EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Approximate calculation of 8-point DCT for various scenarios of practical applications

Dariusz Puchala

Correspondence:
dariusz.puchala@p.lodz.pl
Institute of Information Technology, Lodz University of Technology, Lodz, 215 Wolczanska Str., Lodz, Poland

Abstract

In this paper, based on the parametric model of the matrix of discrete cosine transform (DCT), and using an exhaustive search of the parameters' space, we seek for the best approximations of 8-point DCT at the given computational complexities by taking into account three different scenarios of practical usage. The possible parameter values are selected in such a way that the resulting transforms are only multiplierless approximations, i.e., only additions and bit-shift operations are required. The considered usage scenarios include such cases where approximation of DCT is used: (i) at the data compression stage, (ii) at the decompression stage, and (iii) both at the compression and decompression stages. The obtained results in effectiveness of generated approximations are compared with results of popular known approximations of 8-point DCT of the same class (i.e., multiplierless approximations). In addition, we perform a series of experiments in lossy compression of natural images using popular JPEG standard. The obtained results are presented and discussed. It should be noted, that in the overwhelming number of cases the generated approximations are better than the known ones, e.g., in asymmetric scenarios even by more than 3 dB starting from entropy of 2 bits per pixel. In the last part of the paper, we investigate the possibility of hardware implementation of generated approximations in Field-Programmable Gate Array (FPGA) circuits. The results in the form of resource and energy consumption are presented and commented. The experiment outcomes confirm the assumption that the considered class of transformations is characterized by low resource utilization.

Keywords: Discrete cosine transform, Approximate discrete cosine transform, Lossy data compression

1 Introduction

The need to transmit and archive multimedia data is pervasive. The constantly growing capabilities of modern image acquisition devices impose high demands on present-day communication and data storage systems. The solution to this problem are lossy compression standards which allow for substantial reduction of multimedia data sizes with reasonable loss of image quality. The widely used standards for lossy compression of static images and video sequences take advantage of block quantization where input data is first decorrelated in the domain of linear transform, and then subjected to scalar quantization. Here a commonly used linear transform is the discrete cosine transform (DCT)

which has good properties in terms of compression of highly correlated data. However, DCT requires floating-point arithmetic involving multiplications what can be prohibitive for battery operated embedded systems and low-cost hardware realizations. Hence, in recent years we can observe high interest of the scientific community in searching for novel computationally effective approximations of DCT (e.g., see [1]-[17]). Since modern image compression standards partition input images into 8×8-pixel blocks, and assume the separability of two-dimensional discrete cosine transform, only 8-point DCT has to be taken into consideration in the process of searching for effective approximations.

In this paper, we propose a family of highly effective approximations of 8-point DCT that can be used in three different scenarios of practical applications. In order to find the required approximations, we use the parametric model of the matrix of 8-point DCT, we browse the space of its parameters with an exhaustive search, and at the same time, we evaluate the quality of found approximations with the use of the proposed transform quality measures. The best approximations found are ordered by non-decreasing computational complexities and presented in a form of a specific dictionary of transform approximations. It should be noted that the proposed approximations of 8-point DCT require only additions and bit-shift operations involving integer representation of input and resulting data. It facilitates their implementation in embedded systems based on simple central processing units that do not support floating-point operations, and also imposes significantly smaller requirements on dedicated hardware implementations. The considered scenarios of practical usage include such cases where an approximation of DCT is used: (i) only at the data compression stage, (ii) only at the decompression stage, and (iii) at both the compression and decompression stages. Taking into account three scenarios is an additional contribution of this paper as other authors have not considered scenarios (i) and (ii).

The organization of the paper is as follows. In Section 2, we describe the theoretical background regarding modern image compression standards based on block quantization. The well-known approximations of 8-point DCT are listed and described in Section 3. In Section 4, we introduce the parametric model of 8-point DCT, define the considered scenarios of practical usage, and formulate the quality measures used in the process of searching for the best approximations of 8-point DCT. In the same section we present the best approximations found. The results of practical experiments involving natural images and verifying the effectiveness of the proposed approximations can be found in Section 5. In Section 6, we address the aspects of hardware realizations.

2 Theoretical background

The block quantization scheme (see Fig. 1) in its two-dimensional variant is a core of modern standards for lossy compression of static images and video sequences [18]. The main purpose of block quantization is to decorrelate input data in order to apply simple scalar quantizers instead of more computationally demanding vector quantization. The process of data decorrelation can be implemented with aid of linear transformation (matrix V in Fig. 1a). The linear transformation that allows for perfect decorrelation is a Karhunen-Loève transform (KLT) [19]. The elements of vectors in KLT domain are decorrelated, and also according to the central limit theorem, their values are normally distributed. The lack of correlation and normal distribution of random variables guarantee their statistical independence, which, in turn, justifies the usage of scalar quantization to individual

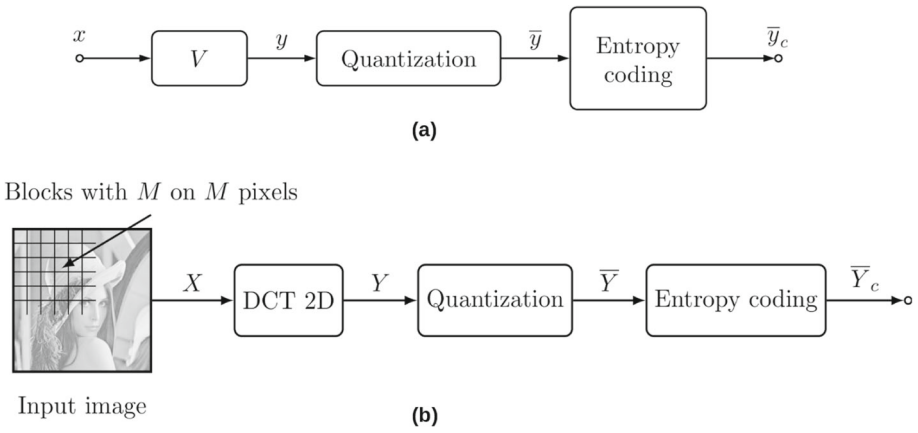

Fig. 1 Block quantization scheme: **a** One-dimensional data. **b** In two-dimensional case

elements of data vectors. Despite its optimality, KLT is not preferred in practical applications due to the following disadvantages: (i) lack of fast algorithms resulting in high computational complexity of order $\mathcal{O}(N^2)$, which is typical for matrix by vector multiplication; (ii) dependence on input data; and (iii) specific hat-like shape of the first base vector which causes undesirable visible artifacts at high compression levels.

In the light of this in practical applications, the fast approximation of KLT in the form of discrete cosine transform (DCT) is used most often. DCT has the following important features (see [19]): (i) it is asymptotically convergent to KLT for highly correlated data (i.e., for the first order Markov random processes with correlation $\rho \to 1$), and hence, it is independent of the input data; (ii) its first base vector is equivalued and allows to calculate an average of image pixels, which in case of two-dimensional lossy compression results in square-like but more pleasing for an eye artifacts; and (iii) it can be calculated with fast discrete cosine transform algorithm (FCT) of $\mathcal{O}(N \log_2 N)$ computational complexity (see [19]). The elements of matrix V of one-dimensional DCT in the case of N-element input vectors can be defined as follows:

$$V_{kn} = \alpha(k) \cos\left(\frac{\pi}{N} k \left(n + \frac{1}{2}\right)\right) \tag{1}$$

for $k, n = 0, 1, \ldots, N-1$, with $\alpha(k) = \sqrt{1/N}$ for $k = 0$ and $\alpha(k) = \sqrt{2/N}$ otherwise. The second stage of block quantization (see Fig. 1a) is scalar quantization based on rounding the values of vector elements previously divided by the quantization factors. The final stage is entropy coding which allows to reduce the data size further on by taking into account some statistical redundancy in data resulting from normal distribution of random variables and possible sequences of zero-valued coefficients obtained after quantization. For this purpose, in popular solutions (e.g., Joint Photographic Experts Group (JPEG) and Moving Picture Experts Group (MPEG) compression standards), the first order entropy coding in the form of Huffman codes, as well as the higher order entropy coding based on Run-Length Encoding (RLE), is used most often (c.f. [20, 21]).

The existing image and video compression standards assume separable models of natural images which means that also two-dimensional decorrelating transforms could be separable. The separability of transformation results in simpler computational scheme. In particular, if we take DCT then its two-dimensional variant can be calculated with use of

one-dimensional transform defined by (1) in a simple row-column approach (see [19]). In such case, it is enough to transform rows (columns) of an image and then columns (rows) of the obtained results of the first step. If X is an input image, then its representation Y in the domain of two-dimensional DCT can be calculated as $Y = VXV^T$. If we compare now both diagrams from Fig. 1a and b, the only difference lies in data dimensionality and preparation of input data. In two-dimensional case, an input image is divided into smaller blocks with M on M pixels (usually $M = 8$) which is a compromise between data decorrelation and computational complexity. In such case, the obtained reduction of a number of calculations can be estimated as $\mathcal{O}(\log_2 N / \log_2 M)$, where N is the size of a square image. Two-dimensional block quantization finds wide applications in popular standards for lossy compression of images and videos, namely, JPEG [20–22], MJPEG, MPEG-1,-2 [20, 21, 23], H.261 [20], H.263 [24], and H.264 [25].

Since two-dimensional DCT can be computed using one-dimensional transform, then the optimization process aiming at the reduction of computational complexity can be focused only on one-dimensional case. Moreover, if we assume image partition into blocks of size 8 by 8 pixels, then the optimization applies only to the computational procedure of 8-point DCT. Direct calculation of 8-point DCT, i.e., based on its definition as $y = Vx$, where V is defined as in (1), requires 64 multiplications and 56 additions. The fast and exact algorithm proposed by Loeffler et al. [1] reduces that number to 11 multiplications and 29 additions. It was shown by Duhamel and H'Mida that a theoretical lower bound on the number of multiplications of 8-point DCT equals precisely 11 (c.f. [2, 3]). However, in case of block quantization, some of multiplications required by DCT can be combined with multiplications needed to implement scalar quantization. Then the practical number of multiplications required by 8-point DCT can be reduced further on down to 5 multiplications as in Arai, Agui, and Nakajima's variant of fast DCT presented in paper [4]. Although the number of arithmetic operations, in particular multiplications, can be significantly reduced when compared to the definition approach, it does not change the fact that the remaining multiplications operate on rational numbers. Hardware realizations of rational numbers arithmetic can be prohibitive for low-power consumption and battery operated systems. One of possible solutions to this are multiplierless approximations of 8-point DCT that require only additions and bit-shift operations. The simplicity of such operations makes it possible to carry out calculations using integer arithmetic (i.e., natural binary code or two's complement representation). In turn integer arithmetic translates into significantly simpler hardware implementations and lower power consumption of the synthesized circuits. The mentioned features of the class of approximate DCTs explain high interest of the scientific community in developing new approximations closer to DCT than the known ones, but retaining the same small computational complexity. For example, we can indicate approximations with a very low number of additions, i.e., smaller or equal to 14, see papers [5, 12, 16], or approximations that except additions require bit-shift operations but both in a moderate number (c.f. [7, 8, 13]), and also such approximations where the number of additions is higher or equal to 24, see [6, 10, 15].

3 Popular approximations of 8-point DCT

In this section, we present a brief overview of popular approximations of 8-point DCT studied in the remaining part of the paper. The considered approximations are listed and

described in a chronological order. In Table 1, we can find the short summary of the transforms including abbreviated names of approximations, names of the authors and references to the papers where such approximations were formulated, numbers of additions and bit-shift operations, and whether the proposed transform has the orthogonality property.

3.1 SDCT2001 approximation

This is historically one of the first known approximations of 8-point discrete cosine transform. It was proposed in the year 2001 by T. I. Haweel in his research paper [6]. The apt idea standing behind the form of the matrix of this approximation can be briefly described as apply a signum function to the matrix of 8-point DCT. If a value of DCT matrix element is positive, then the resulting value would be 1. In turn, if this value is negative, then the resulting value would be -1. Because the base vectors of the resulting matrix are square this transform is called a Square "wave" DCT (SDCT). Its direct computation requires 56 additions. However, due to the symmetry of the row vectors of cosine transform, the matrix of SDCT can be factorized into a product of matrices requiring only 24 additions. It should be also noted that this approximation is not orthogonal. But as it was shown in paper [6], its inverse can be calculated only with additions and bit-shift operations. The SDCT is very often chosen as a starting transform to develop novel computationally effective approximations of 8-point DCT.

3.2 BAS2008I approximation

This is the first of several approximations formulated by Bouguezel et al. (see [7–10]). This transform was proposed in [7]. It is constructed on the basis of the well-known SDCT approximation by introducing into its matrix zero and $1/2$ valued elements. It can be verified that the resulting matrix is orthogonal which simplifies the construction of its inverse (as the transposed matrix of forward transform). Moreover, as it was shown in paper [7], the matrix of this approximation can be factorized into a product of three sparse matrices requiring: 8 additions, 6 additions, 4 additions, and 2 bit-shift operations, respectively. Hence, it is possible to calculate this transform with the total number of 18 additions and 2 bit-shift operations. On the basis of the property of orthogonality, it is obvious that an

Table 1 The list of popular and known from the literature approximations of 8-point DCT

Short name	Introduced by	No. of additions	No. of shifts	Orthogonality
SDCT2001	Haweel in [6]	24	0	-
BAS2008I	Bouguezel et al. in [7]	18	2	+
BAS2008II	Bouguezel et al. in [8]	21	0 (3)*	-
BAS2010	Bouguezel et al. in [10]	24	4	+
BAS2011 (a={0, $\frac{1}{2}$, 1})	Bouguezel et al. in [9]	16/18/18	0/2/0	+
CB2011	Cintra and Bayer [11]	22	0	-
BC2012	Bayer and Cintra [12]	14	0	+
PMCBR2012	Potluri et al. in [15]	24	6	+
PS2012	Puchala and Stokfiszewski in [13]	18	2	+
DR2014	Dhandapani and Ramachandran in [5]	12	0	-
PMCBKE2014	Potluri et al. in [16]	14	0	+

*The value in brackets indicate the number of operations required in the case of an inverse transformation. It should be noted that approximation BAS2011 is parametrized with the value of one parameter a. In this paper, we consider three values $a=\{0, \frac{1}{2}, 1\}$ (also considered in the original paper [9]) which results in three sets of the numbers of additions and bit-shift operations

inverse transformation can be calculated with precisely the same number of arithmetic operations.

3.3 BAS2008II approximation

In paper [8], the authors of BAS2008I transform propose another approximation of DCT taking as a starting point the well-known SDCT approximation. The proposed transform was obtained by appropriately setting to zero some of the entries of SDCT matrix which resulted in the novel approximation that can be calculated as a product of four sparse matrices requiring 8, 6, 6, and 1 additions, respectively. It gives a total number of 21 additions. Although the transform itself is not orthogonal, its inverse matrix can be determined with the same number of additions and three additional bit-shift operations.

3.4 BAS2010 approximation

The following approximation is a multiplication-free transform of any size $N > 4$ (see [10]). For the case of $N = 8$, it can be effectively calculated with the aid of radix-2 like structures, and it requires 24 additions and 4 bit-shift operations. Moreover its matrix is orthogonal which allows to find the inverse matrix immediately by the transposition.

3.5 BAS2011 approximation

This transform (proposed in [9]) is the only one in the group of considered well-known approximations which is parametrized (excluding the parametric models used in papers [15, 16] in order to find appropriate approximations). It was constructed on the basis of approximation presented in paper [7] (BAS2008I) by introducing an arbitrary parameter into the matrix of that transformation and by performing some permutations of the rows of the matrix. As a result a parametric transform was obtained defined by the following matrix parametrized with one parameter a, i.e.:

$$U_a = D_a \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \end{bmatrix},$$

with $D_a = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2\sqrt{1+a^2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{2\sqrt{1+a^2}}\right)$. Hence, this transform can be treated as a parametric variant of approximation BAS2008I. It should be noted that U_a matrix is orthogonal regardless the value of parameter a. This feature greatly simplifies calculation of inverse transformation. Of course, the value of parameter a should be selected in such a way that multiplication operations are not required. In paper [9] the following values were considered, i.e., $a \in \{0, \frac{1}{2}, 1\}$. Moreover a direct calculation of U_a transform may require 36 additions; however, U_a matrix can be factorized into a product of three sparse matrices requiring respectively: 8 additions, 6 additions, and 2 additions for $a = 0$ or 4 additions with $a \neq 0$, and a number of 2 bit-shift operations with $a \neq 0$ and $a \neq 1$.

3.6 CB2011 approximation

On the basis of the matrix of 8-point discrete cosine transform, a novel approximation obtained by means of rounding-off operation was formulated and proposed by Cintra and Bayer in paper [11]. A rounding-off operation, when applied to the scaled by 2 matrix of DCT, allows to obtain, in contrast to the signum function, the approximation containing zeros. As it was shown in paper [11], the matrix of the obtained transform can be further on decomposed into a product of three sparse matrices with the following numbers of additions: 8, 12, and 2. It gives the total number of 22 additions. It should be noted that the matrix of the considered approximation is not orthogonal, which means that an inverse transformation cannot be easily calculated as a transposition of the matrix of forward transform.

3.7 BC2012 approximation

In paper [12], Bayer and Cintra propose a novel approximation of 8-point DCT that requires only 14 additions. It is based on previous approximation formulated in [11] and is constructed by replacing aptly selected elements of CB2011 matrix with zeros. The elements of the resulting matrix are only $\{-1, 0, 1\}$, and hence, no bit-shift operations are required. Moreover transform matrix can be factorized into a product of three sparse matrices requiring 8, 4 and 2 additions, respectively. It can be also easily verified that the obtained transformation is orthogonal.

3.8 PMCBR2012 approximation

This transform, proposed by Potluri et al. in paper [15], was obtained in the optimization process aimed at minimizing the differences between transfer functions of selected row vectors of the DCT transformation and the searched for approximation, by assuming: (i) parametric model of transform matrix, (ii) elements of transform matrix can only take values $\{0, \pm 1, \pm 2\}$, and (iii) the resulting transform must be orthogonal. The exhaustive search procedure allowed to find the approximation characterized by low computational complexity that requires only additions and bit-shift operations. It was shown in paper [15] that transform matrix can be factorized into a product of three sparse matrices requiring: 8 additions, 12 additions and 4 bit-shifts, 4 additions and 2 bit-shifts. It gives the total number of 24 additions and 6 bit-shifts.

3.9 PS2012 approximation

The considered approximation of 8-point DCT was proposed by Puchala and Stokfiszewski in [13]. It is based on the approach presented previously in [14] which in order to approximate 8-point DCT uses two 4-point DCTs operating on the outputs of four 2-point Haar transforms that take as inputs adjacent pairs of elements of input vectors. It means that one 4-point DCT operates on a low-frequency band component of input data (an averaged input data), while the second one takes as input high frequency band, i.e., details of input signal. In approximate transform from paper [13], the first 4-point DCT was calculated in an approximate way using only additions and bit-shift operations while the second one was removed. It allowed to obtain multiplication-free approximation which is orthogonal. The matrix of this transform can be factorized into a product of four matrices which require: 8 additions, 4 additions, 4 additions, and 2 additions and 2 bit-shifts respectively.

Hence, the total number of arithmetical operations equals: 18 additions and 2 bit-shift operations.

3.10 DR2014 approximation

This is a very low computational complexity approximation of DCT. It was proposed in paper [5]. Since it requires only 12 additions and no bit-shift operations, it can be characterized by the smallest number of arithmetical operations among DCT approximations considered in this paper. The proposed approximation is not orthogonal but the authors formulate an inverse matrix which can be characterized by the same computational effectiveness. Moreover the aspects of hardware implementation of the proposed transform are also discussed in paper [5]. For example a pipelined implementation of this transform in FPGA circuit (Xilinx Vertex 7 series) requires 132 LUT tables and 134 logic elements and the delay of data propagation equaled 3.247 ns. This can be viewed as a very small demand for hardware resources.

3.11 PMCBKE2014 approximation

This transform is an another example of the approximation obtained in the process of exhaustive search using a parametric model for transform matrix. It was proposed in paper [16] by Potluri et al. Also in this case the optimization criterion was to minimize the arithmetic complexity with similar constraints as in [15], i.e., (i) a specific parametric model of transform matrix was imposed, (ii) matrix elements can take values from the set $\{0, \pm 1, \pm 2\}$, and (iii) transform matrix must be orthogonal. The obtained approximation of 8-point DCT can be decomposed into a product of three sparse matrices where each of them requires respectively: 4, 2. and 8 additions. It gives the total number of 14 additions required to calculate this transformation.

4 Methods

In this section, we describe the parametric model used to search for approximations of 8-point DCT and formulate the specific requirements that must be met by the model in order to obtain invertible and orthogonal matrices. Next, by taking into account three different scenarios of practical usage, we introduce measures of transform qualities, called quality indexes, which include error components resulting from approximation of 8-point DCT and from quantization of transform coefficients (i.e., energy compaction capability). Then, based on the formulated quality indexes and with use of the proposed procedure, we search for approximated variants of 8-point DCT operating on model input signals. The obtained results are listed and arranged in order of increasing computational complexities.

4.1 Considered parametric model

In order to make possible the search for approximations of 8-point DCT in our research, we take advantage of the model formulated previously in paper [16]. The model is constructed on the basis of the matrix of 8-point DCT (see formula (1)) by replacing its identical elements with symbolic parameters. Since the matrix of 8-point DCT has seven individual values, then seven unique parameters, i.e., $\{a, b, c, d, e, f, g\}$, would be required by the model. The obtained model can thus be described in the matrix form as:

$$U{=}D\begin{bmatrix} a & a & a & a & a & a & a & a \\ g & f & e & d & -d & -e & -f & -g \\ b & c & -c & -b & -b & -c & c & b \\ f & -d & -g & -e & e & g & d & -f \\ a & -a & -a & a & a & -a & -a & a \\ e & -g & d & f & -f & -d & g & -e \\ c & -b & b & -c & -c & b & -b & c \\ d & -e & f & -g & g & -f & e & -d \end{bmatrix}, \tag{2}$$

where D is a diagonal scaling matrix required to guarantee the unit length of row vectors. Its main diagonal contains the elements: $\{D_{0,0}, D_{4,4}\} = 1/(2\sqrt{2}\,|a|)$, $\{D_{1,1}, D_{3,3}, D_{5,5}, D_{7,7}\} = 1/\sqrt{2(d^2+e^2+f^2+g^2)}$, and further on in the case of remaining two vectors we have: $\{D_{2,2}, D_{6,6}\} = 1/(2\sqrt{b^2+c^2})$. It can be easily verified that it is possible to factorize matrix U into the product of the following four matrices:

$$U = DP_1U_2U_3, \tag{3}$$

where matrix U_3 computes the sums and differences of elements of input vector, U_2 is a block diagonal matrix that contains the parameters of the model, i.e.:

$$U_2{=}\left[\begin{array}{c|c} A & O_4 \\ \hline O_4 & B \end{array}\right],\ U_3{=}\left[\begin{array}{c|c} I_4 & J_4 \\ \hline J_4 & -I_4 \end{array}\right]$$

with O_4 being a zero valued matrix with size 4 on 4 elements, I_4 standing for an identity matrix, and J_4 defined as follows:

$$I_4{=}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ J_4{=}\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

and the remaining matrix P_1 describes a permutation required to restore the proper order of row vectors of the resulting matrix U. This matrix takes form:

$$P_1{=}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The 4 on 4 element matrices A and B, which are the block diagonal elements of matrix U_2, hold the symbolic parameters of the model, i.e.:

$$A{=}\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix},\ B{=}\begin{bmatrix} d & e & f & g \\ -e & -g & -d & f \\ f & d & -g & e \\ -g & f & -e & d \end{bmatrix}.$$

Matrix A can be factorized further on with application of *divide-and-conquer* strategy into the product of four sparse matrices of the following form:

$$A = P_2 \left[\begin{array}{cc:cc} a & a & 0 & 0 \\ a & -a & 0 & 0 \\ \hdashline 0 & 0 & b & c \\ 0 & 0 & c & -b \end{array}\right] \left[\begin{array}{c:c} I_2 & I_2 \\ \hdashline I_2 & -I_2 \end{array}\right] P_3, \tag{4}$$

where I_2 is an identity matrix, P_2 and P_3 are permutation matrices defined as:

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

In case of factorization of 8-point DCT matrix B is a 4-point cosine transform of type four, and as such it could be factorized. However, such factorization (i) would not give a significant reduction in a number of arithmetic operations and (ii) would not guarantee the assumed form of matrix B for arbitrary values of free parameters, in particular for the integer powers of 2. Hence, the proposed and adopted in this paper final construction of factorization (3) can be described in graphic form as shown in Fig. 2.

An analysis of computational structure of the proposed factorization (see Fig. 2) allows to determine the numbers of additions and multiplications required by the matrix product described by formula (3). These numbers depend strictly on the values of parameters but in the worst scenario they can be upper bounded by the numbers of 28 additions and 22 multiplications. When compared with the direct matrix by vector multiplication, which requires 56 additions and 64 multiplications, the structure from Fig. 2 describes undeniably an approach with reduced computational complexity. However, it should be emphasized that we are interested in such values of parameters $\{a, b, c, d, e, f, g\}$ which do not require multiplications but much simpler bit-shift operations instead. Then if only the scaling operations described by the matrix D could be combined with the multiplications required at the quantization step, then the resulting structure would be multiplication free. In such case the only operations required to calculate the approximation of 8-point DCT would be additions (upper bounded by a number of 28 operations) and bit-shift operations (with a number less than or equal to 22). In the following part of the paper,

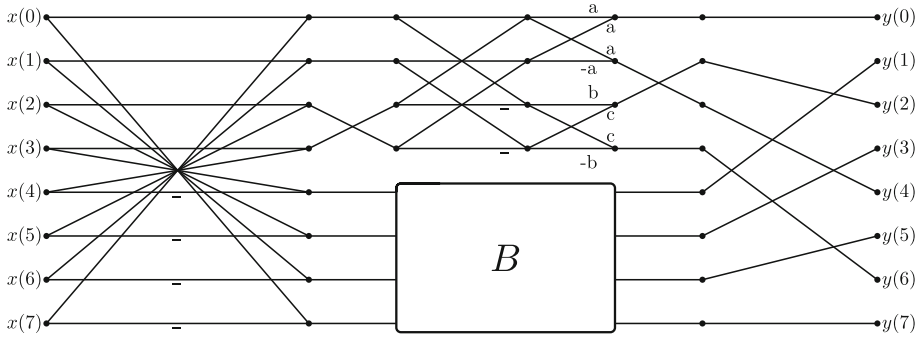

Fig. 2 The dataflow diagram describing factorization of the considered parametric transform model

we assume the acceptable values of parameters specified by the set: $\{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 0, 1, 2\}$. Multiplications by such values can be implemented simply as right bit-shift operations by a number of 3, 2, or 1 bit, or bit-shift by 1 bit left (except $\{0, 1\}$ values which do not require any additional actions). With such assumptions all operations required by matrix B, i.e., additions and bit-shifts, can be implemented according to the parallel computational scheme presented in Fig. 3. Here, all bit-shift operations can be calculated independently in a parallel manner, while additions can be realized in mixed parallel-sequential mode based on *reduction* or *sweep-up* approach. Then such a scheme, which requires at most 12 additions and 16 bit-shifts, can be characterized by high step efficiency. As we can see with such realization of the scheme from Fig. 3 , only a number of three sequential steps would be required. It can be exceptionally important in case of Graphics Processing Unit (GPU) based [17, 26, 27], i.e., massively parallel computations, or Single Instruction, Multiple Data (SIMD) [28, 29], and i.e. vector, parallelism, accelerated implementations. Similarly in the case of hardware implementations, it would translate into relatively short data flow paths.

4.1.1 Invertibility conditions

In order to avoid both trivial solutions or singularities within obtained results, we assume throughout this paper that the matrices U of transforms generated with the considered parametric model are invertible. It is obvious that invertibility of matrix U depends strictly on the values of model parameters. Matrix U will be invertible only when all of the components of factorization (3) are also invertible. It can be easily verified that $\det(U_3) \neq 0$ (to be precise $\det(U_3)=16$) and on the basis of definition of permutation matrix we have also $\det(P_1)=1$. The determinants of the remaining two matrices D and U_2 depend on the values of model parameters. Matrix U_2 will be invertible if only its component matrices A and B are invertible. Based on factorization (4) it can be seen that matrix A is invertible if ($a \neq 0$) and ($b \neq 0$ or $c \neq 0$). For the remaining matrices in formula (4) we have $\det(P_2)=\det(P_3)=1$ and the determinant of matrix (built with identity matrices I_2) calculating sums and differences of data elements equals 4. In case of matrix B it is not so straightforward to provide general conditions of invertibility. However, in the case of the considered, i.e., reduced, set of acceptable values of parameters it can be proved, even by exhaustive search, that matrix B will be invertible if at least one among its parameters is not equal to zero, i.e. $d \neq 0$ or $e \neq 0$ or $f \neq 0$ or $g \neq 0$. Then it can be easily verified that if the

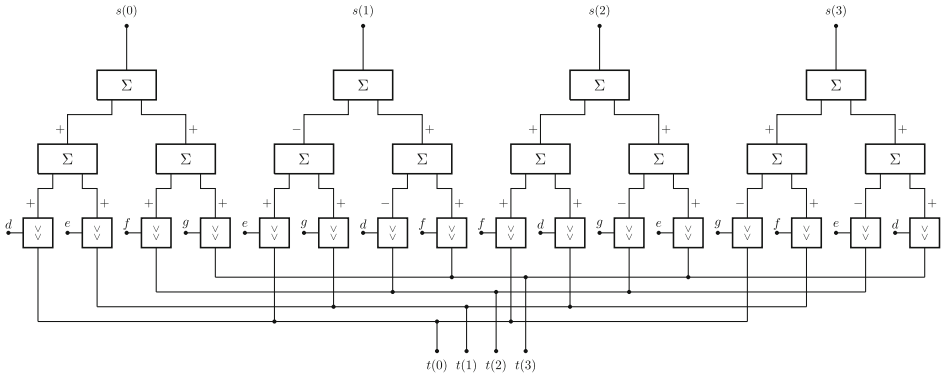

Fig. 3 Block diagram of step efficient and multiplication free calculation scheme of matrix B (i.e., $s = Bt$)

above conditions of invertibility of U_2 matrix are met, we also have $\det(D)\neq 0$, and as a consequence $\det(U)\neq 0$.

4.1.2 Orthogonality conditions

By orthogonality of matrix U, we understand the condition $UU^T=I$, where $(\cdot)^T$ stands for matrix transposition, and I is an identity matrix. It should be noted that the role of matrix D in factorization (3) is to normalize the length of base vectors of resulting transform. Hence, the orthogonality condition will be met if only the following product $(P_1U_2U_3)(P_1U_2U_3)^T$ results in a diagonal matrix. However, since the product $(U_3U_3^T)$ is diagonal and P_1 is a permutation matrix, then the formulated demand can be reduced to simpler form which assumes that the result of $(U_2U_2^T)$ is diagonal. Now on the basis of definition of matrix U_2 we can observe that the reduced demand will be fulfilled if only both products (AA^T) and (BB^T) give diagonal matrices. Next, it is easy to verify that the first product (AA^T) is always diagonal regardless of the values of parameters $\{a,b,c\}$. In the second case, we can formulate a general condition guaranteeing that (BB^T) is diagonal. It takes the following form:

$$f(g-d)-e(g+d)=0. \tag{5}$$

However, with the assumed set $\{\frac{1}{8},\frac{1}{4},\frac{1}{2},0,1,2\}$ of acceptable values of parameters, it can be shown that condition (5) will be satisfied if at least one of the parameters $\{d,e,f,g\}$ will be equal to zero (of course, excluding the case when all parameters are zeros). In this way the necessary conditions for invertibility and orthogonality of matrix U were formulated. In the further part of the section, we formulate the procedure used in the process of exhaustive browsing of the parameters' space.

4.1.3 Procedure for generating the requested approximations

According to the basic assumption of the paper the requested approximations of 8-point DCT are generated with the formulated parametric model in the process of exhaustive browsing of the parameters' space. The model takes 7 parameters whose set of acceptable values $\{\frac{1}{8},\frac{1}{4},\frac{1}{2},0,1,2\}$ consists of 6 elements. This gives the number $6^7=279936$ of all possible combinations of parameters' values. Hence, to make the analysis of results simpler, the list of generated approximations is sorted in order of non-decreasing number of additions, and within the same number of additions, in order of non-decreasing number of bit-shift operations (typically bit-shifts are less computationally demanding than additions). Moreover the approximations that have higher or equal value of the quality index (the smaller value the better) to approximations with lower computational complexity are excluded from the list. In the further part of the paper the appropriate quality indexes of approximations for each of the considered scenarios of practical usage are formulated (see Section 3.2). The approximations generating procedure can be described as follows:
Procedure. (For generating approximations of DCT)

1 Iterate through consecutive combinations of parameter values making steps 2 and 5.
2 If the values of parameters satisfy the invertibility condition, then generate the approximation U. Otherwise skip the steps 3 to 5.
3 If the values of parameters satisfy the conditions of orthogonality, then set an appropriate flag indicating that approximation U is orthogonal.

4 Based on the values of parameters calculate the precise numbers of additions and bit-shift operations required by the given approximation U (it can be done on the basis of factorization (3)).
5 Add the generated approximation U with all additional data to the resulting list.
6 Once the list of approximation is generated sort its element in order of non-decreasing number of additions, and within the same number of additions, sort them locally in order of non-decreasing number of bit-shift operations.
7 Exclude from the list such approximations that can be characterized by a higher or equal value of quality index than any element placed closer to the beginning of the sorted list.
8 Finish.

After executing the above procedure, we obtain the requested list of approximations of 8-point DCT.

4.2 Considered scenarios of data compression and decompression

In this paper, we consider three scenarios for compression and decompression of data. The first two scenarios are named asymmetric because two different transforms, i.e., approximation of 8-point DCT and 8-point DCT itself, are used interchangeably at compression and decompression stages. The last scenario uses approximate DCT at both stages (approximate transform $\mathcal{U}$ for compression and its inverse $\mathcal{U}^{-1}$ or transpose $\mathcal{U}^{T}$ for decompression of data). We also assume that separable Markov fields with high correlation are a good model of natural images and choose block quantization as a tool for compression and decompression of images. Then, it is obvious that two-dimensional transform used by block quantization would be also separable, which means that to its calculation only one-dimensional transform is required. In one-dimensional case, the equivalent model of data would be the first-order Markov process. Hence, in the further part of this section, we can consider only one-dimensional case to determine the quality indexes describing the effectiveness of found approximations.

4.2.1 Asymmetric scenario of type I

The asymmetric scenario of type I assumes to use approximation of DCT (described by matrix $\mathcal{U}$) for compression of data and DCT (described by matrix V) at the decompression stage (see Fig. 4 where we have a simplified diagram of block quantization). An example of practical application of such scenario might be systems in which the compression step must be implemented in a simplified and computationally effective way, e.g., by battery operated hardware. On the other hand, it is not economically justified to modify the data decompression algorithms built into the receiving devices, which operate using DCT.

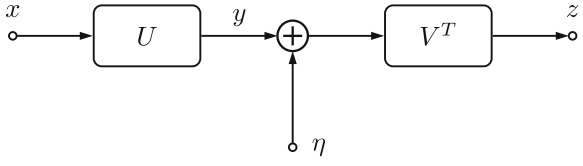

Fig. 4 Block diagram of asymmetric scenario of type I for compression and decompression of data

Then, if the column vector x denotes samples of input data (modeled as realizations of first-order Markov process), then the autocovariance matrix of the signal can be calculated as $R_x=\mathrm{E}(xx^T)$, where $\mathrm{E}(\cdot)$ denotes an expected value operator. We assume in this paper that R_x is an autocovariance matrix of the first-order Markov process with correlation coefficient $\rho=0.95$. In Fig. 4 column vector η represents the quantization noise, which is added to signal in U transform domain as a result of quantization, wherein $R_\eta=\mathrm{E}(\eta\eta^T)=\sigma_\eta^2 I$ with I being an identity matrix and σ_η^2 denoting a variance of quantization noise. In case of optimal bit-allocation applied to scalar quantizers (see [18, 19]) with a given bit budget $N\Theta$ bits, where N is a size of U transform and Θ describes an average number of bits per one random variable (element of y), the variance of quantization noise can be calculated as $\sigma_\eta^2=\kappa 2^{-2\Theta}\pi(U)$. It should be noted that κ is a constant depending on the probability distribution of elements of y vector (we assume normal distribution and $\kappa=5.33$) and $\pi(U)$ describes the product of variances of random variables in U transform domain. We have then:

$$\pi(U)=\left(\prod_{i=0}^{N-1}(R_y)_{ii}\right)^{\frac{1}{N}}, \tag{6}$$

where R_y is an autocovariance matrix in U transform domain, i.e., $R_y=\mathrm{E}(yy^T)$, with $y=Ux$, and by $(R_y)_{ii}$ for $i=0,1,\ldots,N-1$, we denote the diagonal elements of matrix R_y, i.e., the above mentioned variances of random variables in the domain of U transformation. In addition, it is assumed further on that there exist no correlation between the elements of vectors x and η, which results in $\mathrm{E}(x\eta^T)=\mathrm{E}(\eta x^T)=O$, where O is a zero matrix.

In the case of asymmetric scenario of type I, the output vector takes form $z=V^T(Ux+\eta)$ (c.f. Fig. 4). Let $\mathrm{tr}(\cdot)$ denotes a trace of matrix. Then the mean squared error (MSE) of data reconstruction is defined as:

$$\begin{aligned}
\epsilon_{MSE}^{(I)}(U) &= \mathrm{E}\left(\mathrm{tr}\left((z-x)(z-x)^T\right)\right)= \\
&= \mathrm{E}\left(\mathrm{tr}\left((V^TUx-x+V^T\eta)\right.\right. \\
&\quad \left.\left.(V^TUx-x+V^T\eta)^T\right)\right)= \\
&= \mathrm{E}\left(\mathrm{tr}\left((V^TU-I)x+V^T\eta)\right.\right. \\
&\quad \left.\left.((V^TU-I)x+V^T\eta)^T\right)\right).
\end{aligned}$$

Let $W=V^TU-I$, then we have:

$$\begin{aligned}
\epsilon_{MSE}^{(I)}(U) &= \mathrm{E}\left(\mathrm{tr}\left(Wx+V^T\eta)(Wx+V^T\eta)^T\right)\right)= \\
&= \mathrm{E}\left(\mathrm{tr}\left(Wxx^TW^T+Wx\eta^TV+\right.\right. \\
&\quad \left.\left.+V^T\eta x^TW^T+V^T\eta\eta^TV\right)\right)= \\
&= \mathrm{tr}\left(W\mathrm{E}\left(xx^T\right)W^T+W\mathrm{E}\left(x\eta^T\right)V^T+\right. \\
&\quad \left.+V^T\mathrm{E}\left(\eta x^T\right)W^T+V^T\mathrm{E}\left(\eta\eta^T\right)V\right).
\end{aligned}$$

Based on the following assumptions, that $\mathrm{E}(xx^T)=R_x$, $\mathrm{E}(\eta\eta^T)=R_\eta$, and $\mathrm{E}(x\eta^T)=\mathrm{E}(\eta x^T)=O$, we can write:

$$\epsilon_{MSE}^{(I)}(U) = \mathrm{tr}\left(WR_xW^T\right)+\mathrm{tr}\left(V^TR_\eta V\right),$$

which, after taking into account the definition of R_η matrix, and the fact that $VV^T=I$, results in formula:

$$\epsilon^{(I)}_{MSE}(U) \;=\; \mathrm{tr}\left(WR_xW^T\right)+N\kappa 2^{-2\Theta}\pi(U). \tag{7}$$

From formula (7) it can be observed that in this scenario the MSE is a sum of two components, i.e., the first one:

$$\epsilon^{(I)}_{A}(U) \;=\; \mathrm{tr}\left((V^TU-I)R_x(V^TU-I)^T\right),$$

which describes an error resulting from approximation of DCT by U transform (we have $\epsilon^{(I)}_{A}(V)$=0), and the second component, defined as:

$$\epsilon^{(I)}_{Q}(U,\Theta) \;=\; N\kappa 2^{-2\Theta}\pi(U),$$

which describes the quantization error resulting from assumed budget of $N\Theta$ bits and the coding properties of U transform, expressed by the product $\pi(U)$. Thus:

$$\epsilon^{(I)}_{MSE}(U) \;=\; \epsilon^{(I)}_{A}(U)+\epsilon^{(I)}_{Q}(U,\Theta). \tag{8}$$

The last effort is to formulate the quality index that will allow to quantify the efficiency of U transform by taking into account both components of MSE (c.f. formula (8)). However, the value of the second component, i.e., $\epsilon^{(I)}_{Q}(U,\Theta)$, depends on the average number of bits Θ assigned to each element of vector y. The proposed solution is to take into account the quantization error in the mean sense, i.e., calculated as the average value over a fixed set of Θ_i values, where we assume Θ_i=$\frac{1}{2}(i+1)$ bits for i=0, 1, ..., $L-1$, with L=12. Then the proposed index can be formulated as:

$$\chi_1(U)=\epsilon^{(I)}_{A}(U)+\frac{1}{L}\left(\sum_{i=0}^{L-1}\epsilon^{(I)}_{Q}(U,\Theta_i)\right). \tag{9}$$

It should be noted that the smaller the value obtained with formula (9), then the higher the efficiency of U transform.

With the formulated transform quality index it is possible to analyze known approximations and generate approximations based on the adopted parametric model. In the first place known approximations of 8-point DCT, described in Section 2, would be analyzed. The obtained results are collected in Table 2. In addition to the values of quality index $\chi_1(U)$, also the values of approximation error $\epsilon^{(I)}_{A}(U)$, as well as the values of the product $\pi(U)$ are presented in the same table. It should be noted that results are ordered relative to the non-decreasing number of additions required by the specific approximation, and in the case of identical number of additions, they are sorted by the non-decreasing number of bit-shift operations.

The analysis of results from Table 2 shows that the efficiency of approximate transforms increases with increasing computational complexity, which is fully expected. For the worst case, i.e., with transform DR2014 (only 12 additions), we have $\chi_1(U)\approx$15.15, while the best transform, i.e., PCMBR2012 (24 additions and 6 bit-shifts), can be described by $\chi_1(U)\approx$0.57. However, this is not a rule, since we can indicate transforms with lower computational complexity, which give results better than transforms with higher complexities, e.g., CB2011, which is better in the sense of $\chi_1(U)$ index than SDCT2001 and BAS2010 approximations. Especially noteworthy here is the CB2011 approximation, which can be characterized by high efficiency ($\chi_1(U)\approx$0.62) in relation to its computational complexity

Table 2 The results of analysis of known approximations of DCT in the case of the asymmetric scenario of type I ($\pi(V)$=0.131042)

Type	Adds	Bit-shifts	$\chi_1(U)$	$\epsilon_A^{(I)}(U)$	$\pi(U)$	Orthogonality
DR2014	12	0	15.155350	13.228013	0.542537	-
PMCBKE2014	14	0	1.288450	0.631898	0.184816	+
BC2012	14	0	1.131665	0.475113	0.184816	+
BAS2008III (a=0.0)	16	0	1.142898	0.568321	0.161740	+
BAS2008III (a=1.0)	18	0	1.142676	0.568201	0.161711	+
BAS2008III (a=0.5)	18	2	1.090498	0.542735	0.154192	+
PS2012	18	2	0.882784	0.347891	0.150570	+
BAS2008I	18	2	0.738024	0.190261	0.154192	+
BAS2008II	21	0	0.732042	0.152670	0.163090	-
CB2011	22	0	0.618240	0.078402	0.151962	+
SDCT2001	24	0	0.757160	0.165835	0.166455	-
BAS2010	24	4	0.690653	0.168235	0.147058	+
PMCBR2012	24	6	0.569858	0.049666	0.146431	+

(only 22 additions). It can be described by good approximation of DCT ($\epsilon_A^{(I)}(U) \approx 0.08$) and also by high effectiveness in quantization ($\pi(U) \approx 0.15$).

In the second place, the procedure described in Section 3 was used. Considering "pleasant for an eye" artifacts generated during image compression by the first base vector with identical values, we assume a constant value of parameter $a = 1$. The generated approximations are presented in Table 3 in the order of increasing computational complexity, and thus also increasing effectiveness understood in the sense of $\chi_1(U)$ index. It should be noted that approximations APRXI.1 and APRXI.7 correspond to known transforms, i.e., namely BC2012 and CB2011. In the case of approximations PMCBKE2014, BAS2008III, PS2012, BAS2008II, SDCT2001, BAS2010, and PMCBR2012, it was possible to found transforms with higher effectiveness, which can be also characterized by identical or smaller computational complexity. The best result ($\chi_1(U) \approx 0.48$) was obtained with a number of 28 additions and 10 bit-shift operations with APRXI.11 (double underlined row of the table). This approximation is very close to DCT (see $\epsilon_A^{(I)}(U)$=0.010708) and can be also characterized by high performance in block quantization (c.f. $\pi(U)$=0.132910 and $\pi(V) = 0.131042$). The featured approximation, i.e., the one with high ratio of effectiveness to computational complexity, is APRXI.8 (underlined row of Table 3). The mentioned transform reaches the $\chi_1(U) \approx 0.55$ value of quality index with 22 additions and 4 bit-shift operations.

4.2.2 Asymmetric scenario of type II

The block diagram of the asymmetric scenario of type II is presented in Fig. 5. Here, we also have two different transforms at the compression and decompression stages, however, the difference from the scenario of type I lies in the usage of DCT approximation at the decompression stage. Such scenario may find practical applications in the case of image preview devices with low computing power and designed for low energy consumption. This in turn implies the necessity of constructing simple algorithmic solutions.

Table 3 The approximations of DCT generated by the proposed procedure in the case of the asymmetric scenario of type I

Name	Adds	Bit-shifts	$\chi_1(U)$	$\epsilon_A^{(I)}(U)$	$\pi(U)$	{a, b, c, d, e, f, g}	Orthogonality
APRXI.1	14	0	1.131665	0.475113	0.184816	$\{1, 1, 0, 0, 0, 0, 1\}$	+
APRXI.2	16	0	1.131547	0.475113	0.184783	$\{1, 1, 1, 0, 0, 0, 1\}$	+
APRXI.3	16	2	1.071089	0.445176	0.176191	$\{1, 1, \frac{1}{2}, 0, 0, 0, 1\}$	+
APRXI.4	18	0	0.755224	0.166628	0.165687	$\{1, 1, 0, 0, 0, 1, 1\}$	-
APRXI.5	20	0	0.755118	0.166628	0.165657	$\{1, 1, 1, 0, 0, 1, 1\}$	-
APRXI.6	20	2	0.697819	0.136691	0.157954	$\{1, 2, 1, 0, 0, 1, 1\}$	-
APRXI.7	22	0	0.618240	0.078402	0.151962	$\{1, 1, 0, 0, 1, 1, 1\}$	+
APRXI.8	22	4	0.554463	0.052222	0.141378	$\{1, 1, 0, 0, \frac{1}{2}, 1, 1\}$	-
APRXI.9	24	2	0.554373	0.052222	0.141353	$\{1, 1, 1, 0, \frac{1}{2}, 1, 1\}$	-
APRXI.10	24	6	0.501088	0.022286	0.134780	$\{1, 1, \frac{1}{2}, 0, \frac{1}{2}, 1, 1\}$	-
APRXI.11	28	10	0.482868	0.010708	0.132910	$\{1, 1, \frac{1}{2}, \frac{1}{8}, \frac{1}{2}, 1, 1\}$	-

By noting that $z{=}U^T(Vx+\eta)$, and also using analogous mathematical derivation, we get the formula for MSE in the considered case, which takes the following form:

$$\epsilon_{MSE}^{(II)}(U){=}\epsilon_A^{(II)}(U)+\epsilon_Q^{(II)}(U,\Theta), \tag{10}$$

where $\epsilon_A^{(II)}{=}\mathrm{tr}(W^T R_x W)$, with $W=V^T U-I$, is a contribution to total error value resulting from approximation of DCT, and $\epsilon_Q^{(II)}(U,\Theta)=\mathrm{tr}(U^T U)\kappa 2^{-2\Theta}\pi(V)$ represents the quantization error. Thus, also in this case, MSE depends on two components. However, the second one requires further explanation. It should be noted that the effectiveness of quantization depends on the coding transform, i.e., here on DCT (matrix V), what is manifested by $\pi(V)$ term. But in this case, the quantization error may also depend on U transformation (c.f. $\mathrm{tr}(U^T U)$ term in $\epsilon_Q^{(II)}(U,\Theta)$ component). It should be noted that in general $\mathrm{tr}(U^T U)\neq N$, where $\mathrm{tr}(U^T U)=\mathrm{tr}(UU^T)$, if only the row vectors of matrix U are not normalized. Definitely with normalized base vectors of U we always have $\mathrm{tr}(U^T U){=}N$.

On the basis of Eq. (10), it is possible to formulate the quality index $\chi_2(U)$ for the considered scenario. Then we have:

$$\chi_2(U){=}\epsilon_A^{(II)}(U)+\frac{1}{L}\left(\sum_{i=0}^{L-1}\epsilon_Q^{(II)}(U,\Theta_i)\right) \tag{11}$$

with $\Theta_i{=}\frac{1}{2}(i+1)$ for $i{=}0,1,\ldots,L-1$ and $L{=}12$.

The measurements of effectiveness of known DCT approximations are presented in Table 4. The obtained results are similar to those previously presented (see Table 2) with such differences that BC2012 transform is better than BAS2008III ($a=0.0, a=0.5, a=1.0$), BAS2008II outperformed SDCT2001, BAS2010, similarly SDCT2001 gave better results than BAS2010, and CB2011 outperformed SDCT2001,

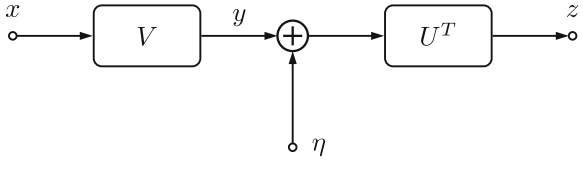

Fig. 5 Block diagram of asymmetric scenario of type II for data compression and decompression

Table 4 The results of analysis of known approximations of DCT in the case of the asymmetric scenario of type II ($\pi(V)$=0.131042)

Type	Adds	Shifts	$\chi_2(U)$	$\epsilon_A^{(II)}(U)$	Orthogonality
DR2014	12	0	13.577153	13.228013	-
PMCBKE2014	14	0	1.097419	0.631898	+
BAS2008III (a=0.0)	16	0	1.033841	0.568321	+
BAS2008III (a=1.0)	18	0	1.033722	0.568201	+
BAS2008III (a=0.5)	18	2	1.008255	0.542735	+
BC2012	14	0	0.940633	0.475113	+
PS2012	18	2	0.813411	0.347891	+
BAS2008I	18	2	0.655782	0.190261	+
BAS2010	24	4	0.633755	0.168235	+
SDCT2001	24	0	0.631355	0.165835	-
BAS2008II	21	0	0.618191	0.152670	-
CB2011	22	0	0.543922	0.078402	+
PMCBR2012	24	6	0.515187	0.049667	+

BAS2010 transformations. Such differences result from the fact that BC2012, BAS2008II, SDCT2001, and CB2011 transforms are characterized by better approximation of DCT (i.e., smaller values of $\epsilon_A^{(II)}(U)$ component), but worse performance in the sense of block quantization (i.e. higher values of the $\pi(U)$ product). However, in this scenario, the coding transformation is DCT and therefore the values of the $\pi(U)$ product do not affect the MSE. The highest and the lowest efficiency was obtained by PMCBR2012 (i.e., $\chi_2(U)\approx 0.51$) and DR2014 (i.e., $\chi_2(U)\approx 13.58$) transformations respectively.

The results in effectiveness of approximations generated with procedure proposed in Section 3 are collected in Table 5. Their analysis shows that for the following computational complexities (adds/shifts), i.e., (18/0) and (24/6), it was possible to obtain better results than with known approximations of the same computational complexities. For the complexities (14/0) and (22/0) the procedure generated known approximations in the form of BC2012 and CB2011 transforms respectively. The best result (i.e., $\chi_2(U)\approx 0.47$) was obtained with 28 additions and 10 bit-shift operations (see APRXII.9 in double underlined row of Table 5). The featured transformation, i.e., the one which can be characterized by good performance in the sense of $\chi_2(U)$ index at moderate computational complexity, is APRXII.6 described by the sixth row of Table 5 (underlined row). It allows to reach the following result $\chi_2(U)\approx 0.52$ with only 22 additions and 4 bit-shifts, which is better than almost all of the considered known approximations, including those with higher computational complexities (except PMCBR2012 transform but here the results are comparable).

4.2.3 Symmetric scenario

The last of the considered scenarios concerns a symmetrical case, i.e., the one where the same transformation is used at the compression and decompression stages. The simplified scheme of block quantization for this scenario is shown in Fig. 6.

It should be noted that at the decompression stage we have $\overline{U}$ transform, which is assumed for derivation purposes. The $\overline{U}$ matrix, depending on the properties of U transform, can take one of the following values: $\overline{U}=U^T$ for orthogonal transforms or $\overline{U}=U^{-1}$, when inverse transformation can be calculated by means of computationally effective

Table 5 The approximations of DCT generated by the proposed procedure in the case of the asymmetric scenario of type II

Name	Adds	Shifts	$\chi_2(U)$	$\epsilon_A^{(II)}(U)$	{ a, b, c, d, e, f, g }	Orthogonality
APRXII.1	14	0	0.940633	0.475113	{ 1, 1, 0, 0, 0, 0, 1 }	+
APRXII.2	16	2	0.910697	0.445176	{ 1, 2, 1, 0, 0, 0, 1 }	+
APRXII.3	18	0	0.632148	0.166628	{ 1, 1, 0, 0, 0, 1, 1 }	-
APRXII.4	20	2	0.602212	0.136691	{ 1, 1, $\frac{1}{2}$, 0, 0, 1, 1 }	-
APRXII.5	22	0	0.543922	0.078402	{ 1, 1, 0, 0, 1, 1, 1 }	+
APRXII.6	22	4	0.517743	0.052222	{ 1, 1, 0, 0, $\frac{1}{2}$, 1, 1 }	-
APRXII.7	24	2	0.513986	0.048465	{ 1, 1, $\frac{1}{2}$, 0, 1, 1, 1 }	+
APRXII.8	24	6	0.487806	0.022286	{ 1, 2, 1, 0, $\frac{1}{2}$, 1, 1 }	-
APRXII.9	28	10	0.474493	0.008973	{ 1, 2, 1, $\frac{1}{4}$, $\frac{1}{2}$, 1, 1 }	-

structure (i.e., only with additions and bit-shifts operations). In this case the approximation of DCT is understood in the sense of pursue for a standard of both high efficiency and fast computational structure, which undoubtedly is DCT. Further on it is easy to verify that in this scenario we have $z{=}\overline{U}(Ux+\eta)$. Then with analogous mathematical derivations, as in both previously discussed scenarios, we can come to the following formula for MSE, i.e.:

$$\epsilon_{MSE}^{(III)}(U,\overline{U}){=}\epsilon_A^{(III)}(U,\overline{U})+\epsilon_Q^{(III)}(U,\overline{U},\Theta), \tag{12}$$

where the first error component, which takes the following form $\epsilon_A^{(III)}(U,\overline{U}) = \mathrm{tr}(\overline{W}R_x\overline{W}^T)$, with matrix $\overline{W}$ defined as $\overline{W}{=}(\overline{U}U{-}I)$ is an error resulting from approximation of inverse matrix U^{-1} with matrix $\overline{U}$, evidently we have then that $\epsilon_A^{(III)}(U,U^{-1}){=}0$. The second component represents the quantization error, i.e., $\epsilon_Q^{(III)}(U,\overline{U},\Theta) = \mathrm{tr}(\overline{U}^T\overline{U})\kappa 2^{-2\Theta}\pi(U)$. It should be noted that the first term in quantization error component can take different values depending on the properties of U transform, i.e., (i) $\mathrm{tr}(\overline{U}^T\overline{U}){=}N$ when U is an orthogonal matrix, because then we have $\overline{U}{=}U^T$, which results in $\mathrm{tr}(\overline{U}^T\overline{U}){=}\mathrm{tr}(UU^T){=}N$; (ii) U matrix is not orthogonal but we assume $\overline{U}{=}U^T{\approx}U^{-1}$, which gives $\mathrm{tr}(\overline{U}^T\overline{U}){=}\mathrm{tr}(UU^T){=}N$ if only row vectors of matrix U are normalized; and (iii) $\mathrm{tr}(\overline{U}^T\overline{U}){\neq}N$ when matrix U is not orthogonal and $\overline{U}{=}U^{-1}$. In the third case, it is possible that $\mathrm{tr}(\overline{U}^T\overline{U}){>}N$ which may result in amplification of quantization error. On the basis of formula (12), we may construct the quality index of the form:

$$\chi_3(U,\overline{U}){=}\epsilon_A^{(III)}(U,\overline{U}){+}\frac{1}{L}\left(\sum_{i=0}^{L-1}\epsilon_Q^{(III)}(U,\overline{U},\Theta_i)\right) \tag{13}$$

with $\Theta_i{=}\frac{1}{2}(i+1)$ for $i{=}0,1,\ldots,L-1$ and $L{=}12$.

The benchmark results of known approximations for the considered scenario are collected in Table 6. In this experiment, we assume as $\overline{U}$ matrix the transpose of U, i.e., $\overline{U}{=}U^T$, for orthogonal transforms, and in the case of non-orthogonal approximations, i.e.,

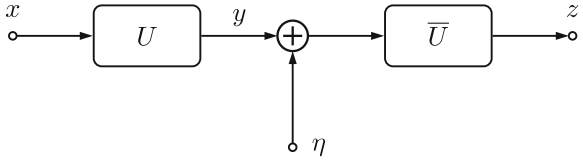

Fig. 6 Block diagram of symmetric scenario of data compression and decompression

Table 6 The results of analysis of known approximations of DCT in the case of the symmetric scenario ($\pi(V)$=0.131042)

Type	Adds	Shifts	$\chi_3(U,\overline{U})$	$\epsilon_A^{(III)}(U,\overline{U})$	$\mathrm{tr}(\overline{U}^T\overline{U})/N$	$\pi(U)$	Orthogonality
DR2014	12	0	5.782034 (2.545508)*	0.000000 (1.100000)*	3.000000 (0.750000)*	0.542537	-
BC2012	14	0	0.656552	0.000000	1.000000	0.184816	+
PMCBKE2014	14	0	0.656552	0.000000	1.000000	0.184816	+
BAS2008III (a=0.0)	16	0	0.574577	0.000000	1.000000	0.161740	+
BAS2008III (a=1.0)	18	0	0.574474	0.000000	1.000000	0.161711	+
BAS2008III (a=0.5)	18	2	0.547763	0.000000	1.000000	0.154192	+
BAS2008I	18	2	0.547763	0.000000	1.000000	0.154192	+
PS2012	18	2	0.534894	0.000000	1.000000	0.150570	+
BAS2008II	21	0	1.032007 (0.794446)*	0.000000 (0.215074)*	1.781250 (1.000000)*	0.163090	-
CB2011	22	0	0.539839	0.000000	1.000000	0.151962	+
SDCT2001	24	0	0.886988 (0.951579)*	0.000000 (0.360253)*	1.500000 (1.000000)*	0.166455	-
BAS2010	24	4	0.522419	0.000000	1.000000	0.147058	+
PMCBR2012	24	6	0.520192	0.000000	1.000000	0.146431	+

*Values in parentheses describe results obtained with $\overline{U}=U^T$; otherwise we have $\overline{U}=U^{-1}$.

DR2014, BAS2008II, and SDCT2001, we explore two possible variants, i.e., $\overline{U}=U^T$ and $\overline{U}=U^{-1}$ (the second variant is possible because for all non-orthogonal approximations computationally effective inverse transformations have been formulated in literature).

An analysis of the $\chi_3(U,\overline{U})$ values from Table 6 shows that the best result, i.e., the smallest index value, was possible to be obtained with PMCBR2012 approximation, while the worst result belongs to the least computationally demanding DR2014 transform. Moreover, in the case of non-orthogonal approximations, as we can see, it is more advantageous to use transposed transform matrix U, i.e., $\overline{U}=U^T$, at the decompression stage, instead of its inverse, i.e., $\overline{U}=U^{-1}$, for DR2014 and BAS2008II approximations. In the case of SDCT2001, we can draw the contrary conclusion. Such twofold behavior of the mentioned non-orthogonal approximations of DCT is a direct consequence of the balance in the proportions between the values taken by $\epsilon_A^{(III)}(U,\overline{U})$ error component, and the values of $\mathrm{tr}(\overline{U}^T\overline{U})/N$ term, which on the other hand determines the final amount of quantization error (c.f. first, ninth and eleventh rows of Table 6).

The results obtained in generating the DCT approximations with use of the parametric model and the proposed procedure are presented in Table 7. In this experiment at the decompression stage, we used a transposed matrix of U transformation, i.e., $\overline{U}=U^T$, which corresponds to the following heuristic, i.e., $U^T \approx U^{-1}$. The reason for this is that in the considered parametric model it is not possible in the general case to calculate the inverse matrix to B using only additions and bit-shift operations. An analysis of obtained results reveals that: (i) for the computational complexities (adds/bit-shifts) of (14/0), (22/0) and (24/0) it was possible to find better or equivalent approximations in the sense of $\chi_3(U,\overline{U})$ index, but starting up from the computational complexity of (24/2) the generated approximations allowed to obtain results even better than the best known approximation PMCBR2012, while in other cases found transforms can be characterized by worse effectiveness that known approximations with comparable computational complexities; (ii) the best found approximation APRXIII.12 requires 28 additions and 10 bit-shifts and it is not orthogonal, but the approximation error component takes relatively small value $\epsilon_A^{(III)}(U,U^T)\approx 0.004$ (double underlined row of Table 7; (iii) the featured

Table 7 The approximations of DCT generated by the proposed procedure in the case of the symmetric scenario ($\pi(V)$=0.131042)

Name	Adds	Shifts	$\chi_3(U,\overline{U})$	$\epsilon_A^{(III)}(U,\overline{U})$	$\mathrm{tr}(\overline{U}^T\overline{U})/N$	$\pi(U)$	{a, b, c, d, e, f, g}	Orthogonality
APRXIII.1	14	0	0.656552	0.000000	1.000000	0.184816	$\{1, 0, 1, 0, 0, 0, 1\}$	+
APRXIII.2	16	0	0.656434	0.000000	1.000000	0.184783	$\{1, 1, 1, 0, 0, 0, 1\}$	+
APRXIII.3	16	2	0.625912	0.000000	1.000000	0.176191	$\{1, 2, 1, 0, 0, 0, 1\}$	+
APRXIII.4	20	2	0.620702	0.000000	1.000000	0.174724	$\{1, 2, 1, 1, 0, 0, 1\}$	+
APRXIII.5	20	6	0.619900	0.021828	1.000000	0.168354	$\{1, 2, 1, 0, 0, \frac{1}{8}, 1\}$	-
APRXIII.6	20	10	0.617859	0.005585	1.000000	0.172352	$\{1, 2, 1, 0, 0, \frac{1}{8}, 2\}$	-
APRXIII.7	22	0	0.539839	0.000000	1.000000	0.151962	$\{1, 1, 0, 0, 1, 1, 1\}$	+
APRXIII.8	24	0	0.539742	0.000000	1.000000	0.151934	$\{1, 1, 1, 0, 1, 1, 1\}$	+
APRXIII.9	24	2	0.514646	0.000000	1.000000	0.144870	$\{1, 1, \frac{1}{2}, 0, 1, 1, 1\}$	+
APRXIII.10	26	8	0.503287	0.004210	1.000000	0.140487	$\{1, 0, 1, \frac{1}{4}, \frac{1}{2}, 1, 1\}$	-
APRXIII.11	28	6	0.503197	0.004210	1.000000	0.140462	$\{1, 1, 1, \frac{1}{4}, \frac{1}{2}, 1, 1\}$	-
APRXIII.12	28	10	0.479996	0.004210	1.000000	0.133931	$\{1, 2, 1, \frac{1}{4}, \frac{1}{2}, 1, 1\}$	-

approximation APRXIII.9 is described by ninth row of Table 7 and it requires 24 additions and 2 bit-shifts and allows to obtain results better than all known approximations (see underlined row); and (iv) since during the experiment it was assumed that $\overline{U}=U^T$, then it is clear that for all generated approximations the term $\mathrm{tr}(\overline{U}^T\overline{U})/N$ equals one.

5 Results and discussion

In order to verify the effectiveness of automatically generated approximations of 8-point DCT, we carried out experiments in compression of natural images (operating on grayscale images) including all of the considered scenarios. The experiments were performed using popular realization of two-dimensional block quantization in the form of JPEG standard. During experiments, we used both the quantization and Huffman code tables recommended by the standard specification for the luminance component. The obtained results are shown in the form of plots of image quality estimations evaluated with popular Peak Signal-to-Noise Ratio (PSNR) measure against the entropy expressed here as a number of bits per single pixel (bpp) which, of course, allows for direct evaluation of the obtained ratio of compression (for grayscale images the initial entropy equals 8 bpp). For the asymmetric scenario of type I the PSNR plots for "Lena.bmp" and "Mandrill.bmp" images (individual results), and also averaged results over the set of "Lena.bmp", "Goldhill.bmp", "Mandrill.bmp" and "Pentagon.bmp" images are presented in Figs. 7, 8, and 9 respectively. The corresponding results for asymmetric scenario of type II and also for symmetric scenario are collected in Figs. 10, 11, 12 , 13, 14, and 15. In addition, to provide an objective assessment of the effectiveness of the considered transformations, we also include the results in the form of fragments of images obtained for different levels of entropy (bpp) after the decompression stage. In Figs. 16, 17, and 18, we show the representative results in the form of the fragments of "Lena.bmp" image obtained after decompression at the following values of entropy, i.e., 0.3, 1.0, and 1.5 bpp.

An analysis of plots from Figs. 7, 8, and 9 shows that the best results were obtained with generated approximation APRXI.11, which fully corresponds to the results of previous experiments (see Table 3). For example, the advantage of that approximation over the best among the others for the entropy value of 1.5 bits per pixel was at the level of 2.75 dB for "Lena.bmp" image, 0.6 dB for "Mandrill.bmp" image, and at the level of 2 dB

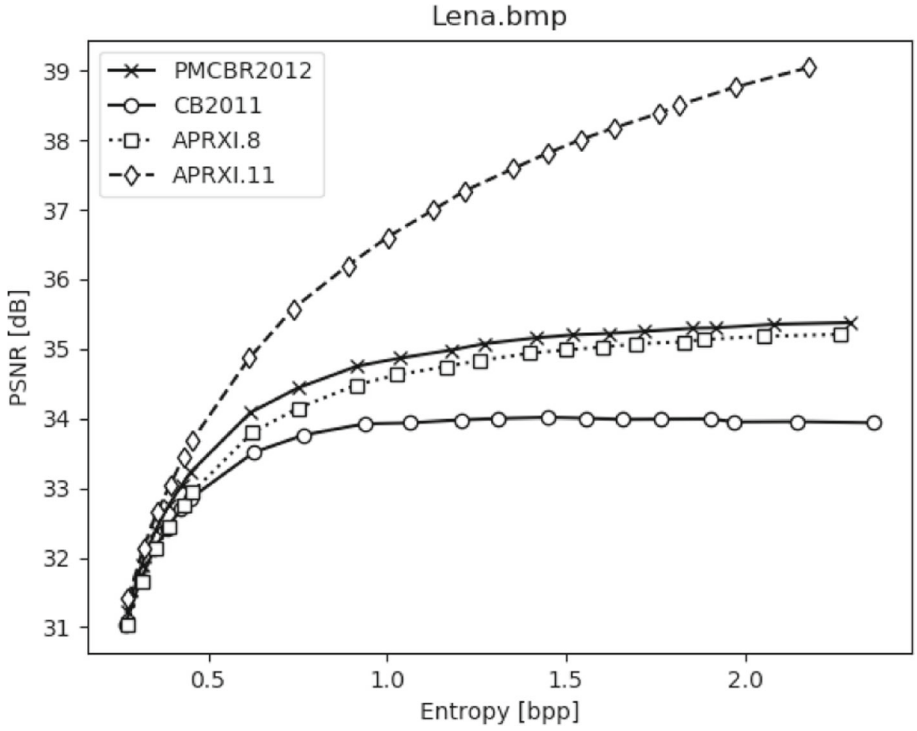

Fig. 7 Results in compression of natural images for asymmetric scenario of type I and "Lena.bmp" image

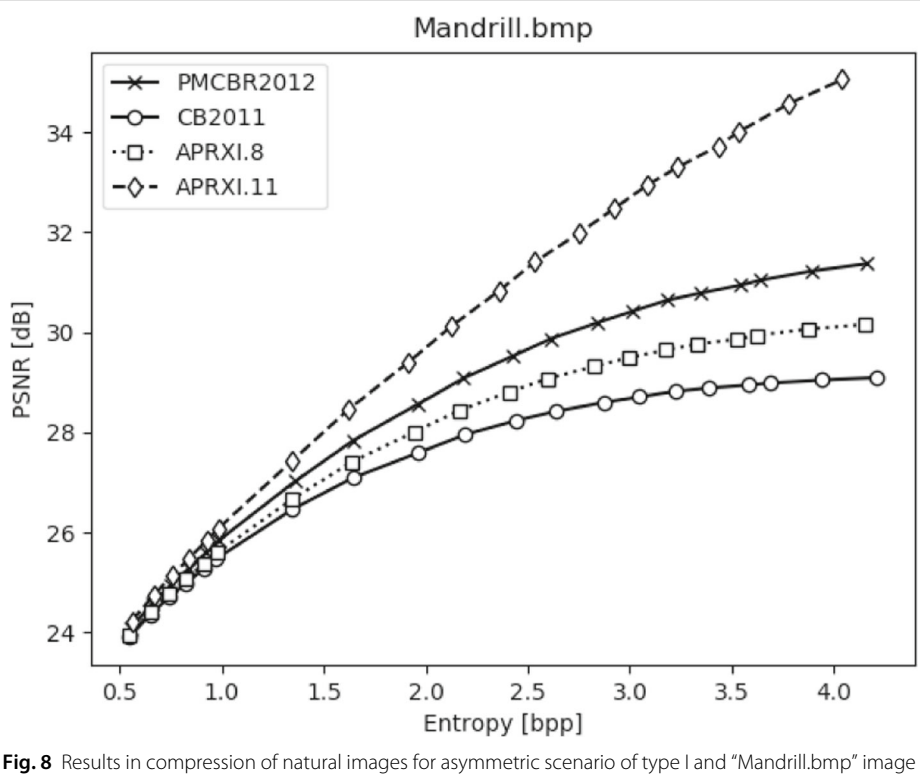

Fig. 8 Results in compression of natural images for asymmetric scenario of type I and "Mandrill.bmp" image

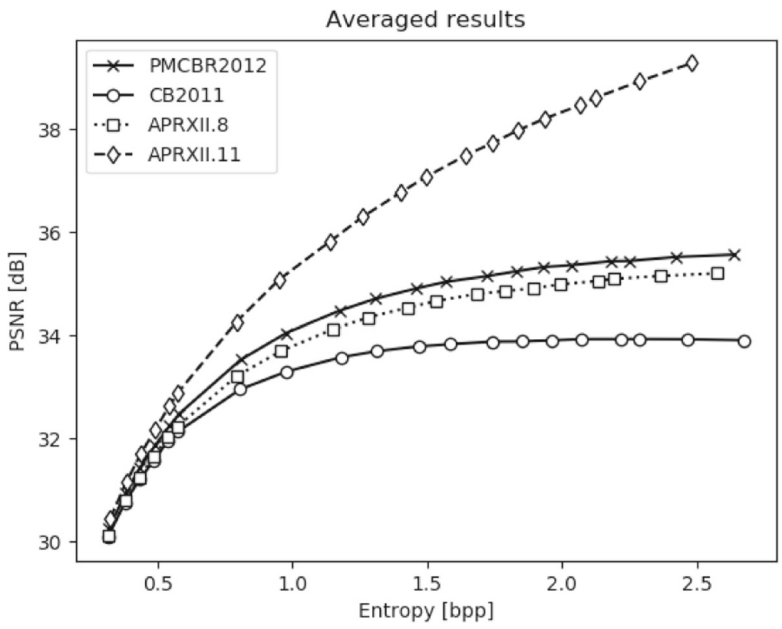

Fig. 9 Averaged results in compression of natural images for asymmetric scenario of type I

in average. Equally important is the fact that the PSNR value for this approximation constantly increases with the increase of entropy, which indicates a good approximation of the DCT transform. The performance of the best known approximation PMCBR2012 and the featured APRXI.8 is comparable; however, PMCBR2012 gives always better results, e.g., for "Lena.bmp" and 0.4 to 0.9 bpp its advantage is highest, and at the level of 0.3 dB, for "Mandrill.bmp," it is constantly growing to obtain the highest observed value of 1.3 dB at 4 bpp, and finally in average, it is better by about 0.25 dB starting from 0.8 bpp. But, as it should be noted, the featured approximation requires less by 2 additions and 2 bit-shift operations. The second best of known approximations, i.e., CB2011 transform, gave results worse than PMCBR2012 and the generated approximations (for example in the average sense the smallest difference is about 2 dB at 2 bpp and grows with the increase of the entropy). However, this transform is the least computationally demanding among transforms considered in this experiment. It should be also noted that the generated approximations give much better results for images with high correlation (c.f. results obtained with "Lena.bmp" and "Mandrill.bmp" images). The possible reason for this is the assumption of a high correlation coefficient $\rho = 0.95$ in the transform generating procedure (see Section 3). The subjective analysis of the performance of DCT approximations based on results from Figs. 16, 17, and 18 allows to draw conclusions corresponding to the results presented in the form of the PSNR plots. It can be seen that the least annoying artifacts appear with APRXI.11 approximation, while in the case of other approximations the artifacts in the form of vertical and horizontal lines are especially visible in places of sharp edges in image (see the hat's hem or the rim of the eye). Subjectively these artifacts are the most visible for PMCBR2012 and CB2011 approximations.

In the second experiment, regarding the asymmetric scenario of type II, the obtained results are analogous to those from the first experiment (see Figs. 10, 11, and 12). The

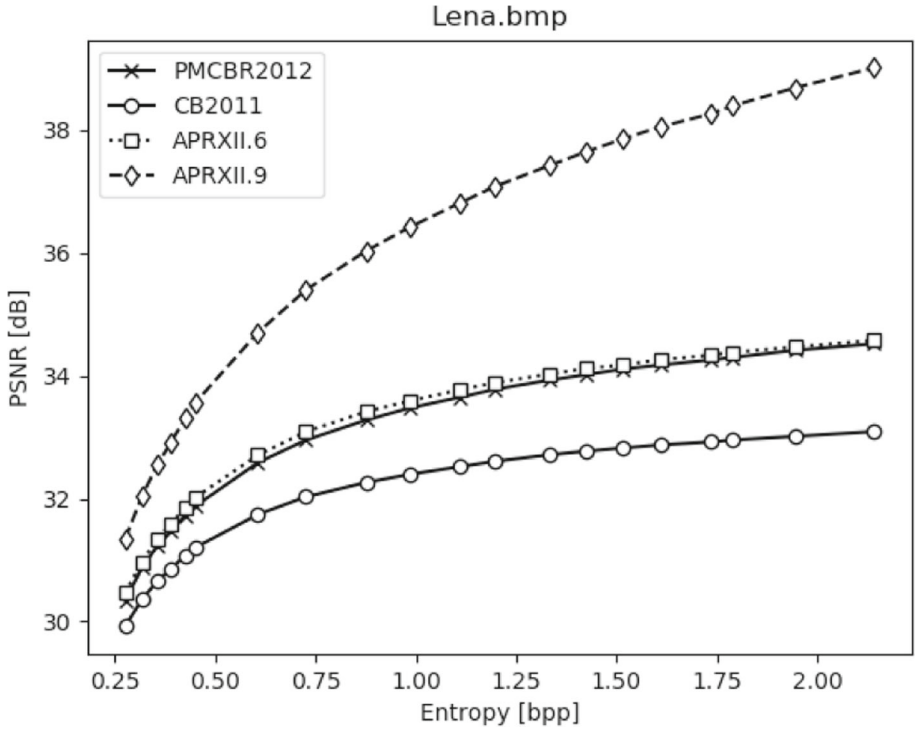

Fig. 10 Results in compression of natural images for asymmetric scenario of type II and "Lena.bmp" image

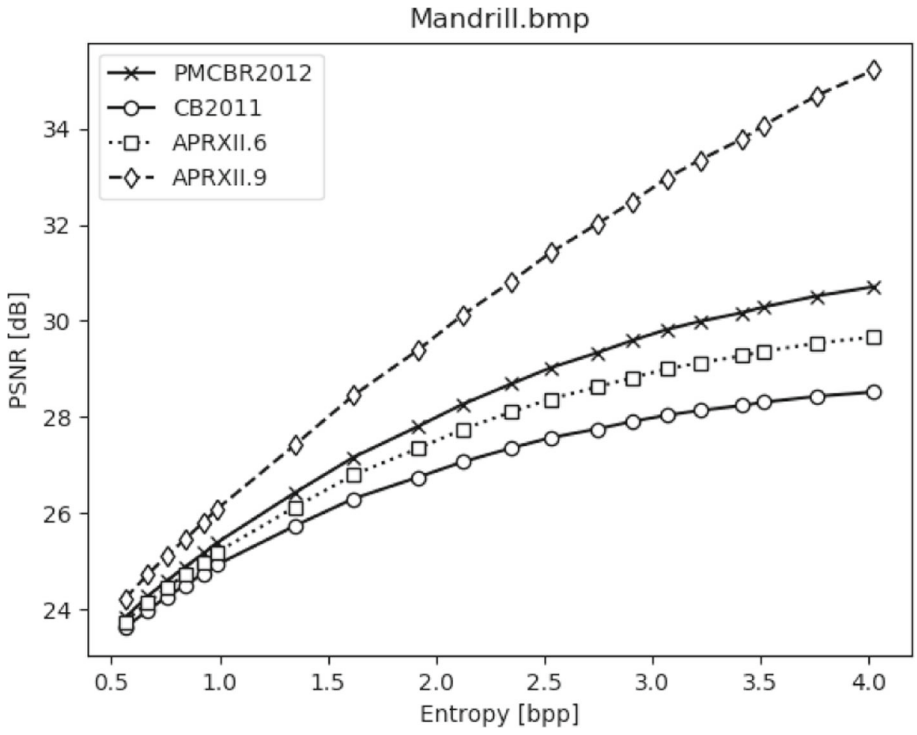

Fig. 11 Results in compression of natural images for asymmetric scenario of type II and "Mandrill.bmp" image

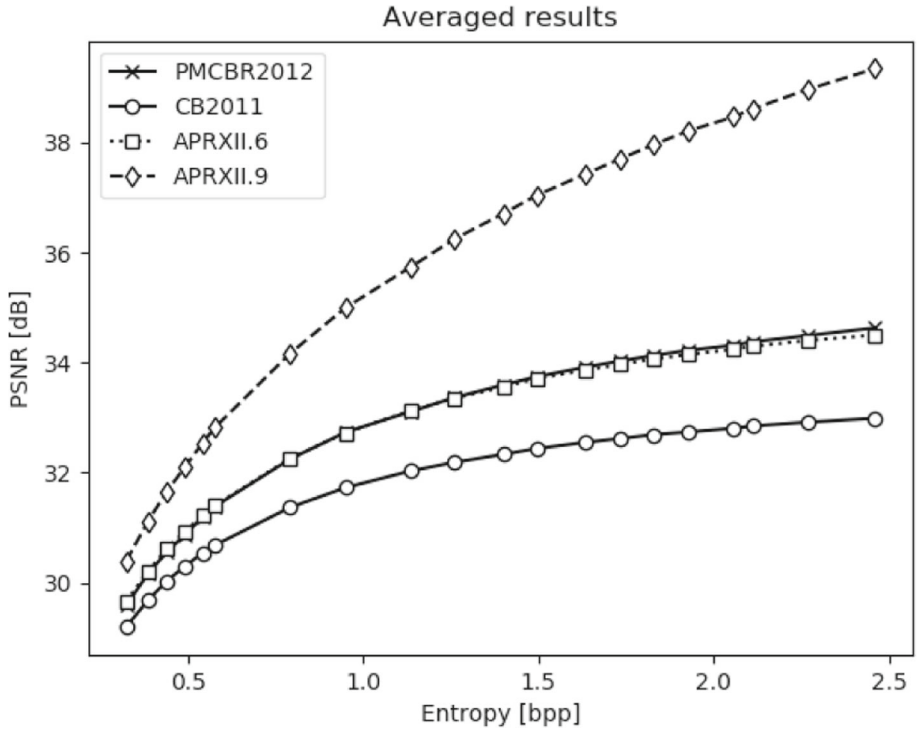

Fig. 12 Averaged results in compression of natural images for asymmetric scenario of type II

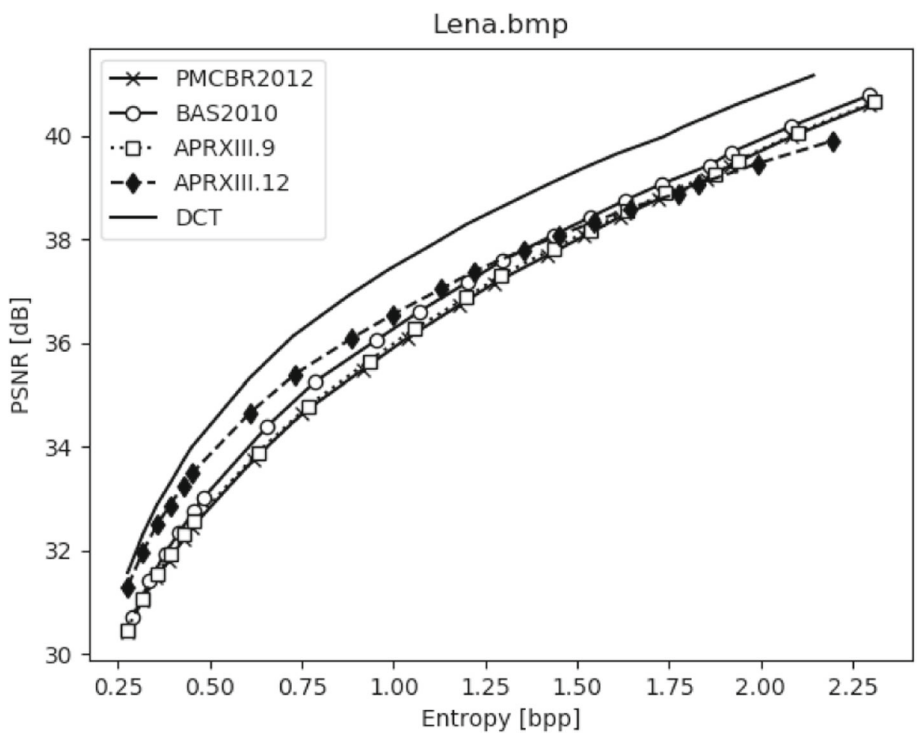

Fig. 13 Results in compression of natural images for symmetric scenario and "Lena.bmp" image

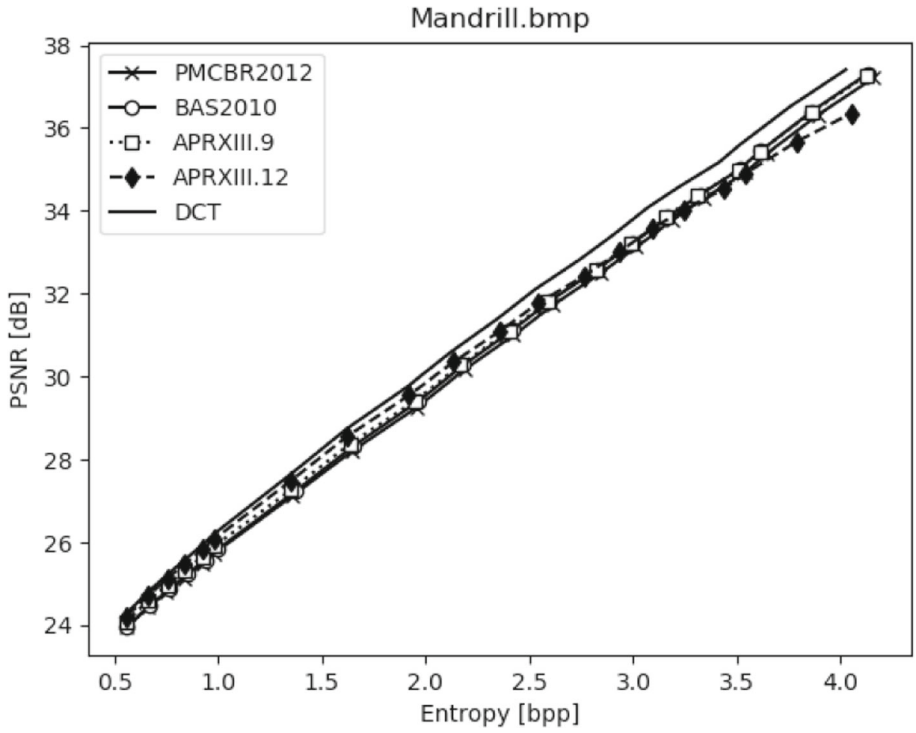

Fig. 14 Results in compression of natural images for symmetric scenario and "Mandrill.bmp" image

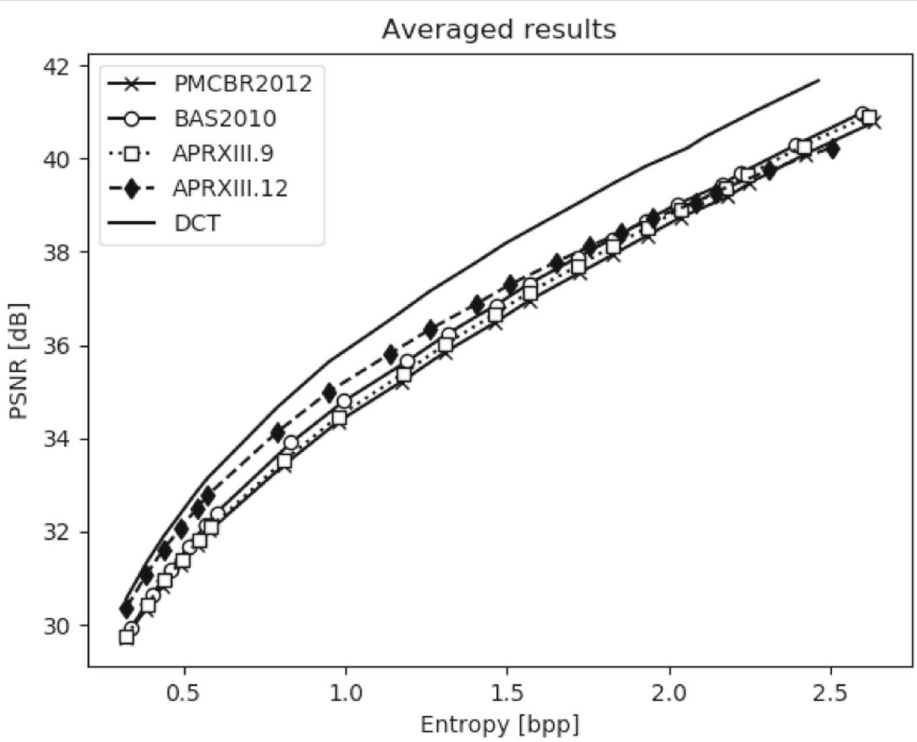

Fig. 15 Averaged results in compression of natural images for symmetric scenario

Fig. 16 Fragments of images showing visible artifacts at approximately 0.3 bpp in case of asymmetric scenario of type I

best results were obtained with generated approximation APRXII.9 and among known approximations the PMCBR2012 transform can be characterized by the highest effectiveness. The featured approximation APRXII.6 with 22 additions and 4 bit-shift operations allowed to obtain results comparable to known PMCBR2012 transform (even better for "Lena.bmp" image and very close in the averaged sense) but with smaller computational complexity. The second best among known approximations, i.e., CB2011, gave average results worse by even about 2 dB (starting from 0.8 bpp entropy). The characteristics and strength of visible artifacts in images in the case of subjective performance evaluation were analogous to those observed in the previous scenario. Hence, we present such results only for the case of 1.5 bpp entropy (see Fig. 19).

In the case of the symmetric scenario, it can be seen that differences between results obtained with various approximations of DCT are very small. In addition, based on averaged results (see Fig. 15), we can conclude that the choice of the best approximation depends strictly on the values of entropy. For entropy smaller than around 1.8 bpp the APROXIII.12 guarantees the highest effectiveness (e.g., higher by 0.5 dB at 0.5 bpp). But starting up from 1.8 bpp, its efficiency drops and the best results can be obtained with known BAS2010 approximation. It can be stated that APROXIII.12 is the best at high compression ratios. The high efficiency of this approximation observed in the first interval of entropy variability is a consequence of small value of $\pi(U)$ component, which translates into small values of quantization error. But the considered approximation is not orthogonal and the error component $\epsilon_A^{(III)}(U,\overline{U})$ for $\overline{U}=U^T$ is not zero (i.e., to be more precise $\epsilon_A^{(III)}(U,\overline{U})\approx 0.0042$). This is the direct reason for the drop of effectiveness for the remaining values of entropy. Plainly, starting from an entropy value of 1.8 bpp, the error component resulting from approximation of inverse matrix U^{-1} by transposed matrix U^T becomes dominant. However, we should have in mind that

Fig. 17 Fragments of images showing visible artifacts at approximately 1.0 bpp in case of asymmetric scenario of type I

Fig. 18 Fragments of images showing visible artifacts at approximately 1.5 bpp in case of asymmetric scenario of type I

APROXIII.12 it is the most computationally complex among the considered approximations. Finally, it should be noted that the featured approximation APROXIII.9 allows to obtain results close to that acquired with the best among known approximation, i.e., BAS2010, also better but very close to those characteristic for PMCBR2012 transform, and all of that with smaller computational demands. An analysis of visible artifacts (see exemplary results in Fig. 20) reveals comparable characteristics of the considered approximations.

6 Hardware realizations

Bearing in mind the fact that hardware implementations of DCT and its approximations in FPGA or Application-Specific Integrated Circuits (ASICs) are significant areas of practical realizations, we consider in this section the hardware designs of selected among generated approximations, i.e., APRXI.1, APRXI.7, APRXI.8 (APRXII.6), APRXI.10, APRXI.11, and also APRXII.9 (APRXIII.12), APRXIII.9. For these transforms, hardware requirements for FPGA implementations are specified, including the required number of logical elements and the predicted consumption of energy (see Table 8). In addition, for APRXIII.9 transform the detailed description of its design for FPGA circuit is provided. The obtained results in hardware utilization by proposed approximations are compared to the equivalent results obtained for well-known approximations, i.e., PMCBKE2014, BC2012, CB2011, BAS2010, and PMCBR2012. The aim of this part of the paper is to verify the feasibility of hardware implementations of the considered approximations. Such transforms, as we know, use only additions and bit-shift operations. This feature should translate directly into low energy and resources consumption.

The APRXIII.9 approximation is the featured transform from the experiment concerning the symmetric scenario. This transformation can be described by the following

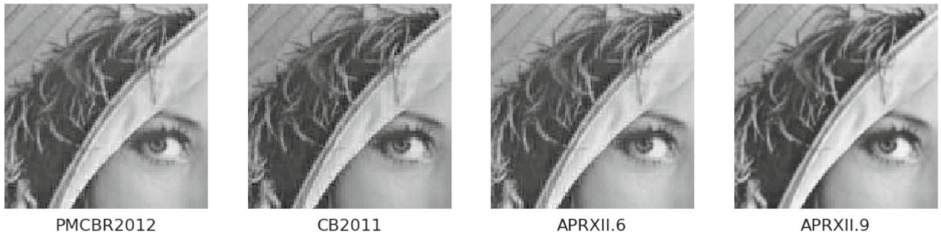

Fig. 19 Fragments of images showing visible artifacts at approximately 1.5 bpp in case of asymmetric scenario of type II

Fig. 20 Fragments of images showing visible artifacts at approximately 1.5 bpp in case of symmetric scenario

parameters, i.e., $\{1, 1, \frac{1}{2}, 0, 1, 1, 1\}$, and it requires 24 additions and 2 bit-shift operations. Since two-dimensional transformation can be realized with one dimensional transforms, i.e., one for rows, and one for columns, in this place, we consider only hardware implementation of one-dimensional 8-point transformation, but in two variants, i.e., regarding rows and columns respectively. Further on we assume that an input image is represented as two-dimensional array with 8-bit integer values describing pixel colors (i.e., luminance or one of the chrominance components interchangeably). On this basis, we can assume that pixel data is stored using 8-bit two's complement numbers format, i.e., integer numbers form an interval $[-128, 127]$. It should be noted, however, that in two's complement number format addition or subtraction operations of two n-bit arguments in general require $n+1$ bits to represent their results. This is a reason why the width of data representation must increase at the following stages of data flow diagram of transform implementation structure (c.f. Fig. 21). For example in the considered case for row variant input data is 8-bit wide, while output results require 12 bits to represent data. In column variant the width of data changes from 12 to 16 bits. This is the only difference between row and column designs of logic circuits, which of course should result in proportionally higher demands on hardware resources for the column transformation.

The data flow diagram of the hardware implementation of APRXIII.9 transform is presented in Fig. 21. It will be analyzed in terms of transform operating on 8-bit image data (i.e., first from the row-column order). In the first place, input data in the form of vector $[x(0), x(1), \ldots, x(7)]$ with eight elements goes through the first stage composed of eight adders: A00, A01 to A07, and eight registers built with D-flops, i.e., D00, D01 to D07. The first stage is the implementation of U_3 matrix from the considered parametric model. In

Table 8 Resource utilization with hardware implementation for Altera Cyclone IV GX device

Name	Logic cells	Maximum delay	Total dissipated power	Maximum frequency
PMCBKE2014	134	2.673	0.70 mW (at 100 MHz)	374 MHz
BC2012	150	2.764	0.85 mW (at 100 MHz)	361 MHz
CB2011	218	3.303	1.18 mW (at 100 MHz)	302 MHz
BAS2010	242	2.697	1.21 mW (at 100 MHz)	370 MHz
PMCBR2012	242	3.057	1.26 mW (at 100 MHz)	327 MHz
APRXI.1	150	2.764	0.85 mW (at 100 MHz)	361 MHz
APRXI.7	218	3.058	1.22 mW (at 100 MHz)	327 MHz
APRXI.8 (APRXII.6)	219	2.923	1.23 mW (at 100 MHz)	351 MHz
APRXI.10	243	2.986	1.54 mW (at 100 MHz)	334 MHz
APRXI.11	285	3.024	1.45 mW (at 100 MHz)	330 MHz
APRXII.9 (APRXIII.12)	278	3.001	1.42 mW (at 100 MHz)	333 MHz
APRXIII.9	240	2.827	1.25 mW (at 100 MHz)	353 MHz

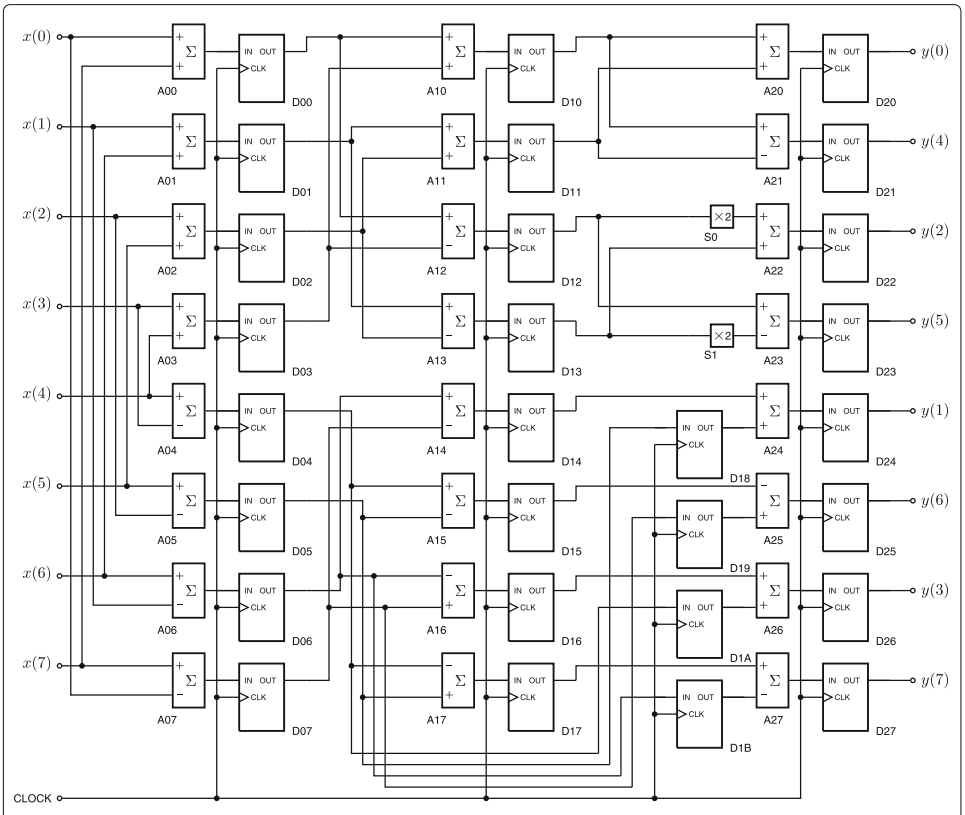

Fig. 21 The block diagram of hardware implementation of APRXIII.9 transform with parametrization: $\{1, 2, 1, 0, 1, 1, 1\}$

the case of row transformation input data is represented with 8-bit two's complement format. Before performing operations within the first stage data must be extended to 9 bits by duplicating the most significant bit (MSB), which simply takes the value of MSB from 8-bit representation (simple duplication of MSB). In result, the adders A00 to A07 and registers D00 to D07 are 9-bit wide. All D*nm* registers in this implementation are used to enable calculations in the pipeline mode. The second stage is composed of eight adders and twelve registers. The input data, which is now 9-bit wide, must be extended to 10-bit representation by duplicating the MSBs. The registers D18, D19, D1A, and D1B are needed to delay by one clock cycle the outputs of D04 to D07, required at the third stage. Further on, at the third stage, we have eight adders A20 to A27, two bit-shift registers S0 and S1, and eight data holding registers D20 to D27. Due to the care for efficiency of hardware implementation (i.e., the reduced data width at the first two stages), the mentioned bit-shift registers S0 and S1 make data shifts by one bit left, which corresponds to the following set of parameters: $\{1, 2, 1, 0, 1, 1, 1\}$. From the point of view of the parametric model, this is precisely the same transformation as obtained with the following parametrization, i.e., $\{1, 1, \frac{1}{2}, 0, 1, 1, 1\}$. Since, both left bit-shift and addition operations increase the data size each by one bit, then the input data for this stage must be transformed in an analogous way (by duplicating their MSBs) from 10-bit to 12 bit representation. At the same time this is the width of the output data, i.e., $[y(0), y(1), \ldots, y(7)]$. It should be noted that bit-shift operations by a constant number of bits can be simply hardwired and thus do not constitute an additional resource burden.

Both row and column variants of APRXIII.9 approximation were implemented in FPGA using Altera Cyclone IV GX EP4CGX15BF14C6 chipset. The resource utilization in both cases equaled 240 and 336 of logic elements (LEs) respectively, which constitutes approximately 1.7% and 2.4% of a total number of LEs available in this basic chipset (i.e., the total number of LEs equals 14,400). Moreover the timing analysis of both considered implementations showed that the time of data propagation between stages is smaller than 3.13 ns, which is also the time required to calculate approximate DCT for one input vector when the data pipeline is filled.

The remaining among selected approximations, as well as the selected well-known approximations, were also implemented with the same chipset but only in a row variant, i.e., operating on 8-bit input data. The obtained results in the form of LEs utilization and energy consumption are collected in Table 8. It should be noted that energy consumption, i.e., the total dissipated power (predicted for 100 MHz clock frequency), concerns only the part of the chipset responsible for realization of the specific transformation. The values of power consumption are very low as the usage of hardware resources is also very low.

The analysis of results from Table 8 allows to compare selected pairs of transforms according to similar computational complexities. In case of asymmetric scenario of type I, we can indicate two pairs of transforms, namely, APRXI.8 - CB2011, and APRXI.11 - PMCBR2012. The APRXI.8 approximation allows to obtain much better results in image compression than CB2011, and although it requires additional 4 bit-shift operations, its hardware utilization is almost identical to the one imposed by CB2011 approximation (219 LEs vs. 218 LEs). It is possible because in FPGA realizations bit-shift operations can be hard-wired and do not require additional resources. In case of the second pair of transforms, APRXI.11 requires more LEs by around 18% than PMCBR2012 transform. However, higher utilization of hardware resources is fully compensated by the best results in image compression obtained among the considered transforms. In asymmetric scenario of type II, we can indicate the following two pairs of transforms: APRXII.6 - CB2011, and APRXII.9 - PMCBR2012. Since APRXII.6 is precisely the same transform as APRXI.8, its comparison with CB2011 leads to the same conclusions. Here, the best results in compression can be obtained with APRXII.9 approximation. But its implementation requires by around 15% more LEs than the implementation of the best known approximation PMCBR2012. However, the additional implementation costs can be fully compensated by the possible improvement of image quality obtained after decompression stage. The symmetric scenario allows to select another two pairs of transforms with similar computational complexities. We mean here transforms: APRXIII.9 - BAS2010 and APRXIII.12 - PMCBR2012. The APRXIII.9 allows to obtain better results in compression than PMCBR2012 with slightly lower hardware utilization (less by 2 LEs). The BAS2010 transform is better in the sense of results in image compression and it can be characterized by the same hardware utilization as PMCBR2012 transform (the difference in the number of bit-shift operations is not reflected in resource utilization). The best results in compression can be obtained with APRXIII.12 transform. However, it requires more by around 15% of LEs than BAS2010 and PMCBR2012 transforms.

The general analysis of results from Table 8 indicates two classes of transforms, which can be characterized by very small hardware utilization around 150 LEs (PMCBKE2014, BC2012, or APRXI.1), and utilization higher than 218 LEs (the remaining ones). Although

the transforms belonging to the first group can be characterized by very low hardware utilization, they do not allow to obtain good results in image compression. Among the transforms belonging to the second group, the utilization of LEs is comparable and differs at most by 28%. Of course the higher utilization of LEs, the best results in compression can be obtained, which is fully expected when considering theoretical computational complexities of those transforms. It should be noted that bit-shift operations have no effect on utilization of hardware resources, which is also fully expected. The energy consumption is proportional to the utilization of LEs and among the transforms belonging to the second group the difference is at most around 30%.

7 Conclusions and the future work

In this paper, the approximate variants of 8-point discrete cosine transform (DCT) were generated with aid of the parametric matrix model and by taking into account three scenarios of practical usage. The found approximations can be characterized by (i) effective computational structures requiring only additions and bit-shift operations; (ii) in many cases higher effectiveness in tasks of lossy compression of data than known approximations of the same class, i.e., multiplierless transforms with comparable computational complexity (e.g., in lossy compression of natural images, it was possible to obtain results better by at least about 3 dB starting from 2 bpp entropy in assymetric scenarios and approximately 0.5 dB for entropy in range 0.3 to 0.8 bpp for symmetric scenario when compared with the best known approximation PMCBR2012 [15]); and (iii) good adaptation to the considered scenarios of compression and decompression of data. It should be noted that in the paper three scenarios of signal compression and decompression were considered, i.e., two asymmetric scenarios, where DCT and approximate transform are used interchangeably at the compression and decompression stages, as well as a symmetric scenario, where the approximate transform is used both to compress and decompress data. The approximations found can be treated as a unique dictionary of transformations ranked in relation to the increasing efficiency, and also the increasing computational complexity, and as such they can be freely chosen according to the needs of a particular application. In addition, the experiments in the tasks of lossy compression of natural images were carried out using JPEG compression standard, which, as it was already exemplified, confirmed the effectiveness of selected from obtained approximations. In the last part of the paper, we consider the issues of hardware implementation of the approximations of 8-point DCT. The aim of this task was to demonstrate in practice that transforms requiring only additions and bit-shift operations can be effectively implemented in FPGA devices. The obtained results fully confirm this assumption. For example, an approximate transform with 24 additions and 2 bit-shift operations required only 240 or 336 logic elements depending on whether it was row or column transform while considering the typical row-column approach to calculation of two-dimensional transform.

The results obtained in this paper, as well as in paper [16], confirm the validity of approaches based on parametric matrix models and exhaustive search of the parameters' space. Taking into account relatively small number of parameters, what results from the small size of transformation, it can be stated that naive techniques are completely sufficient. A different approach might be the techniques based on evolutionary programming which would allow for simultaneous search for the model and its parametrization.

Nevertheless, the possible directions of future research may refer to other matrix models, in particular including those based on well known factorizations of fast algorithms for calculation of discrete cosine transform in the form of Chen et al. [30] and Loeffler et al. [1] approaches.

Abbreviations

ASIC: Application-specific integrated circuit; BAS: Approximations of 8-point DCT by Bouguezel, Ahmad and Swamy; BC, CB: Approximations of 8-point DCT by Bayer and Cintra; DCT: Discrete cosine transform; DR: Approximation of 8-point DCT by Dhandapani and Ramachandran; FCT: Fast discrete cosine transform; FPGA: Field-programmable gate array; GPU: Graphics processing unit; JPEG: Joint photographic experts group; KLT: Karhunen-Loève transform; LE: Logic element; MPEG: Moving picture experts group; MSB: Most significant bit; MSE: Mean squared error; PMCBKE: Approximation of 8-point DCT by Potluri, Madanayake, Cintra, Bayer, Kulasekera and Edirisuriya; PMCBR: Approximation of 8-point DCT by Potluri, Madanayake, Cintra, Bayer and Rajapaksha; PS: Approximation of 8-point DCT by Puchala and Stokfiszewski; PSNR: Peak signal-to-noise ratio; RLE: Run-length encoding; SDCT: Square "wave" discrete cosine transform; SIMD: Single instruction, multiple data

Acknowledgements

The author would like to thank Prof. Mykhaylo Yatsymirskyy for his valuable comments, which allowed to significantly improve the quality of this paper.

Authors' information

Institute of Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, ul. Wolczanska 215, 90-924 Lodz, Poland, e-mail: dariusz.puchala@p.lodz.pl.

Author's contributions

The author read and approved the final manuscript.

Funding

This work was supported by the Institute of Information Technology at the Lodz University of Technology, Lodz, Poland.

Availability of data and materials

Not available online. Please contact author for data requests.

Declarations

Competing interests

The author declares that he has no competing interests.

Received: 15 October 2018 Accepted: 19 April 2021
Published online: 17 May 2021

References

1. C. Loeffler, A. Ligtenberg, G. S. Moyschytz, in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Practical Fast 1-D DCT Algorithms with 11 Multiplications, vol. 2, (1989), pp. 988–991
2. P. Duhamel, H. H'Mida, in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, New 2^n DCT Algorithms suitable for VLSI Implementation, (1987), pp. 1805–1808
3. S. Winograd, Some bilinear forms whose multiplicative complexity depends on the field of constants. Math. Syst. Theory. **10**, 169–180 (1977)
4. Y. Arai, T. Agui, M. Nakajima, A fast DCT-SQ scheme for images. IEICE Trans. **E-71**(11), 1095–1097 (1988)
5. V. Dhandapani, S. Ramachandran, Area and power efficient DCT architecture for image compression. EURASIP J. Adv. Signal Process. **2014**(180), 1–9 (2014). https://doi.org/10.1186/1687-6180-2014-180
6. T. I. Haweel, A new square wave transform based on the DCT. Signal Process. **81**, 2309–2319 (2001)
7. S. Bouguezel, M. O. Ahmad, M. N. S. Swamy, Low-complexity 8x8 transform for image compression. Electron. Lett. **44**(21), 1249–1250 (2008)
8. S. Bouguezel, M. O. Ahmad, M. N. S. Swamy, in *Proc. of International Conference on Signals, Circuits and Systems*, A multiplication-free transform for image compression, (2008), pp. 2145–2148
9. S. Bouguezel, M. O. Ahmad, M. N. S. Swamy, in *Proc. of IEEE International Symposium of Circuits and Systems (ISCAS)*, A low-complexity parametric transform for image compression, (2011), pp. 2145–2148
10. S. Bouguezel, M. O. Ahmad, M. N. S. Swamy, in *Proc. of 53rd IEEE International Midwest Symposium on Circuits and Systems*, A novel transform for image compression, (2010), pp. 509–512
11. R. J. Cintra, F. M. Bayer, A DCT approximation for image compression. IEEE Signal Process. Lett. **18**(10), 579–582 (2011)
12. F. M. Bayer, R. J. Cintra, DCT-like transform for image compression requires 14 additions only. Electron. Lett. **48**(15), 919–921 (2012)
13. D. Puchala, K. Stokfiszewski, Low-complexity approximation of 8-point DCT for image compression. J. Appl. Comp. Sci. **20**(2), 107–117 (2012)
14. R. L. de Queiroz, in *Proc. of IEEE International Conference on Image Processing*, DCT approximation for low bit rate coding using a conditional transform, vol. 1, (2002), pp. 237–240

15. U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, N. Rajapaksha, Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. Meas. Sci. Technol. **23**(11), 1–15 (2012)
16. U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, A. Edirisuriya, Improved 8-point approximate DCT for image and video compression requiring only 14 additions. IEEE Trans. Circ. Syst. I: Regular Pap. **61**(6), 1727–1740 (2014)
17. D. Puchala, K. Stokfiszewski, B. Szczepaniak, M. Yatsymirskyy, Effectiveness of Fast Fourier Transform Implementations on GPU and CPU. Electr. Rev. (Przeglad Elektrotechniczny). **R.92**(7), 69–71 (2016)
18. J. Huang, P. Schultheiss, Block quantization of correlated Gaussian random variables. IEEE Trans. Commun. Syst. **11**(3), 289–296 (1963). https://doi.org/10.1109/TCOM.1963.1088759
19. K. Rao, P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. (Academic Press, 1990)
20. K. R. Rao, J. J. Hwang, *Techniques and standards for image, video, and audio coding*. (Prentice Hall, 1996), p. 563
21. K. Sayood, *Introduction to Data Compression*. (Morgan Kaufmann, 2000)
22. International Organization for Standardization, Digital compression and coding of continuous-tone still images, ISO/IEC IS 10918-1 (1994)
23. International Organization for Standardization, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, ISO/IEC 11172-1. **1993** (1993)
24. International Telecomunication Union (ITU), Video coding for low bit rate communication, ITU-T Recommendation H.263 (2005)
25. International Telecomunication Union (ITU), Advanced video coding for generic audiovisual services, ITU-T Recommendation H.263 (2017)
26. M. Masoumi, H. Ahmadifar, in *IEEE 4th Int. Conference on Knowledge-Based Engineering and Innovation*, Performance of HEVC Discrete Cosine and Sine transforms on GPU using CUDA, (2017), pp. 857–861
27. M. Wawrzonowski, M. Daszuta, D. Szajerman, P. Napieralski, in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017*, Mobile devices' GPUs in cloth dynamics simulation, (2017), pp. 1283–1290
28. C. G. Kim, S. J. Lee, S. D. Kim, in *Pattern Recognition and Image Analysis. IbPRIA 2005*. ed. by J. S. Marques, N. Pérez de la Blanca, and P. Pina, 2-D Discrete Cosine Transform (DCT) on Meshes with Hierarchical Control Modes, vol. 3522, (2005)
29. C. S. Lubobya, M. E. Dlodlo, G. de Jager, K. L. Ferguson, in *Proc. of IEEE Africon Conference*, SIMD implementation of integer DCT and Hadamard transforms in H.264/AVC encoder, (2011)
30. W. Chen, C. H. Smith, S. C. Fralick, A fast computational algorithm for the discrete cosine transform. IEEE Trans. Commun. **COMM-25**, 1004–1009 (1977)

Publisher's Note