in [4], quaternions are especially suited to the parameterization of 4×4 orthogonal matrices. Namely, every matrix belonging to special orthogonal group $SO(4)$, can be represented as a product of left and right unit quaternions P and Q ($|P| = 1$ and $|Q| = 1$) $\underset{\mathbf{R}\in SO(4)}{\forall}\ \underset{P,Q\in\text{unit quat.}}{\exists}\ \mathbf{R} = \mathbf{M}^{+}(P)\cdot\mathbf{M}^{-}(Q) = \mathbf{M}^{-}(Q)\times\mathbf{M}^{+}(P)$ directly (contrary to Givens rotations) to preserve their orthogonality in spite of quantization. A quaternionic critically sampled linear phase filter bank with pairwise-mirror-image (PMI) symmetric frequency responses PMI LP PUFBs results from substitution ($\mathbf{E}(z)$ is paraunitary polyphase transfer matrices of an analysis filter bank) [9]:

$$\mathbf{E}(z) = \mathbf{G}_{N-1}\mathbf{G}_{N-2}\dots\mathbf{G}_1\mathbf{E}_0, \tag{1}$$

$$\mathbf{E}_0 = (2^{-0.5})\mathbf{\Phi}_0\mathbf{W}_M\,\mathrm{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right),$$

$$\mathbf{G}_i = (2^{-1})\mathbf{\Phi}_i\mathbf{W}_M\mathbf{\Lambda}_M(z)\mathbf{W}_M,\ \ i = \overline{1, N-1},$$

$$\mathbf{W}_M = \begin{bmatrix}\mathbf{I}_{M/2} & \mathbf{I}_{M/2}\\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2}\end{bmatrix};\ \mathbf{\Lambda}_M(z) = \mathrm{diag}\left(\mathbf{I}_{M/2}, z^{-1}\mathbf{I}_{M/2}\right),$$

where N is order of the factorization; $\mathbf{I}_{M/2}$ and $\mathbf{J}_{M/2}$ denote the $M/2 \times M/2$ identity and reversal matrices, respectively; $i = \overline{1, N-1}$ is interval notation $1 \le i \le N-1$.

A 4-channel PMI LP Q-PUFB realized according to the following factorization of the matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{N-1}$:

$$\mathbf{\Phi}_i = \mathbf{M}^{+}(P_i) \tag{2}$$

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

$$\mathbf{\Phi}_{N-1} = \mathbf{M}^{+}\left(P_{N-1}\right) \cdot \operatorname{diag}\left(\mathbf{J}_{M/2} \cdot \mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right),$$

where $\mathbf{\Gamma}_{M/2}$ is diagonal matrix the elements of which are defined as $\gamma_{mm} = (-1)^{m-1}$, $m = \overline{1, M-1}$.

The corresponding factorization of the matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{N-1}$ for an 8-channel PMI LP Q-PUFB is shown in [4].

III. 2D NON-SEPARABLE Q-PUFB

A. 2D non-separable transform

In general case 1D transformation can be formulated as follows: $\mathbf{y}_{n,n} = \mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n}$, where $\mathbf{\Theta}_{n,n}$ is conversion matrix, whose size is $n \times n$); $\mathbf{y}_{n,n}$ is transform result $n \times n$, $\mathbf{x}_{n,n}$ is block of input signal size $n \times n$. The two-dimensional transform based on the orthogonal transform $\mathbf{\Theta}_{n,n}$ applied to 2D input signal $\mathbf{x}_{n,n}$ separately by column and row is expressed by

$$\mathbf{y}_{n,n} = \mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n} \cdot \mathbf{\Theta}_{n,n}^{T}. \tag{3}$$

Comparing the 1D and 2D transformations, we can note that the 2D transform is performed over a size signal $n \times n$, it is executed in blocks. Also, in order to perform a 2D transform, it is needed to obtain an intermediate result $\mathbf{x}_{n,n}\mathbf{\Theta}_{n,n}^{T}$, which requires additional memory of size $n \times n$.

The given block of input signal $\mathbf{x}_{n,n}$ can be transformed into vector $\mathbf{x}_{n\cdot n,1}$ and $\mathbf{z}_{n\cdot n,1}$ respectively as follows

$\mathbf{x}_{n\cdot n,1} = \operatorname{tv}\left(\mathbf{x}_{n,n}\right)$:

$$\left[x_{1,1} \ldots x_{1,n} \ldots x_{n,1} \ldots x_{n,n}\right]^{T} \underleftarrow{\operatorname{tv}} \left[\mathbf{x}_{n,n}\right];$$

$\mathbf{z}_{n\cdot n,1} = \operatorname{tv}\left(\mathbf{x}_{n,n}^{T}\right)$:

$$\left[x_{1,1} \ldots x_{n,1} \ldots x_{1,n} \ldots x_{n,n}\right]^{T} \underleftarrow{\operatorname{tv}} \left[\mathbf{x}_{n,n}{}^{T}\right].$$

The transformation **tv** only performs a line-by-line mapping of the matrices $\mathbf{x}_{n,n}$, $\mathbf{x}_{n,n}^{T}$ into vectors $\mathbf{x}_{n\cdot n,1}$ and $\mathbf{z}_{n\cdot n,1}$. On the basis of given definitions, the vector $\mathbf{z}_{n\cdot n,1} = \mathbf{P} \cdot \mathbf{x}_{n\cdot n,1} = = \mathbf{P} \cdot \operatorname{tv}\left(\mathbf{x}_{n,n}\right)$, where $\mathbf{P}$ is the permutation matrix of size $(n^2 \times n^2)$, and thus

$$\operatorname{tv}\left(\mathbf{x}_{n,n}^{T}\right) = \mathbf{P} \cdot \operatorname{tv}\left(\mathbf{x}_{n,n}\right) = \mathbf{P} \cdot \mathbf{x}_{n\cdot n,1}. \tag{4}$$

Two-dimensional transform (3) can be rewritten as

$$\mathbf{y}_{n,n} = \mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n} \cdot \mathbf{\Theta}_{n,n}^{T} = \mathbf{\Theta}_{n,n} \cdot \left(\mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n}^{T}\right)^{T}. \tag{5}$$

Based on the definitions below, a 2D transformation result $\mathbf{y}_{n,n}$ can be represented as the vector:

$$\mathbf{y}_{n\cdot n,1} = \mathfrak{D}\left(\mathbf{\Theta}\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{\Theta}\right) \cdot \underbrace{\mathbf{P} \cdot \mathbf{x}_{n\cdot n,1}}_{\mathbf{x}_{n,n}^{T}} = \ddot{\mathbf{\Theta}}_{n^2,n^2} \cdot \mathbf{x}_{n\cdot n,1},$$

$$\underbrace{\underbrace{\underbrace{\mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n}^{T}}_{\mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n}^{T}}}_{\left(\mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n}^{T}\right)^{T}}}_{\mathbf{\Theta}_{n,n} \cdot \mathbf{x}_{n,n} \cdot \mathbf{\Theta}_{n,n}^{T}}$$

where $\mathfrak{D}\left(\mathbf{\Theta}\right)$ denotes the matrix with transform matrices $\mathbf{\Theta}_{n,n}$ on the main diagonal (the number of matrices $\mathbf{\Theta}_{n,n}$ is n), $\mathfrak{D}\left(\mathbf{\Theta}\right) = \operatorname{diag}\underbrace{\left(\mathbf{\Theta}, \ldots, \mathbf{\Theta}\right)}_{n \text{ times}} = \mathbf{I}_n \otimes \mathbf{\Theta}_{n,n}$ also can be described using Kronecker tensor product $\otimes$ with identity matrix $\mathbf{I}_n$; $\ddot{\mathbf{\Theta}}_{n^2,n^2}$ is the 2D transformation matrix.

It may seem that the complexity of the hardware implementation of the given conversion has increased, but this is not so. The calculation of $\mathbf{y}_{n,n}$ requires performing $(2 \cdot n^2)$ vector multiplication operations, which is equivalent for the case of $\mathbf{y}_{n\cdot n,1}$ (if we neglect multiplication by 0). Multiplication by the matrix $\mathbf{P}$ is the commutation of the input signal, which does not incur additional hardware costs.

B. Two-dimensional non-separable PMI LP Q-PUFB

When a factorization of PMI LP Q-PUFB matrix $\mathbf{E}$ (5) is applied to a 2D input signal $\mathbf{x}_{n,n}$ in horizontal and vertical directions, the output signal $\mathbf{y}_{n,n}$ is expressed as

$$\begin{aligned}\mathbf{y}_{n,n} = \mathbf{E} \cdot \mathbf{x}_{n,n} \cdot \mathbf{E}^{T} = \mathbf{G}_{N-1} \ldots \mathbf{G}_1 \mathbf{E}_0 \mathbf{x}_{n,n} \times \\ \times \mathbf{E}_0^{T} \mathbf{G}_1^{T} \ldots \mathbf{G}_{N-1}^{T}.\end{aligned} \tag{6}$$

On the other hand

$$\begin{aligned}\mathbf{y}_{n\cdot n,1} = \ddot{\mathbf{E}} \cdot \mathbf{x}_{n\cdot n,1} = \ddot{\mathbf{G}}_{N-1}(z)\ddot{\mathbf{G}}_{N-2}(z) \cdot \ldots \\ \ldots \cdot \ddot{\mathbf{G}}_1(z) \cdot \ddot{\mathbf{E}}_0 \cdot \mathbf{x}_{n\cdot n,1},\end{aligned} \tag{7}$$

where $\ddot{}$ denotes the 2D-transformation matrix. The underscore in equation (12) shows the sequence of matrix replacement to obtain equation (7). This Eq. (6) means that the 2D implementation of $\mathbf{G}_k$ is performed after that of $\mathbf{G}_{k-1}$ $(1 \le k \le N-1)$, i.e., the matrices $\mathbf{W}_M$, $\mathbf{\Lambda}_M(z)$, $\mathbf{M}^{+}\left(P\right)$ can be operated separately. The resulting representations of the matrices $\mathbf{W}_M$, $\mathbf{\Lambda}_M(z)$ as 2D transform are

$$\ddot{\mathbf{W}} = \mathfrak{D}\left(\mathbf{W}_M\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{W}_M\right) \cdot \mathbf{P}, \tag{8}$$

$$\ddot{\mathbf{\Lambda}}(z) = \mathfrak{D}\left(\mathbf{\Lambda}_M(z)\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{\Lambda}_M(z)\right) \cdot \mathbf{P}, \tag{9}$$

$$\ddot{\mathbf{G}}(z) = \mathfrak{D}\left(\mathbf{G}(z)\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{G}(z)\right) \cdot \mathbf{P}. \tag{10}$$

For a 4-channel PMI LP Q-PUFB ($M = 4$), the two-dimensional analogues of the matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{N-1}$ are defined as follows:

$$\begin{aligned}\ddot{\mathbf{\Phi}}_i &= \mathfrak{D}\left(\mathbf{M}^{+}\left(P_i\right)\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{M}^{+}\left(P_i\right)\right) \cdot \mathbf{P}, \\ \ddot{\mathbf{\Phi}}_{N-1} &= \ddot{\mathbf{\Phi}}_i \cdot \ddot{\mathbf{S}}_1, \quad \ddot{\mathbf{S}}_1 = \mathfrak{D}\left(\mathbf{S}_1\right) \cdot \mathbf{P} \cdot \mathfrak{D}\left(\mathbf{S}_1\right) \cdot \mathbf{P}, \\ \mathbf{S}_1 &= \operatorname{diag}\left(\mathbf{J}_{M/2} \cdot \mathbf{\Gamma}_{M/2}, \mathbf{I}_{M/2}\right).\end{aligned} \tag{11}$$

Thus, the factorization components (7) of the 2-D non-separable 4-channel PMI LP Q-PUFB, whose prototype filter is given by the relations (2), are represented by the following expressions (8-11).

The structure of the critically sampled 2-D non-separable 4-channel PMI LP Q-PUFB for $N = 2$ in according with the factorizations (8-11) .The given factorization structure of 2-D non-separable PMI LP Q-PUFB will be call "16in-16out" and for two dimensional 8-channel Q-PUFB — "64in-64out". The analysis of the 2-D non-separable PMI LP Q-PUFB circuit shows that it can be mapped to parallel-pipeline processor structures with a minimum time of latency $2N$ quaternion multiplication operations, where N is transformation order of the PMI LP Q-PUFB. It should be noted that the latency of parallel-pipeline processing does not depend on the size

$$\mathbf{y}_{n,n} = \ldots \cdot (2^{-0.5}) \cdot \mathbf{\Phi}_0 \cdot \underbrace{\mathbf{W} \cdot \underline{\mathrm{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right)} \cdot \mathbf{x}_{n,n} \cdot \underline{\mathrm{diag}\left(\mathbf{I}_{M/2}, \mathbf{J}_{M/2}\right)^T} \cdot \mathbf{W}^T}_{} \cdot \mathbf{\Phi}_0^T \cdot (2^{-0.5}) \cdot \ldots \tag{12}$$

of the original image in contrast to the conventional 2-D transform (3).

IV. The integer implementation of the 2-D-NSQ-PUFBs

Analysis of the equation (2) have shown that quaternion multiplier (Q-MUL) with different multiplication matrices: $\mathbf{M}^+(Q)$, $\mathbf{M}^-(Q)$, $\mathbf{M}^+\left(\overline{Q}\right)$, $\mathbf{M}^-\left(\overline{Q}\right)$ is an important building block for Q-PUFB. The mentioned algorithm use quaternion multiplications in which one of the operands is a constant with unit magnitude. Q-MULs are required to be invertible transforms, i.e. integer-to-integer transform, to apply 2-D NSQ-PUFB for lossless image coding. The Q-MUL circuit should be more hardware efficient than simply switching between four different circuits. So, Q-MUL should be versatile quaternion multiplier. The most important design parameters are data throughput, resource consumption and arithmetic precision. The memory-efficient and high-performance quaternion multiplier architecture based on the ROM-based DA modules and 4D CORDIC algorithm for multiplying quaternions are presented in [10] and [11] respectively. The main disadvantages of the given ROM-based DA quaternion multiplier and 4D CORDIC scheme [11] are the following: those are not invertible and not applicable to L2L image coding directly.

A. The block-lifting quaternionic multiplier structure as integer-to-integer transform

Block-lifting-based quaternion multiplication operator can be rewritten as follows [12]:

$$\mathbf{M}^+(Q) = \pm \begin{bmatrix} \mathbf{C}(Q) & -\mathbf{S}(Q) \\ \mathbf{S}(Q) & \mathbf{C}(Q) \end{bmatrix},$$

$$\mathbf{C}(Q) = \begin{bmatrix} q_1 & -q_2 \\ q_2 & q_1 \end{bmatrix}, \mathbf{S}(Q) = \begin{bmatrix} q_3 & q_4 \\ q_4 & -q_3 \end{bmatrix}.$$

$$\mathbf{M}^+(Q) = \pm \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{F}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(Q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(Q) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(Q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{H}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(Q)}.$$

$$\mathbf{M}^-(Q) = $$

$$= \pm \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}_2\mathbf{H}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(Q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(Q)\mathbf{\Gamma}_2 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}_2\mathbf{F}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(q)}.$$

Multiplication operators $\mathbf{M}^+(Q)$ and $\mathbf{M}^-(Q)$ can be considered as an extension of 2D rotation in the case of 4D. A set of matrix equations can be defined for a given unit-norm hypercomplex coefficient Q and multiplication matrix. At the same time the set can be solved uniquely for $\mathbf{F}(Q)$, $\mathbf{G}(Q)$, and $\mathbf{H}(Q)$, provided that $\mathbf{S}(Q)$ is nonsingular, or more specifically, nonzero: $\mathbf{F}(Q) = (\mathbf{C}(Q) - \mathbf{I}_2)\mathbf{S}(Q)^{-1}$, $\mathbf{G}(Q) = = \mathbf{S}(Q)$, $\mathbf{H}(Q) = \mathbf{S}(Q)^{-1}(\mathbf{C}(Q) - \mathbf{I}_2)$. Its elements represent real-valued lifting coefficients. The scheme comprises only 12 multiply-add operations, which form a sequence of lifting steps. Besides the computational load reduces.

Thus, the structure of the reversible multiplication operator of quaternions based on the block lifting circuit parametrization is a cascade connection of block matrices $\mathbf{F}(Q)$, $\mathbf{G}(Q)$, and $\mathbf{H}(Q)$, and their inverse inclusion with a negative sign (the inverse transform is the multiplication by the conjugate quaternion $\overline{Q}$, because $\mathbf{M}^+\left(\overline{Q}\right) = \mathbf{M}^+(Q)^T$). In this case, round-off errors of the corresponding multiplication results are mutually compensated by sections of direct and inverse transformations. The input vector $\mathbf{x}$ passes through three stages of processing $\mathbf{F}(Q)$, $\mathbf{G}(Q)$, and $\mathbf{H}(Q)$, to produce $\mathbf{M}^+(Q)\,\mathbf{x}$ (see Fig. 1).

However block-lifting coefficients $\mathbf{F}(Q)$, $\mathbf{G}(Q)$, and $\mathbf{H}(Q)$ can not fit the dynamic range $[-1, 1]$ of fixed point number, it can be solved using modifications of original quaternion $\widetilde{Q}$, whose lifting coefficients of multiplication operator $\mathbf{M}^{\pm}\left(\widetilde{Q}\right)$ are in dynamic range $[-1, 1]$ [12].

$$\mathbf{M}^+(Q) = \begin{cases} \mathbf{P}_{post} \cdot \mathbf{M}^+\left(\widetilde{Q}\right) \cdot \mathbf{P}_{pre}, & \text{if } \det(\mathbf{P}_Q) = 1, \\ \mathbf{P}_{post} \cdot \mathbf{M}^-\left(\widetilde{Q}\right) \cdot \mathbf{P}_{pre}, & \text{if } \det(\mathbf{P}_Q) = -1, \end{cases}$$

where $\widetilde{Q}$ is a quaternion whose coefficients are rearranged in some way, for example, $Q = q_1 + q_2 i + q_3 j + q_4 k$ and $\widetilde{Q} = = -q_3 - q_1 i + q_4 j + q_2 k$; in same time $\widetilde{Q} = \mathbf{P}_Q \cdot Q$ where $\mathbf{P}_Q$ is a quaternion permutation matrix, while for multiplication operator $\mathbf{M}^{\pm}(Q)$ matrices $\mathbf{P}_{post}$, $\mathbf{P}_{pre}$ are used for pre- and post-processing (see Fig. 1).

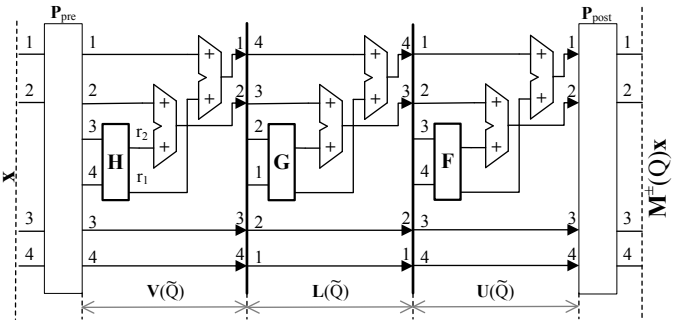


Fig. 1. Q-MUL block-lifting structure for the $\mathbf{M}^+(Q)$

In order to unify the quaternion multiplier structure, only left multiplication $\mathbf{M}^+(Q)$ can be used, to adjust it to the required multiplication operator $\mathbf{M}^+\left(\overline{Q}\right)$, $\mathbf{M}^-(Q)$ or $\mathbf{M}^-\left(\overline{Q}\right)$ following rules for modifying block-lifting matrix coefficients $\mathbf{F}(Q)$, $\mathbf{G}(Q)$, $\mathbf{H}(Q)$ and permutation matrices $\mathbf{P}_{pre}$, $\mathbf{P}_{post}$ are described in Table I.

Taking into account $\mathbf{M}^{+}\left(\overline{Q}\right) = \mathbf{M}^{+}(Q)^{T}$, block-lifting structure for modified quaternion $\widetilde{Q}$ for target operator $\mathbf{M}^{+}\left(\overline{Q}\right)$ (rule 2), defines as follows:

$$\mathbf{M}^{+}\left(\overline{Q}\right) = \mathbf{P}_{pre}^{T} \cdot \mathbf{V}^{T}(Q) \cdot \mathbf{L}^{T}(Q) \cdot \mathbf{U}^{T}(Q) \cdot \mathbf{P}_{post}^{T}.$$

But one can notice that $\mathbf{M}^{+}\left(\overline{Q}\right) = -\mathbf{P}_{Q} \cdot \mathbf{M}^{+}(Q) \cdot \mathbf{P}_{Q}$, where $\mathbf{P}_{Q} = \mathbf{\Gamma}_{4}\mathbf{J}_{4}$ and $\mathbf{P}_{Q}^{2} = -\mathbf{I}_{4}$. Thus, the rule 2 is proved as:

$$\mathbf{M}^{+}\left(\overline{Q}\right) = \mathbf{P}_{pre}^{T} \times$$
$$\times \begin{bmatrix} \mathbf{I}_2 & -\mathbf{H}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G}(Q) & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & -\mathbf{F}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \cdot \mathbf{P}_{post}^{T}.$$

Based on the quaternion multiplication properties $\mathbf{M}^{-}(Q) = \mathbf{D}_C \cdot \mathbf{M}^{+}(Q)^{T} \cdot \mathbf{D}_C = \mathbf{D}_C \cdot \mathbf{M}^{+}\left(\overline{Q}\right) \cdot \mathbf{D}_C$, where $\mathbf{D}_C = \mathrm{diag}\,(1, -\mathbf{I}_3)$. Taking into account above equation and rules 3 form Table I is:

$$\mathbf{M}^{-}(Q) = \mathbf{D}_C \cdot \mathbf{P}_{pre}^{T} \cdot \begin{bmatrix} \mathbf{I}_2 & -\mathbf{H}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \times$$
$$\times \begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G}(Q) & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & -\mathbf{F}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \cdot \mathbf{P}_{post}^{T} \cdot \mathbf{D}_C.$$

Block-lifting structure modified quaternion $\widetilde{Q}$ for target operator $\mathbf{M}^{-}\left(\overline{Q}\right)$ (rule 4, Table I), will be defined $\mathbf{M}^{-}\left(\overline{Q}\right) =$ $= \mathbf{D}_C \cdot \mathbf{M}^{+}\left(\overline{Q}\right)^{T} \cdot \mathbf{D}_C$ and taking into account $\mathbf{M}^{+}\left(\overline{Q}\right) =$ $= \mathbf{M}^{+}(Q)^{T}$:

$$\mathbf{M}^{-}\left(\overline{Q}\right) = \mathbf{D}_C \cdot \mathbf{M}^{+}(Q)^{T} \cdot \mathbf{D}_C = \mathbf{D}_C \cdot \mathbf{P}_{post} \times$$
$$\times \begin{bmatrix} \mathbf{I}_2 & \mathbf{F}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(Q) & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{H}(Q) \\ 0 & \mathbf{I}_2 \end{bmatrix} \cdot \mathbf{P}_{pre} \cdot \mathbf{D}_C.$$

TABLE I
MODIFICATION RULES FOR $\mathbf{M}^{\pm}(Q)$

Rule	Target op.	Modification rule for $\widetilde{Q}$
1	$\mathbf{M}^{+}(Q)$	$\widetilde{\mathbf{P}}_{pre} = \mathbf{P}_{pre}$; $\widetilde{\mathbf{P}}_{post} = \mathbf{P}_{post}$; $\widetilde{\mathbf{F}} = \mathbf{F}$; $\widetilde{\mathbf{G}} = \mathbf{G}$; $\widetilde{\mathbf{H}} = \mathbf{H}$;
2	$\mathbf{M}^{+}\left(\overline{Q}\right)$	$\widetilde{\mathbf{P}}_{pre} = \mathbf{P}_{post}^{T}$; $\widetilde{\mathbf{P}}_{post} = \mathbf{P}_{pre}^{T}$; $\widetilde{\mathbf{F}} = -\mathbf{F}$; $\widetilde{\mathbf{G}} = -\mathbf{G}$; $\widetilde{\mathbf{H}} = -\mathbf{H}$;
3	$\mathbf{M}^{-}(Q)$	$\widetilde{\mathbf{P}}_{pre} = (\mathbf{P}_{post}^{T}) \cdot \mathbf{D}_c$; $\widetilde{\mathbf{P}}_{post} = \mathbf{D}_c \cdot (\mathbf{P}_{pre}^{T})$; $\widetilde{\mathbf{F}} = -\mathbf{H}$; $\widetilde{\mathbf{G}} = -\mathbf{G}$; $\widetilde{\mathbf{H}} = -\mathbf{F}$;
4	$\mathbf{M}^{-}\left(\overline{Q}\right)$	$\widetilde{\mathbf{P}}_{pre} = \mathbf{P}_{pre} \cdot \mathbf{D}_c$; $\widetilde{\mathbf{P}}_{post} = \mathbf{D}_c \cdot \mathbf{P}_{post}$; $\widetilde{\mathbf{F}} = \mathbf{F}$; $\widetilde{\mathbf{G}} = \mathbf{G}$; $\widetilde{\mathbf{H}} = \mathbf{H}$;

B. Pipeline structure of the integer Q-MUL multiplier

Distributed arithmetic [13] has been proposed as an efficient method for computing vector inner products. The process of obtaining the result of multiplying the input quaternion $X = [x1, x2, x3, x4]^{T}$ by the constant-quaternion Q is divided into five stages: pre $\mathbf{P}_{pre}$ and post $\mathbf{P}_{post}$ processing for multiplying by a modified quaternion $\widetilde{Q}$ and three stages obtaining the inner products $\mathbf{r} = [r_1, r_2]^{T}$ corresponding expansions: $\mathbf{V}(\widetilde{Q})$, $\mathbf{L}(\widetilde{Q})$, $\mathbf{U}(\widetilde{Q})$ on the adder-based distributed arithmetic ($DA_{\sum}$) [13]. The efficient procedure for computing inner products $\mathbf{r}$ between fixed coefficients of the block-lifting step (the matrices $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$) and variable vectors data X can be formulated in terms of the $DA_{\sum}$ the following way (for $\mathbf{H}$ stage, see Fig. 1):

$$r_1 = \begin{bmatrix} 1 \dots 1 \end{bmatrix} \times$$
$$\times \begin{bmatrix} -h_{11}^{\{0\}} \cdot x_3 & -h_{12}^{\{0\}} \cdot x_4 \\ 2^{-1} \cdot h_{11}^{\{1\}} \cdot x_3 & 2^{-1} \cdot h_{12}^{\{1\}} \cdot x_4 \\ \vdots & \vdots \\ 2^{-L} \cdot h_{11}^{\{L\}} \cdot x_3 & 2^{-L} \cdot h_{12}^{\{L\}} \cdot x_4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (13)$$

where $h_{ij}^{\{t\}} \in \{0, 1\}$ are binary coefficients elements of the stage matrix $\mathbf{H}$ in 2's-complement code, ij – element index, t – bit position, B is word length, $L = B-1$ is less significant bit position, $h_{ij}^{\{0\}}$ – sign bit. The equation (13) shows the organization of the inner products $\mathbf{r}$ calculation based on the two serial-connected processor elements (PE1), (PE2) and the accumulator (ACC) (see Fig. 2). The calculation of inner product is carried out by summing all elements of the matrix: the summation of the rows element is carried out by (PE1) and (PE2), the resulting elements of column are formed in the (ACC). At the same time, at the beginning of the calculation cycle, variables x_1 and x_2 are written to the accumulators, respectively. Hence, computation sum of $h_{ij}^{\{0\}}$ requires taking into account the control sign signal T_s (see Fig. 2). The latency of one block-lifting stage ($\mathbf{V}(\widetilde{Q})$) is $\approxeq 3Bt_{\sum}$, where $t_{\sum}$ is time of adder. Thus, the performance of pipeline Q_{MUL} is $\approxeq (3Bt_{\sum})^{-1}$ quaternion multiplication per second.

The same expression can be obtained for r_2 using second row of matrix $\mathbf{H}$ (h_{21} and h_{22}). For stages $\mathbf{V}(\widetilde{Q})$, $\mathbf{L}(\widetilde{Q})$, $\mathbf{U}(\widetilde{Q})$ result formation take B clock cycles, permutation matrices $\mathbf{P}_{pre}$ and $\mathbf{P}_{post}$ do not require additional logic elements, a sign change can be taken into account at subsequent stages.

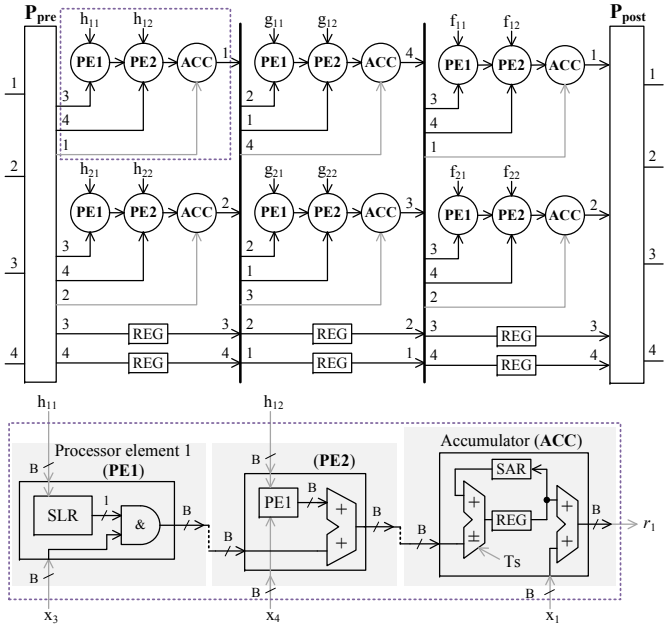
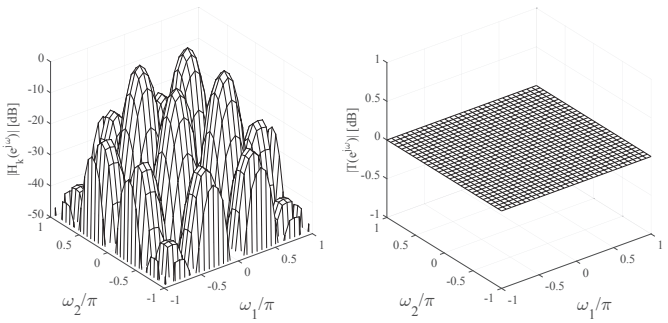


Fig. 2. Pipeline structure of Q-MUL (SAR – shift arithmetic right, SLR – shift logical left, B – word length, T_s – signal of sign bit)



a b

Fig. 3. 8-channel 2D NSQ-PUFB: (a) – magnitude response of the 3rd channel, (b) – total magnitude response of analysis-synthesis system

V. Design examples

Design problem of a two dimensional non-separable Int-Q-PUFB can be defined as: find a set of quaternions P_i and Q_i for a 1D Q-PUFB and word length B of block-lifting coefficients $\mathbf{F}(q)$, $\mathbf{G}(q)$, and $\mathbf{H}(q)$, which provide high value of the coding gain (CG_{MD}) of 2D NSB Q-PUFB with the following constraints: the maximum stopband attenuation (SBE) measured on terms of energy, as a result of quantization of block-lifting coefficients. The coding gain CG_{MD} of 2D NSB Q-PUFB and the transition to new iteration of the synthesis algorithm is determined according to the following procedure: 1) white noise $n \times n$ is generated; 2) an input image $\mathbf{x}_{n,n}$ is formed on the basis of the $AR(1)$ model applied to rows and columns consistently; 3) the input image is processed by synthesized 2D NSB Q-PUFB; 4) coding gain CG_{MD} is calculated. We design critically sampled 4-channel (12 taps) and 8-channels (24 taps) 2-D non-separable PMI LP Q-PUFB for the factorisation order $N = 2$. The coding gains CG_{MD} of 2-D non-separable PMI LP Q-PUFB for the isotropic autocorrelation function with the correlation factor $\rho = 0.95$ are the following: $CG_{MD} = 13.4\,\mathrm{dB}$ for "16in-16out" structure (proposed structure "16in-16out" in comparison with 2D PUFB in [6] $CG_{MD} = 11.55\,\mathrm{dB}$ is almost two decibels more); structure "64in-64out": $CG_{MD} = 17.15\,\mathrm{dB}$, amplitude response of the third analysis filter is shown in fig. 3a, SBE $\leq -20\,\mathrm{dB}$, direct current attenuation $DC_{att} \approx -300\,\mathrm{dB}$.

The total magnitude response of the analysis-synthesis system based on the 2D NSB PMI LP Q-PUFB (structure "64in-64out") is plane at $0\,\mathrm{dB}$ (see Fig. 3b). Thus, the given 2-D non-separable filter bank is a perfect reconstruction system. The dependence of the coding gain on the word length is following: $B = 8$ bits $CG_{MD} = 17.11\,\mathrm{dB}$; $B = 12$ bits $CG_{MD} = 17.15\,\mathrm{dB}$; $B = 16$ bits $CG_{MD} = 17.15\,\mathrm{dB}$.

Compares the energy compaction level between the 2D separable Q-PUFB based on the 1D Q-PUFB (8×24 Q-PUFB $CG_{1D} = 9.38\,\mathrm{dB}$) and 2D non-separable Q-PUFB (8×24 2D-NSQ-PUFB, $CG_{MD} = 17.15\,\mathrm{dB}$) proposed in this article, for the Barbara image, shows that the 2D non-separable Q-PUFB consistently generates a higher percentage of small-value coefficients, hence creates a significant increase in the number of zero trees.

VI. Conclusion

We devised the 2-D technique of non-separable factorization for 4-th and 8-th chanel PMI LP Q-PUFBs based on the integer-to-integer invertible quaternionic multipliers. The proposed multiplier is versatile, which allows using only $\mathbf{M}^{+}(Q)$ left multiplication matrix. The 2-D-NSQ-PUFB based on the given Q_{MUL} is a perfect reconstruction filter bank for finite precision, compared to know solutions it have less implementation complexity, higher coding gain and stopband attenuation. The factorization structures of 2-D NSQ-PUFB can be easily mapped on the parallel-pipeline architecture of quternionic multipliers. It should be noted that the latency of parallel-pipeline processing does not depend on the size of the original image in contrast to the conventional 2-D transform.

This work was supported by Belarusian Republican Foundation for Fundamental Research (project no. F18MV-016).

References

[1] E. Bayro-Corrochano, "The theory and use of the quaternion wavelet transform," *J. Math. Imaging Vis.*, vol. 24, pp. 19–35, 2006.

[2] W.L. Chan, H. Choi, and R.G. Baraniuk, "Coherent multiscale image processing using dual-tree quaternion wavelets," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1069–1082, July 2008.

[3] M. Kadiri, M. Djebbouri, and P. Carré, "Magnitude-phase of the dual-tree quaternionic wavelet transform for multispectral satellite image denoising," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, aug 2014.

[4] M. Parfieniuk and A. Petrovsky, "Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers," *Signal Process.*, vol. 90, pp. 1755–1767, 2010.

[5] N. A. Petrovsky, A. V. Stankevich, and A. A. Petrovsky, "Pipelined block-lifting-based embedded processor for multiplying quaternions using distributed arithmetic," in *5th Mediterranean Conference on Embedded Computing (MECO)*, 2016, pp. 222–225.

[6] S. Muramatsu, A. Yamada, and H. Kiya, "A design method of multidimensional linear-phase paraunitary filter banks with a lattice structure," *IEEE Trans. Signal Process.*, vol. 47, no. 3, pp. 690 – 700, 1999.

[7] T. Yoshida, S. Kyochi, and M. Ikehara, "A simplified lattice structure of two-dimensional generalized lapped orthogonal transform (2-D GenLOT) for image coding," in *17th IEEE International Conference on Image Processing (ICIP)*, Sept 2010, pp. 349–352.

[8] T. Suzuki and H. Kudo, "2D non-separable block-lifting structure and its application to M-channel perfect reconstruction filter banks for lossy-to-lossless image coding," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4943 – 4951, Aug. 2015.

[9] S. Oraintara, T. D. Tran, P. N. Heller, and T. Q. Nguyen, "Lattice structure for regular paraunitary linear-phase filterbanks and m-band orthogonal symmetric wavelets," *IEEE Trans. Signal Process.*, vol. 49, no. 11, pp. 2659–2672, Nov. 2001.

[10] N. Petrovsky, A. Stankevich, and A. Petrovsky, "Low read-only memory distributed arithmetic implementation of quaternion multiplier using split matrix approach," *Electronics Letters*, vol. 50, no. 24, pp. 1809–1811, Nov. 2014.

[11] M. Parfieniuk and S. Y. Park, "Sparse-iteration 4D CORDIC algorithms for multiplying quaternions," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2859–2871, Sept 2016.

[12] M. Parfieniuk and A. Petrovsky, "Quaternion multiplier inspired by the lifting implementation of plane rotations," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 57, no. 10, pp. 2708–2717, Oct. 2010.

[13] T. S. Chang, C. Chen, and C. W. Jen, "New distributed arithmetic algorithm and its application to IDCT," *IEE Proceedings - Circuits, Devices and Systems*, vol. 146, no. 4, pp. 159–163, Aug 1999.