



Article

Low-Complexity Loeffler DCT Approximations for Image and Video Coding

Diego F. G. Coelho ¹, Renato J. Cintra ^{2,3,*}, Fábio M. Bayer ⁴, Sunera Kulasekera ⁵ ,
Arjuna Madanayake ⁶, Paulo Martinez ^{3,7}, Thiago L. T. Silveira ⁸ , Raíza S. Oliveira ⁹ and
Vassil S. Dimitrov ²

¹ Independent Researcher, Calgary AB, T3A 2H6, Canada; diegofgcoelho@gmail.com

² Department of Electrical and Computer Engineering, University of Calgary, 2500 University Dr. NW
Calgary, AB T2N 1N4, Canada; vdvds103@gmail.com

³ Signal Processing Group, Departamento de Estatística, Universidade Federal de Pernambuco,
Recife 50670-901, PE, Brazil; pacm@protonmail.com

⁴ Departamento de Estatística and LACESM, Universidade Federal de Santa Maria, Santa Maria 97105-900, RS,
Brazil; bayer@ufsm.br

⁵ Department of Electrical and Computer Engineering, University of Akron, Akron, OH 44325, USA;
sunera87@gmail.com

⁶ Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174,
USA; amadanay@fiu.edu

⁷ Friedrich-Alexander-Universität Erlangen-Nürnberg, 91054 Erlangen, Germany

⁸ Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul (UFRGS),
Porto Alegre 91501-970, RS, Brazil; thiago@inf.ufsm.br

⁹ Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Pernambuco,
Recife 50670-901, PE, Brazil; raiza.s.oliveira@gmail.com

* Correspondence: rjdsc@de.ufpe.br

Received: 9 October 2018; Accepted: 16 November 2018; Published: 22 November 2018



Abstract: This paper introduced a matrix parametrization method based on the Loeffler discrete cosine transform (DCT) algorithm. As a result, a new class of 8-point DCT approximations was proposed, capable of unifying the mathematical formalism of several 8-point DCT approximations archived in the literature. Pareto-efficient DCT approximations are obtained through multicriteria optimization, where computational complexity, proximity, and coding performance are considered. Efficient approximations and their scaled 16- and 32-point versions are embedded into image and video encoders, including a JPEG-like codec and H.264/AVC and H.265/HEVC standards. Results are compared to the unmodified standard codecs. Efficient approximations are mapped and implemented on a Xilinx VLX240T FPGA and evaluated for area, speed, and power consumption.

Keywords: discrete cosine transform; approximation; multicriteria optimization; image/video compression

1. Introduction

Discrete time transforms have a major role in signal-processing theory and application. In particular, tools such as the discrete Haar, Hadamard, and discrete Fourier transforms, and several discrete trigonometrical transforms [1,2] have contributed to various image-processing techniques [3–6]. Among such transformations, the discrete cosine transform (DCT) of type II is widely regarded as a pivotal tool for image compression, coding, and analysis [2,7,8]. This is because the DCT closely approximates the Karhunen–Loève transform (KLT) which can optimally decorrelate highly correlated stationary Markov-I signals [2].

Indeed, the recent literature reveals a significant number of works linked to DCT computation. Some noteworthy topics are: (i) cosine–sine decomposition to compute the 8-point DCT [9]; (ii) low-complexity-pruned 8-point DCT approximations for image encoding [10]; (iii) improved 8-point approximate DCT for image and video compression requiring only 14 additions [11]; (iv) HEVC multisize DCT hardware with constant throughput, supporting heterogeneous coding unities [12]; (v) approximation of feature pyramids in the DCT domain and its application to pedestrian detection [13]; (vi) performance analysis of DCT and discrete wavelet transform (DWT) audio watermarking based on singular value decomposition [14]; (vii) adaptive approximated DCT architectures for HEVC [15]; (viii) improved Canny edge detection algorithm based on DCT [16]; and (ix) DCT-inspired feature transform for image retrieval and reconstruction [17]. In fact, several current image- and video-coding schemes are based on the DCT [18], such as JPEG [19], MPEG-1 [20], H.264 [21], and HEVC [22]. In particular, the H.264 and HEVC codecs employ low-complexity discrete transforms based on the 8-point DCT. The 8-point DCT has also been applied to dedicated image-compression systems implemented in large web servers with promising results [23]. As a consequence, several algorithms for the 8-point DCT have been proposed, such as: Lee DCT factorization [24], Arai DCT scheme [25], Feig–Winograd algorithm [26], and the Loeffler DCT algorithm [27]. Among these methods, the Loeffler DCT algorithm [27] has the distinction of achieving the theoretical lower bound for DCT multiplicative complexity [28,29].

Because the computational complexity lower bounds of the DCT have been achieved [28], the research community resorted to approximation techniques to further reduce the cost of DCT calculation. Although not capable of providing exact computation, approximate transforms can furnish very close computational results at significantly smaller computational cost. Early approximations for the DCT were introduced by Haweel [30]. Since then, several DCT approximations have been proposed. In Reference [31], Lengwehasatit and Ortega introduced a scalable approximate DCT that can be regarded as a benchmark approximation [5,31–40]. Aiming at image coding for data compression, a series of approximations have been proposed by Bouguezal-Ahmad-Swamy (BAS) [5,34–37,41]. Such approximations offer very low complexity and good coding performance [2,39].

The methods for deriving DCT approximation include: (i) application of simple functions, such as signum, rounding-off, truncation, ceil, and floor, to approximate the elements of the exact DCT matrix [30,39]; (ii) scaling and rounding-off [2,32,39,42–47]; (iii) brute-force computation over reduced search space [38,39]; (iv) inspection [33–35,41]; (v) single-variable matrix parametrization of existing approximations [37]; (vi) pruning techniques [48]; and (vii) derivations based on other low-complexity matrices [5]. The above-mentioned methods are capable of supplying single or very few approximations. In fact, a systematized approach for obtaining a large number of matrix approximations and a unifying scheme is lacking.

The goal of this paper is two-fold. First we aim at unifying the matrix formalism of several 8-point DCT approximations archived in the literature. For that, we consider the 8-point Loeffler algorithm as a general structure equipped with a parametrization of the multiplicands. This approach allows the definition of a matrix subspace where a large number of approximations could be derived. Second, we propose an optimization problem over the introduced matrix subspace in order to discriminate the best approximations according to several well-known figures of merit. This discrimination is important from the application point of view. It allows the user to select the transform that fits best to their application in terms of balancing performance and complexity. The optimally found approximations are subject to mathematical assessment and embedding into image- and video-encoding schemes, including the H.264/AVC and the H.265/HEVC standards. Third, we introduce hardware architecture based on optimally found approximations realized in field programmable gate array (FPGA). Although there are several subsystems in a video and image codec, this work is solely concentrated on the discrete transform subsystem.

The paper unfolds as follows. Section 2 introduces a novel DCT parametrization based on the Loeffler DCT algorithm. We provide the mathematical background and matrix properties, such as

invertibility, orthogonality, and orthonormalization, are examined. Section 3 reviews the criteria employed for identifying and assessing DCT approximations, such as proximity and coding measures, and computational complexity. In Section 4, we propose a multicriteria-optimization problem aiming at deriving optimal approximation subject to Pareto efficiency. Obtained transforms are sought to be comprehensively assessed and compared with state-of-the-art competitors. Section 5 reports the results of embedding the obtained transforms into a JPEG-like encoder, as well as in H.264/AVC and H.265/HEVC video standards. In Section 6, an FPGA hardware implementation of the optimal transformations is detailed, and the usual FPGA implementation metrics are reported. Section 7 presents our final remarks.

2. DCT Parametrization and Matrix Space

2.1. DCT Matrix Factorization

The type II DCT is defined according to the following linear transformation matrix [2,8]:

$$\mathbf{C}_{\text{DCT}} = \frac{1}{2} \begin{bmatrix} c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{bmatrix}, \quad (1)$$

where $c_k = \cos(k\pi/16)$, $k = 1, 2, \dots, 7$. Because several entries of \mathbf{C}_{DCT} are not rational, they are often truncated/rounded and represented in floating-point arithmetic [7,49], which requires demanding computational costs when compared with fixed-point schemes [5,7,33,50].

Fast algorithms can minimize the number of arithmetic operations required for the DCT computation [2,7]. A number of fast algorithms have been proposed for the 8-point DCT [24–26]. The vast majority of DCT algorithms consist of the following factorization [2]:

$$\mathbf{C}_{\text{DCT}} = \mathbf{P} \cdot \mathbf{M} \cdot \mathbf{A}, \quad (2)$$

where \mathbf{A} is the additive matrix that represents a set of butterfly operations, \mathbf{M} is a multiplicative matrix, and \mathbf{P} is a permutation matrix that simply rearranges the output components to natural order. Matrix \mathbf{A} is often fixed [2] and given by:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad (3)$$

Multiplicative matrix \mathbf{M} can be further factorized. Since matrix factorization is not unique, each fast algorithm is linked to a particular factorization of \mathbf{M} . Finally, the permutation matrix is given below:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

Among the DCT fast algorithms, the Loeffler DCT achieves the theoretical lower bound of the multiplicative complexity for 8-point DCT, which consists of 11 multiplications [27]. In this work, multiplications by irrational quantities are sought to be substituted with trivial multipliers, representable by simple bit-shifting operations (Sections 2.2 and 3.2). Thus, we expect that

approximations based on Loeffler DCT could generate low-complexity approximations. Therefore, the Loeffler DCT algorithm was separated as the starting point to devise new DCT approximations.

The Loeffler DCT employs a scaled DCT with the following transformation matrix:

$$\mathbf{C}_{\text{Loeffler-DCT}} = 2\sqrt{2} \cdot \mathbf{C}_{\text{DCT}}. \quad (5)$$

Such scaling eliminates one multiplicand because $2\sqrt{2} \cdot c_4 = 1$. Therefore, we can write the following expression:

$$\mathbf{C}_{\text{Loeffler-DCT}} = \mathbf{P} \cdot \mathbf{M}' \cdot \mathbf{A}, \quad (6)$$

where

$$\begin{aligned} \mathbf{M}' &= 2\sqrt{2} \cdot \mathbf{M} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \sqrt{2}c_2 & \sqrt{2}c_6 & -\sqrt{2}c_6 & -\sqrt{2}c_2 & 0 & 0 & 0 & 0 \\ \sqrt{2}c_6 & -\sqrt{2}c_2 & \sqrt{2}c_2 & -\sqrt{2}c_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{2}c_1 & \sqrt{2}c_3 & -\sqrt{2}c_5 & \sqrt{2}c_7 \\ 0 & 0 & 0 & 0 & -\sqrt{2}c_5 & -\sqrt{2}c_1 & -\sqrt{2}c_7 & \sqrt{2}c_3 \\ 0 & 0 & 0 & 0 & \sqrt{2}c_3 & \sqrt{2}c_7 & -\sqrt{2}c_1 & \sqrt{2}c_5 \\ 0 & 0 & 0 & 0 & \sqrt{2}c_7 & \sqrt{2}c_5 & \sqrt{2}c_3 & \sqrt{2}c_1 \end{bmatrix}. \end{aligned} \quad (7)$$

Matrix \mathbf{M}' carries all multiplications by irrational quantities required by Loeffler fast algorithm. It can be further decomposed as:

$$\mathbf{M}' = \mathbf{B} \cdot \mathbf{C} \cdot \mathbf{D}, \quad (8)$$

where

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & 0 & 0 & c_5 \\ 0 & 0 & 0 & 0 & 0 & c_1 & c_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c_7 & c_1 & 0 \\ 0 & 0 & 0 & 0 & -c_5 & 0 & 0 & c_3 \end{bmatrix}, \quad (9)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}c_6 & \sqrt{2}c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{2}c_2 & \sqrt{2}c_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad (10)$$

and

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

In order to achieve the minimum multiplicative complexity, the Loeffler fast algorithm uses fast rotation for the rotation blocks in matrix \mathbf{B} and \mathbf{C} [2,50]. Since each of the three fast rotations requires three multiplications, and we have the two additional multiplications on matrix \mathbf{D} , the Loeffler fast algorithm requires a total of 11 multiplications.

2.2. Loeffler DCT Parametrization

DCT factorization suggests matrix parametrization. In fact, replacing multiplicands $\sqrt{2} \cdot c_i$, $i \in \{1, 2, 3, 5, 6, 7\}$ in Matrix (8) by parameters α_k , $k = 1, 2, \dots, 6$, respectively, yields the following parametric matrix:

$$\mathbf{M}_{\alpha} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \alpha_2 & \alpha_5 & -\alpha_5 & -\alpha_2 & 0 & 0 & 0 & 0 \\ \alpha_5 & -\alpha_2 & \alpha_2 & -\alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_1 & \alpha_3 & -\alpha_4 & \alpha_6 \\ 0 & 0 & 0 & 0 & -\alpha_4 & -\alpha_1 & -\alpha_6 & \alpha_3 \\ 0 & 0 & 0 & 0 & \alpha_3 & \alpha_6 & -\alpha_1 & \alpha_4 \\ 0 & 0 & 0 & 0 & \alpha_6 & \alpha_4 & \alpha_3 & \alpha_1 \end{bmatrix}, \quad (12)$$

where subscript $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_6]^\top$ denotes a real-valued parameter vector. Mathematically, the following mapping is introduced:

$$\begin{aligned} f: \mathbb{R}^6 &\longrightarrow \mathcal{M}(8) \\ \alpha &\longmapsto \mathbf{T}_{\alpha} = \mathbf{P} \cdot \mathbf{M}_{\alpha} \cdot \mathbf{A}, \end{aligned} \quad (13)$$

where $\mathcal{M}(8)$ represents the space of 8×8 matrices over the real numbers [51]. Mapping $f(\cdot)$ results in image set $\mathcal{C}(8) \subset \mathcal{M}(8)$ [51] that contains 8×8 matrices with the DCT matrix symmetry. In particular, for $\alpha_0 = \sqrt{2} \cdot [c_1 \ c_2 \ c_3 \ c_5 \ c_6 \ c_7]^\top$, we have that $f(\alpha_0) = \mathbf{C}_{\text{Loeffler-DCT}}$. Other examples are $\alpha_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^\top$ and $\alpha_2 = 1/2 \cdot [1 \ 2 \ 1 \ 1 \ 1 \ 1 \ 2]^\top$, which result in the following matrices:

$$\mathbf{T}_{\alpha_1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{T}_{\alpha_2} = \frac{1}{2} \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & -2 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & -2 & -1 & -1 & 1 & 1 & 2 & -1 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 1 & -1 & 2 & 1 & -1 & -2 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 2 & -1 & 1 & -1 & 1 & -1 & 1 & -2 \end{bmatrix}, \quad (14)$$

respectively. Although the above matrices have low complexity, they may not necessarily lead to a good transform matrix in terms of mathematical properties and coding capability.

Hereafter, we adopt the following notation:

$$\mathbf{M}_{\alpha} = \begin{bmatrix} \mathbf{E}_{\alpha} & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{O}_{\alpha} \end{bmatrix}, \quad (15)$$

where $\mathbf{0}_4$ is the the 4×4 null matrix, and

$$\mathbf{E}_{\alpha} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ \alpha_2 & \alpha_5 & -\alpha_5 & -\alpha_2 \\ \alpha_5 & -\alpha_2 & \alpha_2 & -\alpha_5 \end{bmatrix} \quad \text{and} \quad \mathbf{O}_{\alpha} = \begin{bmatrix} -\alpha_1 & \alpha_3 & -\alpha_4 & \alpha_6 \\ -\alpha_4 & -\alpha_1 & -\alpha_6 & \alpha_3 \\ \alpha_3 & \alpha_6 & -\alpha_1 & \alpha_4 \\ \alpha_6 & \alpha_4 & \alpha_3 & \alpha_1 \end{bmatrix}. \quad (16)$$

Submatrices \mathbf{E}_{α} and \mathbf{O}_{α} compute the even and odd index DCT components, respectively.

2.3. Matrix Inversion

The inverse of \mathbf{T}_{α} is directly given by:

$$\mathbf{T}_{\alpha}^{-1} = \mathbf{A}^{-1} \cdot \mathbf{M}_{\alpha}^{-1} \cdot \mathbf{P}^{-1}. \quad (17)$$

Because $\mathbf{A}^{-1} = \frac{1}{2}\mathbf{A}^\top = \frac{1}{2}\mathbf{A}$ and $\mathbf{P}^{-1} = \mathbf{P}^\top$ are well-defined, nonsingular matrices, we need only check the invertibility of \mathbf{M}_α [52]. By means of symbolic computation, we obtain:

$$\mathbf{M}_\alpha^{-1} = \frac{1}{4} \cdot (\mathbf{M}_{\alpha'}^\top), \quad (18)$$

where $\alpha' = [\alpha'_1 \ \alpha'_2 \ \alpha'_3 \ \alpha'_4 \ \alpha'_5 \ \alpha'_6]^\top$ with coefficients equals to:

$$\begin{aligned} \alpha'_1 &= -4(\alpha_1^3 + 2\alpha_1\alpha_3\alpha_4 + \alpha_1\alpha_6^2 + \alpha_3^2\alpha_6 - \alpha_4^2\alpha_6) / \det(\mathbf{O}_\alpha), \\ \alpha'_2 &= -16\alpha_2 / \det(\mathbf{E}_\alpha), \\ \alpha'_3 &= 4(-\alpha_1^2\alpha_4 - 2\alpha_1\alpha_3\alpha_6 - \alpha_3^3 - \alpha_3\alpha_4^2 + \alpha_4\alpha_6^2) / \det(\mathbf{O}_\alpha), \\ \alpha'_4 &= -4(\alpha_1^2\alpha_3 - 2\alpha_1\alpha_4\alpha_6 + \alpha_3^2\alpha_4 - \alpha_3\alpha_6^2 + \alpha_4^3) / \det(\mathbf{O}_\alpha), \\ \alpha'_5 &= -16\alpha_5 / \det(\mathbf{E}_\alpha), \\ \alpha'_6 &= 4(-\alpha_1^2\alpha_6 - \alpha_1\alpha_3^2 + \alpha_1\alpha_4^2 + 2\alpha_3\alpha_4\alpha_6 - \alpha_6^3) / \det(\mathbf{O}_\alpha), \end{aligned} \quad (19)$$

where $\det(\cdot)$ returns the determinant.

Note that the expression in Matrix (18) implies that the inverse of the matrix \mathbf{M}_α is a matrix with the same structure, whose coefficients are a function of the parameter vector α . For the matrix inversion to be well-defined, we must have: $\det(\mathbf{E}_\alpha) \cdot \det(\mathbf{O}_\alpha) \neq 0$. By explicitly computing $\det(\mathbf{E}_\alpha)$ and $\det(\mathbf{O}_\alpha)$, we obtain the following condition for matrix inversion:

$$\begin{cases} \alpha_2^2 + \alpha_5^2 \neq 0, \\ \frac{(\alpha_1^2 + \alpha_6^2)^2 - 4\alpha_3\alpha_4(\alpha_6^2 - \alpha_1^2)}{(\alpha_3^2 + \alpha_4^2)^2 - 4\alpha_1\alpha_6(\alpha_4^2 - \alpha_3^2)} \neq -1. \end{cases} \quad (20)$$

2.4. Orthogonality

In this paper, we adopt the following definitions. A matrix \mathbf{T} is orthonormal if $\mathbf{T} \cdot \mathbf{T}^\top$ is an identity matrix [53]. If product $\mathbf{T} \cdot \mathbf{T}^\top$ is a diagonal matrix, \mathbf{T} is said to be orthogonal. For \mathbf{T}_α , we have that symbolic computation yields:

$$\mathbf{T}_\alpha \cdot \mathbf{T}_\alpha^\top = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2s_1 & 0 & -2d & 0 & 2d & 0 & 0 \\ 0 & 0 & 2s_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2d & 0 & 2s_1 & 0 & 0 & 0 & 2d \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 2d & 0 & 0 & 0 & 2s_1 & 0 & 2d \\ 0 & 0 & 0 & 0 & 0 & 0 & 2s_0 & 0 \\ 0 & 0 & 0 & 2d & 0 & 2d & 0 & 2s_1 \end{bmatrix}, \quad (21)$$

where $s_0 = 2(\alpha_2^2 + \alpha_5^2)$, $s_1 = \alpha_1^2 + \alpha_3^2 + \alpha_4^2 + \alpha_6^2$, and $d = \alpha_1(\alpha_4 - \alpha_3) + \alpha_6(\alpha_4 + \alpha_3)$. Thus, if $d = 0$, then the transform \mathbf{T}_α is orthogonal.

2.5. Near Orthogonality

Some important and well-known DCT approximations are nonorthogonal [30,34]. Nevertheless, such transformations are nearly orthogonal [39,54]. Let \mathbf{A} be a square matrix. Deviation from orthogonality can be quantified according to the deviation from diagonality [39] of $\mathbf{A} \cdot \mathbf{A}^\top$, which is given by the following expression:

$$\delta(\mathbf{A} \cdot \mathbf{A}^\top) = 1 - \frac{\|\text{diag}(\mathbf{A} \cdot \mathbf{A}^\top)\|_F^2}{\|\mathbf{A} \cdot \mathbf{A}^\top\|_F^2}, \quad (22)$$

where $\text{diag}(\cdot)$ returns a diagonal matrix with the diagonal elements of its argument and $\|\cdot\|_F$ denotes the Frobenius norm [2]. Therefore, considering \mathbf{T}_α , we obtain:

$$\delta(\mathbf{T}_\alpha \cdot \mathbf{T}_\alpha^\top) = 1 - \frac{1}{1 + \frac{32d^2}{128+8s_0^2+16s_1^2}}. \quad (23)$$

Nonorthogonal transforms have been recognized as useful tools. The signed DCT (SDCT) is a particularly relevant DCT approximation [30] and its deviation from orthogonality is 0.20. We adopt such deviation as a reference value to discriminate nearly orthogonal matrices. Thus, for \mathbf{T}_α , we obtain the following criterion for near orthogonality:

$$0 < d^2 \leq 1 + \frac{s_0^2}{16} + \frac{s_1^2}{8}. \quad (24)$$

2.6. Orthonormalization

Discrete transform approximations are often sought to be orthonormal. Orthogonal transformations can be orthonormalized as described in References [2,33,53]. Based on polar decomposition [55], orthonormal or nearly orthonormal matrix $\hat{\mathbf{C}}_\alpha$ linked to \mathbf{T}_α is furnished by:

$$\hat{\mathbf{C}}_\alpha = \begin{cases} \mathbf{S}_\alpha \cdot \mathbf{T}_\alpha, & \text{if } d = 0, \\ \tilde{\mathbf{S}}_\alpha \cdot \mathbf{T}_\alpha, & \text{if (24) holds true,} \end{cases} \quad (25)$$

where $\mathbf{S}_\alpha = \sqrt{(\mathbf{T}_\alpha \cdot \mathbf{T}_\alpha^\top)^{-1}}$, $\tilde{\mathbf{S}}_\alpha = \sqrt{\text{diag}(\mathbf{T}_\alpha \cdot \mathbf{T}_\alpha^\top)^{-1}}$, and $\sqrt{\cdot}$ is the matrix square root [2].

3. Assessment Criteria

In this section, we describe the selected figures of merit for assessing the performance and complexity of a given DCT approximation. We separated the following performance metrics: (i) total error energy [32,54]; (ii) mean square error (MSE) [2,49]; (iii) unified coding gain [2,54], and (iv) transform efficiency [2]. For computational complexity assessment, we adopted arithmetic operation counts as figures of merit.

3.1. Performance Metrics

3.1.1. Total Error Energy

Total error energy quantifies the error between matrices in a Euclidean distance way. This measure is given by References [32,54]:

$$\epsilon(\hat{\mathbf{C}}_\alpha) = \pi \cdot \|\mathbf{C}_{\text{Loeffler-DCT}} - \hat{\mathbf{C}}_\alpha\|_F^2. \quad (26)$$

3.1.2. Mean Square Error

The MSE of given matrix approximation $\hat{\mathbf{C}}_\alpha$ is furnished by:

$$\text{MSE}(\hat{\mathbf{C}}_\alpha) = \frac{1}{8} \cdot \text{trace}((\mathbf{C}_{\text{Loeffler-DCT}} - \hat{\mathbf{C}}_\alpha) \cdot \mathbf{R}_{xx} \cdot (\mathbf{C}_{\text{Loeffler-DCT}} - \hat{\mathbf{C}}_\alpha)^\top), \quad (27)$$

where \mathbf{R}_{xx} represents the autocorrelation matrix of a Markov I stationary process with correlation coefficient ρ , and $\text{trace}(\cdot)$ returns the sum of main diagonal elements of its matrix argument. The (i, j) -th entry of \mathbf{R}_{xx} is given by $\rho^{|i-j|}$, $i, j = 0, 1, \dots, 7$ [1,2]. The correlation coefficient is assumed as equal to 0.95, which is representative for natural images [2].

3.1.3. Unified Transform Coding Gain

Unified transform coding gain provides a measure to quantify the compression capabilities of a given matrix [54]. It is a generalization of usual transform coding gain as in Reference [2]. Let \mathbf{g}_k and \mathbf{h}_k be the k th row of $\hat{\mathbf{C}}_{\alpha}^{\top}$ and \mathbf{C}_{α} , respectively. Then, the unified transform coding gain is given by Reference [56]:

$$C_g(\hat{\mathbf{C}}_{\alpha}) = 10 \cdot \log_{10} \left[\prod_{k=1}^8 \frac{1}{(A_k \cdot B_k)^{\frac{1}{8}}} \right] \quad (\text{dB}), \quad (28)$$

where $A_k = \text{sum}[(\mathbf{h}_k \cdot \mathbf{h}_k^{\top}) \circ \mathbf{R}_{\text{xx}}]$, $\text{sum}(\cdot)$ returns the sum of elements of its matrix argument, operator \circ denotes the element-wise matrix product, $B_k = \|\mathbf{g}_k\|_2^2$, and $\|\cdot\|_2$ is the usual vector norm.

3.1.4. Transform Efficiency

Another measure for assessing coding performance is transform efficiency [2]. Let matrix $\mathbf{R}_{\text{XX}} = \hat{\mathbf{C}}_{\alpha} \cdot \mathbf{R}_{\text{xx}} \cdot \hat{\mathbf{C}}_{\alpha}^{\top}$ be the covariance matrix of transformed signal \mathbf{X} . The transform efficiency of $\hat{\mathbf{C}}_{\alpha}$ is given by [2]:

$$\eta(\hat{\mathbf{C}}_{\alpha}) = \frac{\text{trace}(|\mathbf{R}_{\text{XX}}|)}{\text{sum}(|\mathbf{R}_{\text{XX}}|)}. \quad (29)$$

3.2. Computational Cost

Based on Loeffler DCT factorization, a fast algorithm for \mathbf{T}_{α} is obtained and its signal flow graph (SFG) is shown in Figure 1. Stage 1, 2, and 3 correspond to matrices \mathbf{A} , \mathbf{M}_{α} , and \mathbf{P} , respectively. The computational cost of such an algorithm is closely linked to selected parameter values α_k , $k = 1, 2, \dots, 6$. Because we aim at proposing multiplierless approximations, we restricted parameter values α_k to set $\mathcal{P} = \{0, \pm 1/2, \pm 1, \pm 2\}$, i.e., $\alpha \in \mathcal{P}^6$. The elements in \mathcal{P} correspond to trivial multiplications that affect null multiplicative complexity. In fact, the elements in \mathcal{P} represent only additions and minimal bit-shifting operations.

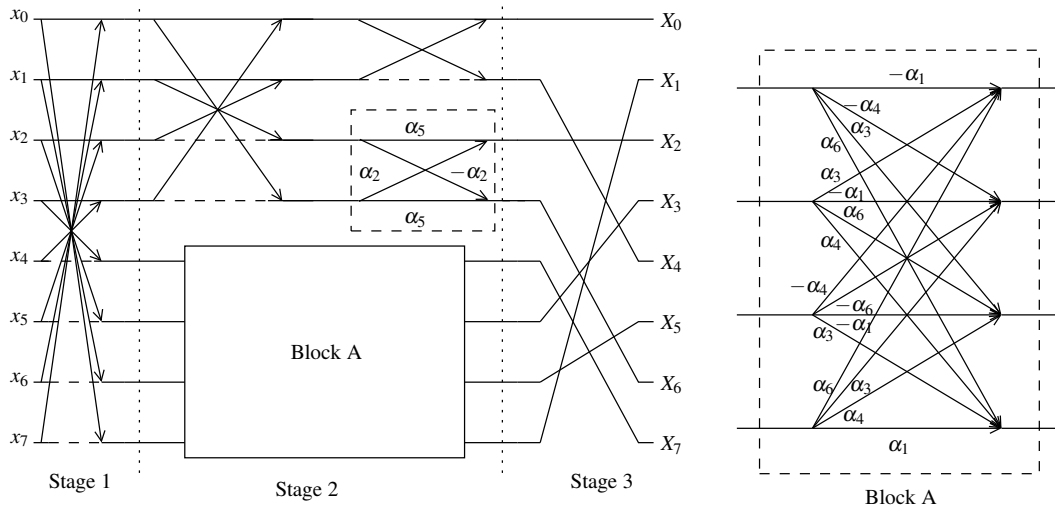


Figure 1. Signal flow graph (SFG) of the Loeffler-based transformations. Dashed lines represents multiplication by -1 . Stages 1, 2, and 3 are represented by matrices \mathbf{A} , \mathbf{M}' , and \mathbf{P} , respectively, as in Matrix (6).

The number of additions and bit-shifts can be evaluated by inspecting the discussed algorithm (Figure 1). Thus, we obtain the following expressions for the addition and bit-shifting counts, respectively:

$$A(\alpha) = 8 + 2 \cdot \max \left\{ 1, \mathbf{1}_{\mathcal{P}-\{0\}}(\alpha_2) + \mathbf{1}_{\mathcal{P}-\{0\}}(\alpha_5) \right\} + 4 \cdot \max \left\{ 1, \sum_{k \in \{1,3,4,6\}} \mathbf{1}_{\mathcal{P}-\{0\}}(\alpha_k) \right\}, \quad (30)$$

$$S(\alpha) = 2 \cdot [\mathbf{1}_{\{\pm\frac{1}{2}, \pm 2\}}(\alpha_2) + \mathbf{1}_{\{\pm\frac{1}{2}, \pm 2\}}(\alpha_5)] + 4 \cdot \sum_{k \in \{1,3,4,6\}} \mathbf{1}_{\{\pm\frac{1}{2}, \pm 2\}}(\alpha_k), \quad (31)$$

where $\mathbf{1}_A(x) = 1$, if $x \in A$, and 0 otherwise.

4. Multicriteria Optimization and New Transforms

In this section, we introduce an optimization problem that aims at identifying optimal transformations derived from the proposed mapping (Matrix (13)). Considering the various performances and complexity metrics discussed in the previous section, we set up the following multicriteria optimization problem [57,58]:

$$\min_{\alpha \in \mathcal{P}^6} (\epsilon(\hat{\mathbf{C}}_\alpha), \text{MSE}(\hat{\mathbf{C}}_\alpha), -C_g(\hat{\mathbf{C}}_\alpha), -\eta(\hat{\mathbf{C}}_\alpha), A(\alpha), S(\alpha)), \quad (32)$$

subject to:

- i the existence of inverse transformation, according to the condition established in Matrix (20);
- ii the entries of the inverse matrix must be in \mathcal{P} ; to ensure both forwarded and inverse low-complexity transformations;
- iii the property of orthogonality or near-orthogonality according to the criterion in Equation (24).

Quantities $C_g(\hat{\mathbf{C}}_\alpha)$ and $\eta(\hat{\mathbf{C}}_\alpha)$ are in negative form to comply to the minimization requirement.

Being a multicriteria optimization problem, Problem (32) is based on objective function set $\mathcal{F} = \{\epsilon(\cdot), \text{MSE}(\cdot), -C_g(\cdot), -\eta(\cdot), A(\cdot), S(\cdot)\}$. The problem in analysis is discrete and finite since there is a countable number of values to the objective function. However, the nonlinear, discrete nature of the problem renders it unsuitable for analytical methods. Therefore, we employed exhaustive search methods to solve it. The discussed multicriteria problem requires the identification of the Pareto efficient solutions set [57], which is given by:

$$\{\alpha^* \in \mathcal{P}^6 : \text{there is no } \alpha \in \mathcal{P}^6, \text{ such that} \\ f(\alpha) \leq f(\alpha^*) \text{ for all } f \in \mathcal{F} \text{ and} \\ f_0(\alpha) < f_0(\alpha^*) \text{ for some } f_0 \in \mathcal{F}\}. \quad (33)$$

4.1. Efficient Solutions

The exhaustive search [57] returned six efficient parameter vectors, which are listed in Table 1. For ease of notation, we denote the low-complexity matrices and their associated approximations linked to efficient solutions according to: $\mathbf{T}_i \triangleq \mathbf{T}_{\alpha_i^*}$ and $\hat{\mathbf{C}}_i \triangleq \hat{\mathbf{C}}_{\alpha_i^*}$, respectively. Table 2 summarizes the performance metrics, arithmetic complexity, and orthonormality property of obtained matrices $\hat{\mathbf{C}}_i$, $i = 1, 2, \dots, 6$. We included the DCT for reference as well. Note that all DCT approximations except those by $\hat{\mathbf{C}}_3$ are orthonormal.

Table 1. Efficient solutions.

i	α_i^*
1	$[1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$
2	$[1 \ 1 \ 0 \ 0 \ \frac{1}{2} \ 0]^T$
3	$[1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$
4	$[1 \ 1 \ 1 \ 1 \ \frac{1}{2} \ 0]^T$
5	$[1 \ 2 \ 0 \ 0 \ 1 \ 0]^T$
6	$[1 \ 2 \ 1 \ 1 \ 1 \ 0]^T$

Table 2. Efficient Loeffler-based discrete cosine transform (DCT) approximations and the DCT.

Transform	ϵ	MSE	C_g	η	$A(\alpha)$	$S(\alpha)$	Orthonormal?	Description
\hat{C}_1	8.66	0.059	7.33	80.90	14	0	Yes	Proposed in Reference [33]
\hat{C}_2	7.73	0.056	7.54	81.99	16	2	Yes	Proposed as T_{10} in Reference [54]
\hat{C}_3	1.44	0.007	8.30	89.77	18	0	No	Proposed as \tilde{T}_1 in Reference [39]
\hat{C}_4	0.87	0.006	8.39	88.70	24	2	Yes	Proposed as \hat{D}_1 in Reference [31]
\hat{C}_5	7.73	0.056	7.54	81.99	16	2	Yes	Equivalent to \hat{C}_2 and proposed as T_4 in Reference [54]
\hat{C}_6	0.87	0.006	8.39	88.70	24	2	Yes	Equivalent to \hat{C}_4 and proposed as T_9 in Reference [54]
DCT	0	0	8.85	93.99	–	–	Yes	Exact DCT as described in Reference [2]

4.2. Comparison

Several DCT approximations are encompassed by the proposed matrix formalism. Such transformations include: the SDCT [30], the approximation based on round-off function proposed in Reference [38], and all the DCT approximations introduced in Reference [39]. For instance, the SDCT [30] is another particular transformation fully described by the proposed matrix mapping. In fact, the SDCT can be obtained by taking $f(\alpha_1)$, where $\alpha_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$. Nevertheless, none of these approximations is part of the Pareto efficient solution set induced by the discussed multicriteria optimization problem [57]. Therefore, we compare the obtained efficient solutions with a variety of state-of-the-art 8-point DCT approximations that cannot be described by the proposed Loeffler-based formalism. We separated the Walsh–Hadamard transform (WHT) and the Bouguezel–Ahmad–Swamy (BAS) series of approximations labeled BAS₁ [34], BAS₂ [35], BAS₃ [41], BAS₄ [36], BAS₅ [37] (for $a = 1$), BAS₆ [37] (for $a = 0$), BAS₇ [37] (for $a = 1/2$), and BAS₈ [5]. Table 3 shows the performance measures for these transforms. For completeness, we also show the unified coding gain and the transform efficiency measures for the exact DCT [2].

Some approximations, such as the SDCT, were not explicitly included in our comparisons. Although they are in the set of matrices generated by Loeffler parametrization, they are not in the efficient solution set. Thus, we removed them from further analyses for not being an optimal solution.

Table 3. Performance of the Bouguezel–Ahmad–Swamy (BAS) approximations, the Walsh–Hadamard transform (WHT), and the DCT.

Transform	Orthogonalizable	ϵ	MSE	C_g	η	$A(\alpha)$	$S(\alpha)$
BAS ₁ [34]	No	4.19	0.019	6.27	83.17	21	0
BAS ₂ [35]	Yes	5.93	0.024	8.12	86.86	18	2
BAS ₃ [41]	Yes	6.85	0.028	7.91	85.38	18	0
BAS ₄ [36]	Yes	4.09	0.021	8.33	88.22	24	4
BAS ₅ [37]	Yes	26.86	0.071	7.91	85.38	18	0
BAS ₆ [37]	Yes	26.86	0.071	7.91	85.64	16	0
BAS ₇ [37]	Yes	26.40	0.068	8.12	86.86	18	2
BAS ₈ [5]	Yes	35.06	0.102	7.95	85.31	24	0
WHT [2]	Yes	5.05	0.025	7.95	85.31	24	0
DCT [2]	Yes	0	0	8.85	93.99	–	–

In order to compare all the above-mentioned transformations, we aimed at identifying the Pareto frontiers [57] in two-dimensional plots considering the performance figures of the obtained efficient solution as well as the WHT and BAS approximations. Thus, we devised scatter plots considering the arithmetic complexity and performance measures. The resulting plots are shown in Figure 2. Orthogonal transform approximations are marked with circles, and nonorthogonal approximations with cross signs. The dashed curves represent the Pareto frontier [57] for each selected pair of the measures. Transformations located on the Pareto frontier are considered optimal, where the points are dominated by the frontier correspond to nonoptimal transformations. The bivariate plots in Figure 2a,b reveal that the obtained Loeffler-based DCT approximations are often situated at the optimality site prescribed by the Pareto frontier. The Loeffler approximations perform particularly well in terms of total error energy and the MSE, which capture the matrix proximity to the exact DCT matrix in a Euclidean sense. Such approximations are particularly suitable for problems that require computational proximity to the the exact transformation as in the case of detection and estimation problems [59,60]. Regarding coding performance, Figure 2c,d shows that transformations \hat{C}_1 , \hat{C}_3 , \hat{C}_6 , and BAS_6 are situated on the Pareto frontier, being optimal in this sense. These approximations are adequate for data compression and decorrelation [2].

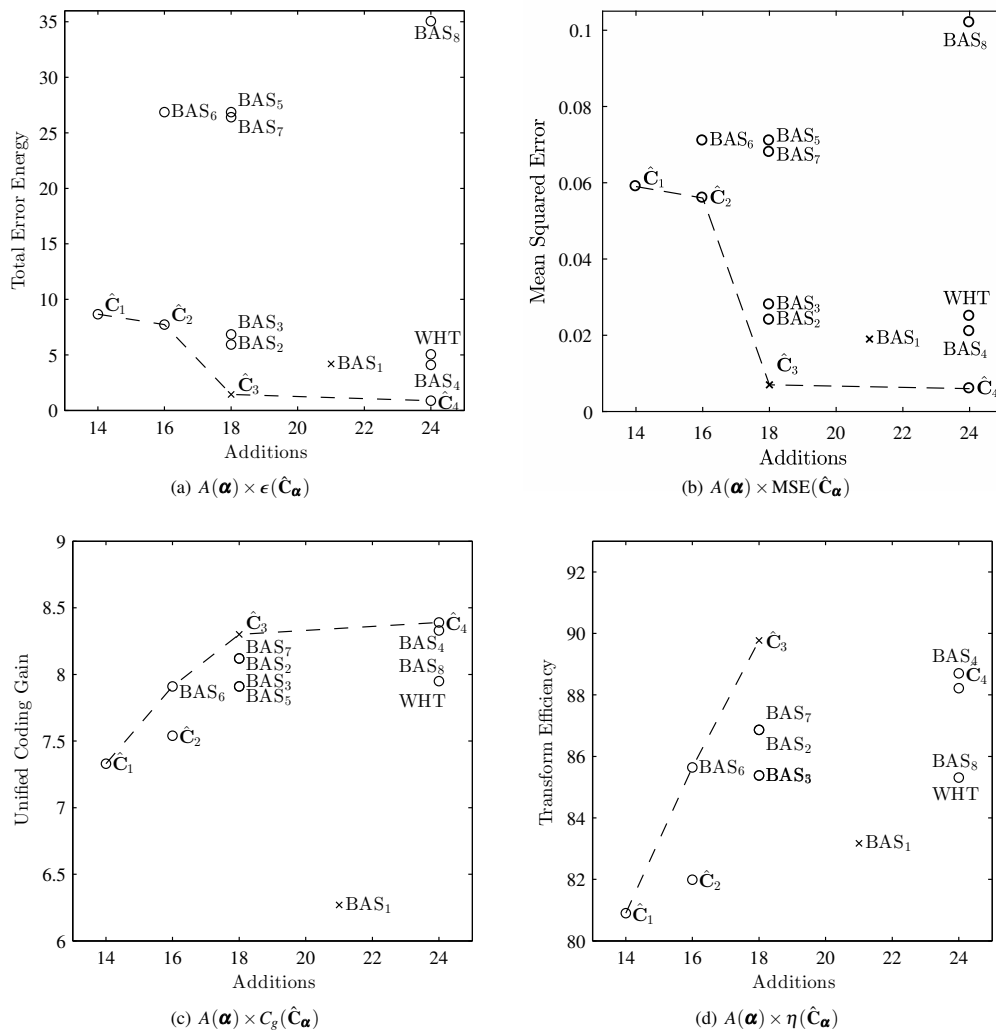


Figure 2. Performance plots and Pareto frontiers for the discussed transformations. Orthogonal transforms are marked with circles (o) and nonorthogonal approximations with cross sign (x). Dashed curves represent the Pareto frontier.

5. Image and Video Experiments

5.1. Image Compression

We implemented the JPEG-like compression experiment described in References [30,34,37,37,41] and submitted the standard Elaine image to processing at a high compression rate. For quantitative assessment, we adopted the structural similarity (SSIM) index [61] and the peak signal-to-noise rate (PSNR) [8,18] measure. Figure 3 shows the reconstructed Elaine image according to the JPEG-like compression considering the following transformations: DCT, \hat{C}_1 , \hat{C}_2 , \hat{C}_3 , \hat{C}_4 , and BAS_6 . We employed fixed-rate compression and retained only five coefficients, which led to 92.1875% compression rate. Despite very low computational complexity, the approximations could furnish images with quality comparable to the results obtained from the exact DCT. In particular, approximations \hat{C}_1 and \hat{C}_2 offered good trade-off, since they required only 14–16 additions and were capable of providing competitive image quality at smaller hardware and power requirements (Section 6).

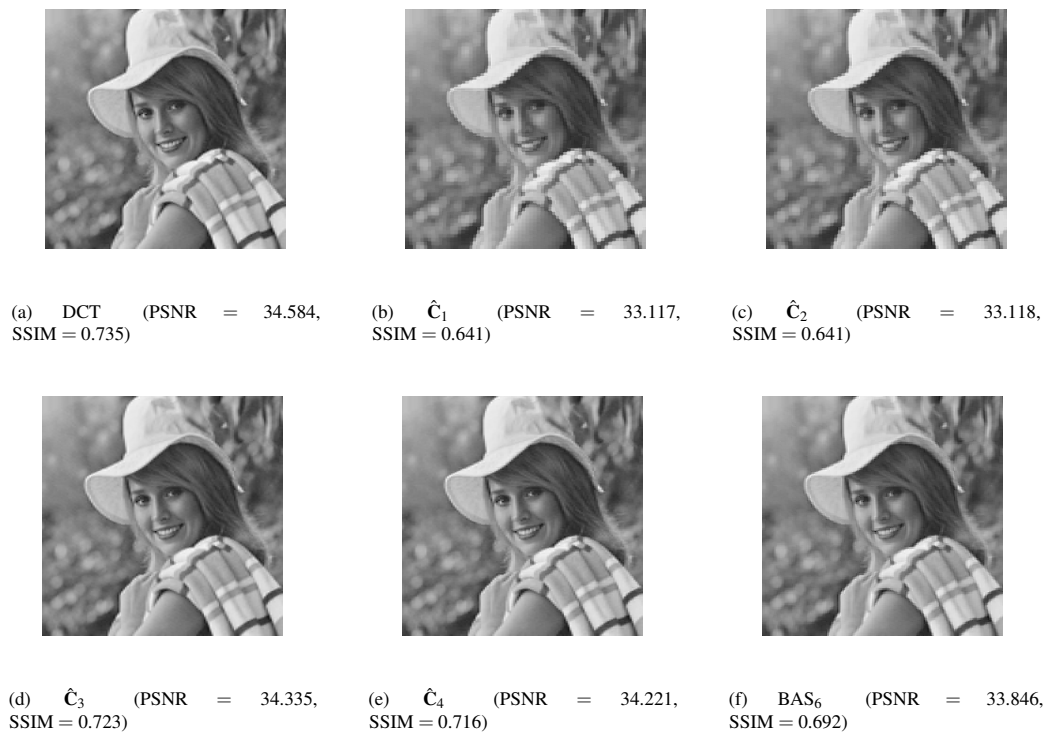


Figure 3. Elaine image compressed for DCT and selected approximations: \hat{C}_1 , \hat{C}_2 , \hat{C}_3 , \hat{C}_4 , and BAS_6 . Only five transform-domain coefficients were retained.

Figure 4 shows the PSNR and SSIM for different number of retained coefficients. Considering PSNR measurements, the difference between the measurements associated to the BAS_6 and to the efficient approximation were less than ≈ 1 dB. Similar behavior was reported when SSIM measurements were considered.

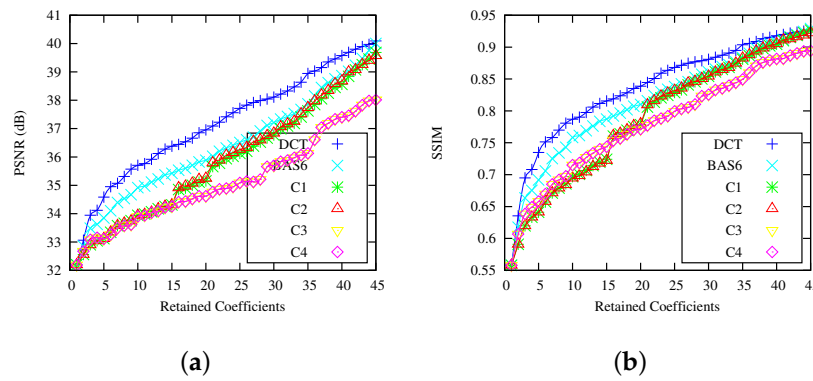


Figure 4. Peak signal-to-noise rate (PSNR) (a) and structural similarity (SSIM) (b) for the Elaine image compressed for DCT, and selected approximations \hat{C}_1 , \hat{C}_2 , \hat{C}_3 , \hat{C}_4 , and BAS_6 .

5.2. Video Compression

5.2.1. Experiments with the H.264/AVC Standard

To assess the Loeffler-based approximations in the context of video coding, we embedded them in the x264 [62] software library for encoding video streams into the H.264/AVC standard [63]. The original 8-point transform employed in H.264/AVC is a DCT-based integer approximation given by Reference [64]:

$$\hat{C}_{AVC} = \frac{1}{8} \cdot \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}. \quad (34)$$

The fast algorithm for the above transformation requires 32 additions and 14 bit-shifting operations [64].

We encoded 11 common intermediate format (CIF) videos with 300 frames from a public video database [65] using the standard and modified codec. In our simulation, we employed default settings and the resulting video quality was assessed by means of the average PSNR of chrominance and luminance representation considering all reconstructed frames. Psychovisual optimization was also disabled in order to obtain valid PSNR values and the quantization step was unaltered.

We computed the Bjøntegaard delta rate (BD-Rate) and the Bjøntegaard delta PSNR (BD-PSNR) [66] for modified codec compared to the usual H.264 native DCT approximation. BD-Rate and BD-PSNR were automatically furnished by the x264 [62] software library. A comprehensive review of Bjøntegaard metrics and their specific mathematical formulations can be found in Reference [67]. For that, we followed the procedure specified in References [66–68]. We adopted the same testing point as determined in Reference [69] with fixed quantization parameter (QP) in $\{22, 27, 32, 37\}$. Following References [67,68], we employed cubic spline interpolation between the testing points for better visualization.

Figure 5 shows the resulting average BD-Rate distortion curves for the selected videos, where H.264 represents the integer DCT used in the x264 software library shown in Matrix (34). We note the superior performance of \hat{C}_3 and \hat{C}_4 approximations compared to BAS_2 and BAS_6 transforms. This performance is more evident for small bitrates. As the bitrate increases, the quality performance of \hat{C}_3 , \hat{C}_4 , BAS_2 , and BAS_6 approximates native DCT implementation. Table 4 shows the BD-Rate and BD-PSNR measures for the 11 CIF video sequence selected from Reference [65].

Figure 6a–f displays the first encoded frame of the ‘Foreman’ video sequence at QP = 27 according to the modified coded based on the proposed efficient approximations and BAS approximations. For reference, Figure 6g shows the resulting frame obtained from the standard H.264.

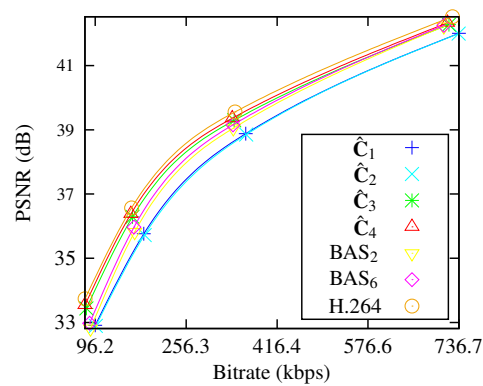


Figure 5. Average rate-distortion curves of the modified H.264/AVC software for the selected 11 CIF videos from Reference [65] for the PSNR (dB) in terms of bitrate.



Figure 6. First frame of the H.264/AVC compressed ‘Foreman’ sequence with QP = 27.

Table 4. Average Bjøntegaard delta (BD)-Rate and BD-PSNR for the 11 common intermediate format (CIF) videos selected from Reference [65].

Transform	BD-Rate (%)	BD-PSNR (dB)
\hat{C}_1	29.98	−1.08
\hat{C}_2	31.40	−1.13
\hat{C}_3	7.48	−0.29
\hat{C}_4	3.94	−0.15
BAS ₂	17.43	−0.68
BAS ₆	13.90	−0.54

Note that Loeffler approximations \hat{C}_1 and \hat{C}_2 , again, show better performance in terms of BD-PSNR measures than BAS transforms BAS_2 and BAS_6 . We attribute the difference in quality between \hat{C}_3 and \hat{C}_4 in relation to \hat{C}_1 and \hat{C}_2 to the associated complexity discrepancy as depicted in Table 2. The reduction in the number of additions and bit-shifting operations come at the expense of performance.

5.2.2. Experiments with the H.265/HEVC Standard

We also embedded Loeffler-based approximations into the HM-16.3 H.265/HEVC reference software [70]. The H.265/HEVC employs the following integer transform:

$$\hat{C}_{HEVC} = \frac{1}{64} \cdot \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix}, \quad (35)$$

which demands 72 additions and 62 bit-shifting operations, when considering the factorization method based on odd and even decomposition described in Reference [2]. Besides a particular 8-point integer transformation, the HEVC standard also employs 4-, 16-, and 32-point transforms [22]. To obtain a modified version of the H.265/HEVC codec, we derived 16- and 32-point approximations based on the scalable recursive method proposed in Reference [71]. The 8-point Loeffler-based efficient solutions were employed as fundamental building blocks for scaling up. Since the 4-point DCT matrix only possesses integer coefficients, it was employed unaltered. The additive complexity of the scaled 16- and 32-point approximations is $2 \cdot A(\alpha) + 16$ and $4 \cdot A(\alpha) + 64$, respectively [71]. For the bit-shifting count, we have $2 \cdot S(\alpha)$ and $4 \cdot S(\alpha)$, respectively.

In order to perform image-quality assessment, we encoded the first 100 frames of one standard sequence of each A to F video classes defined in the Common Test Conditions and Software Reference Configurations (CTCSRC) document [72]. The selected videos and their attributes are made explicit in the first two columns of Table 5. Classes A and B present high-resolution content for broadcasting and video on demand services; Classes C and D are lower-resolution contents for the same applications; Class E reflects typical examples of video-conferencing data; and Class F provides computer-generated image sequences [73]. All encoding parameters, including QP values, were set according to the CTCSRC document for the Main profile and All-Intra (AI), Random Access (RA), Low Delay B (LD-B), and Low Delay P (LD-P) configurations. In the AI configuration, each frame from the input video sequence is independently coded. RA and LD (B and P) coding modes differ mainly by order of coding and outputting. In the former, the coding order and output order of the pictures (frames) may differ, whereas equal order is required for the latter (Reference [74], p. 94). RA configuration relates to broadcasting and streaming use, whereas LD (B and P) represent conventional applications.

It is worth mentioning that Class A test sequences are not used for LDB and LDP coding tests, whereas Class E videos are not included in the RA test cases because of the nature of the content they represent (Reference [74], p. 93).

As figures of merit, we computed the BD-Rate and BD-PSNR for the modified versions of the codec. Figures 7–10 depict the rate-distortion curves and Table 5 presents the Bjøntegaard metrics for the considered sets of 8- to 32-point approximations for modes AI, RA, LDB, and LDP. BD-Rate and BD-PSNR are also automatically furnished by HM-16.3 H.265/HEVC reference software [70]. We employed the cubic spline interpolation based on four testing points on Figures 7–10, as specified in References [67,68]. The curves denoted by HEVC represents the native integer approximations for the 8-, 16-, and 32-point DCT in the HM-16.3 H.265/HEVC software [70]. These four testing points were determined by the specification for HEVC as in Reference [69] with fixed $QP \in \{22, 27, 32, 37\}$. One can note that there is no significant image-quality degradation when considering approximate

transforms. Furthermore, in all the cases, it can be seen that \hat{C}_4 performs better with a degradation of no more than 0.52 dB.

The Loeffler-based approximations could present competitive image quality while possessing low computational complexity. Table 6 shows the arithmetic costs of both the original and modified extended versions, and the H.265/HEVC native transform. As a qualitative result, Figure 11 depicts the first frame of the ‘BasketballDrive’ video sequence resulting from the unmodified H.265/HEVC [70] and from the modified codec according to the discussed approximations and their scaled versions at QP = 32. We also display the related frames according to the BAS₂ and BAS₆ transforms.

Table 5. Bjøntegaard metrics of the modified HEVC reference software for tested video sequences.

Class	Name	Resolution and Frame Rate	BD-PSNR (dB) and BD-Rate (%)					
			\hat{C}_1	\hat{C}_2	\hat{C}_3	\hat{C}_4	BAS ₂	BAS ₆
A	‘PeopleOnStreet’	2560 × 1600 30 fps	0.54 dB −9.75%	0.53 dB −9.61%	0.53 dB −9.49%	0.34 dB −6.30%	0.47 dB −8.47%	0.50 dB −8.99%
B	‘BasketballDrive’	1920 × 1080 50 fps	0.33 dB −11.56%	0.32 dB −11.28%	0.45 dB −15.52%	0.20 dB −7.24%	0.25 dB −8.90%	0.27 dB −9.50%
C	‘RaceHorses’	832 × 480 30 fps	0.67 dB −8.06%	0.66 dB −7.98%	0.83 −9.70%	0.52 dB −6.36%	0.61 dB −7.38%	0.61 dB −7.38%
D	‘BlowingBubbles’	416 × 240 50 fps	0.26 dB −4.35%	0.25 dB −4.25%	0.26 dB −4.46%	0.15 dB −2.59%	0.20 dB −3.48%	0.22 dB −3.74%
E	‘KristenAndSara’	1280 × 720 60 fps	0.47 dB −8.97%	0.46 dB −8.78%	0.47 dB −9.00%	0.30 dB −5.93%	0.39 dB −7.49%	0.41 dB −7.90%
F	‘BasketbalDrillText’	832 × 480 50 fps	0.20 dB −3.82%	0.20 dB −3.79%	0.19 dB −3.63%	0.12 dB −2.26%	0.16 dB −3.06%	0.17 dB −3.28%

Table 6. Computational complexity for N -point transforms employed in the H.265/ High Efficiency Video Coding (HEVC) experiments.

N	Transform	Add.	Mult.	Shifts
8	H.265/HEVC [75]	28	22	0
	T ₁	14	0	0
	T ₃	18	0	0
	T ₅	16	0	2
	T ₆	24	0	2
16	H.265/HEVC [75]	100	86	0
	T ₁	44	0	0
	T ₃	52	0	0
	T ₅	48	0	4
	T ₆	64	0	4
32	H.265/HEVC [75]	372	342	0
	T ₁	120	0	0
	T ₃	136	0	0
	T ₅	128	0	8
	T ₆	160	0	8

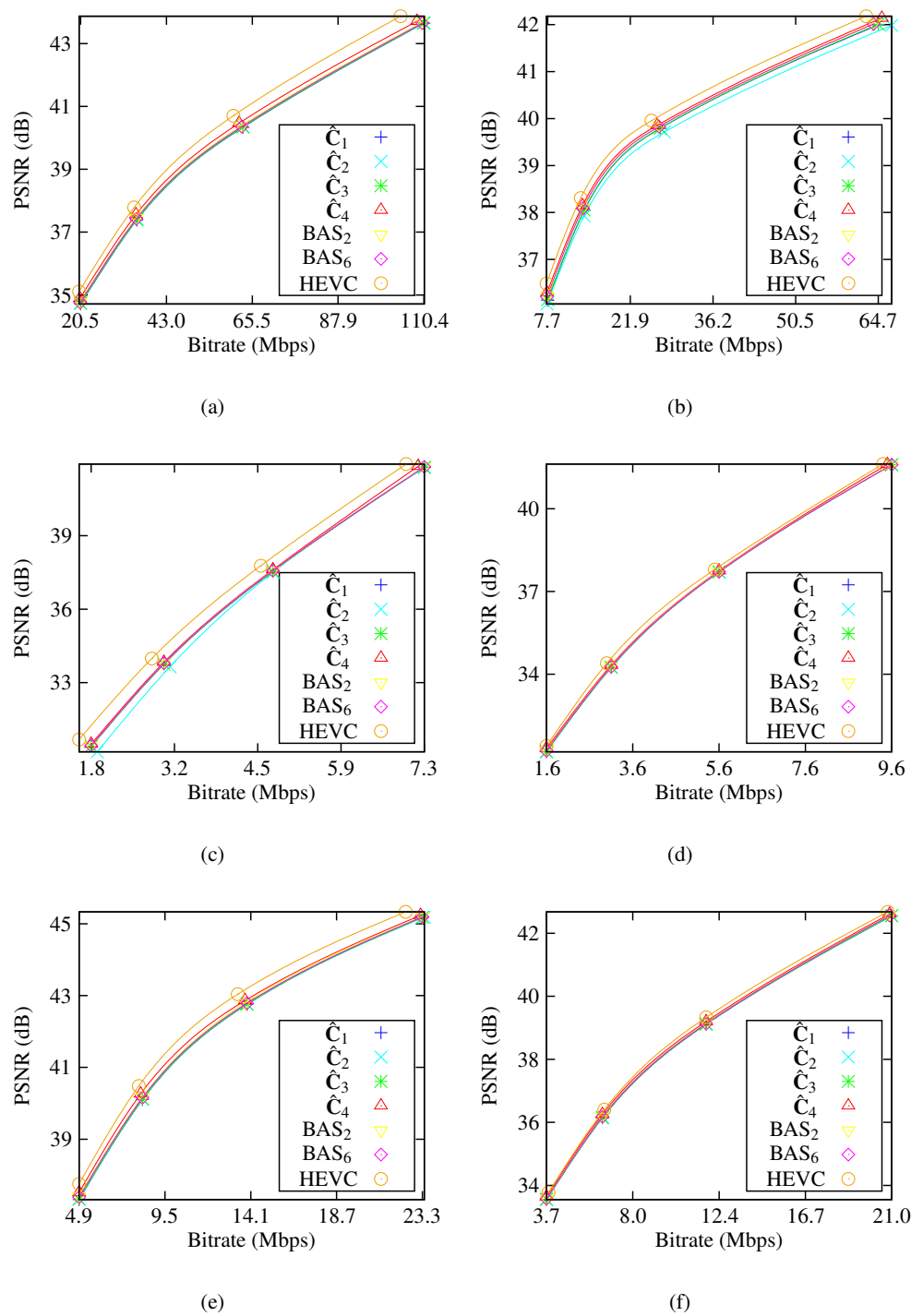


Figure 7. Rate-distortion curves of the modified HEVC software in AI mode for test sequences: (a) 'PeopleOnStreet', (b) 'BasketballDrive', (c) 'RaceHorses', (d) 'BlowingBubbles', (e) 'KristenAndSara', and (f) 'BasketbalDrillText'.

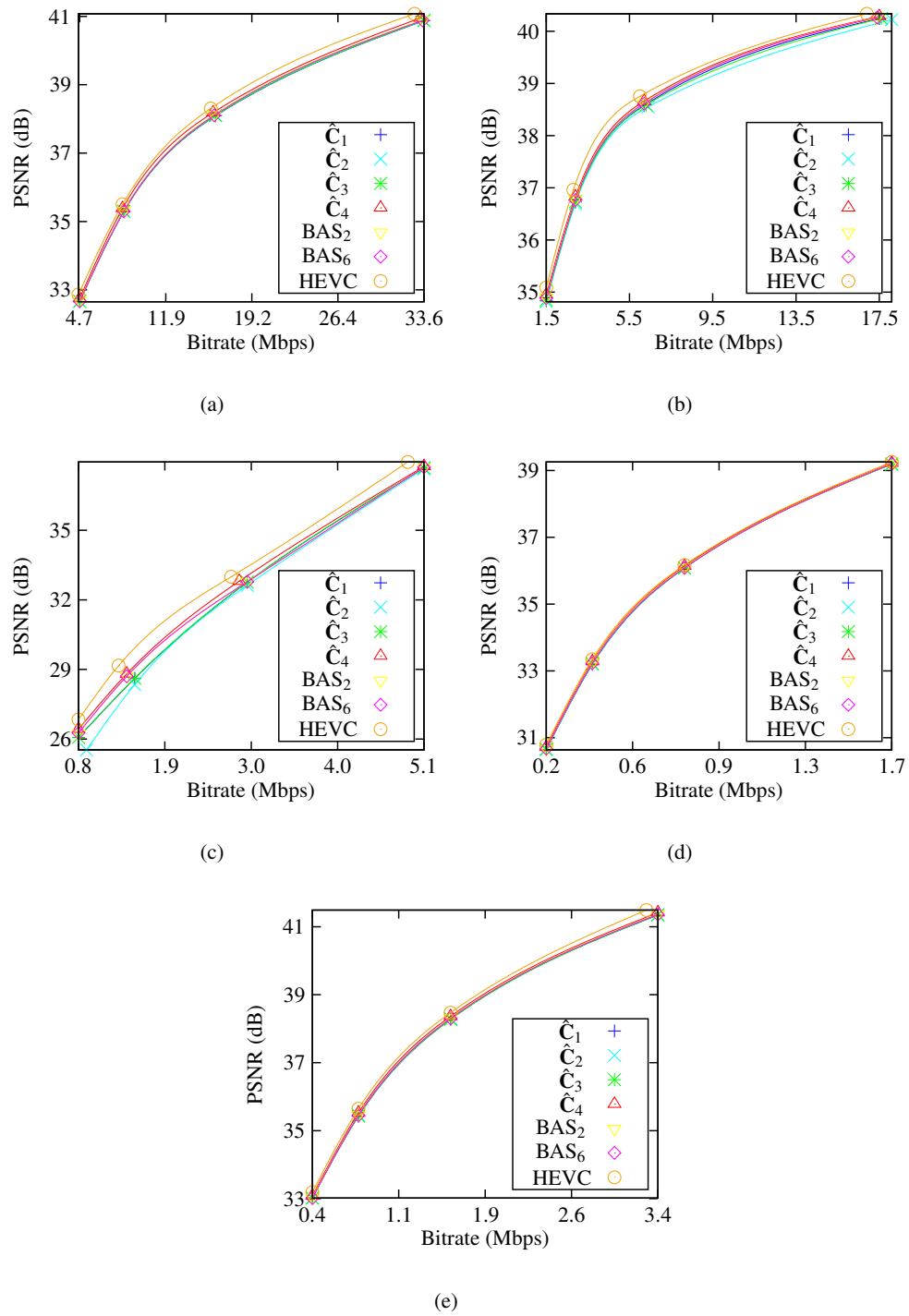


Figure 8. Rate-distortion curves of the modified HEVC software in RA mode for test sequences: (a) 'PeopleOnStreet', (b) 'BasketballDrive', (c) 'RaceHorses', (d) 'BlowingBubbles', and (e) 'BasketbalDrillText'.

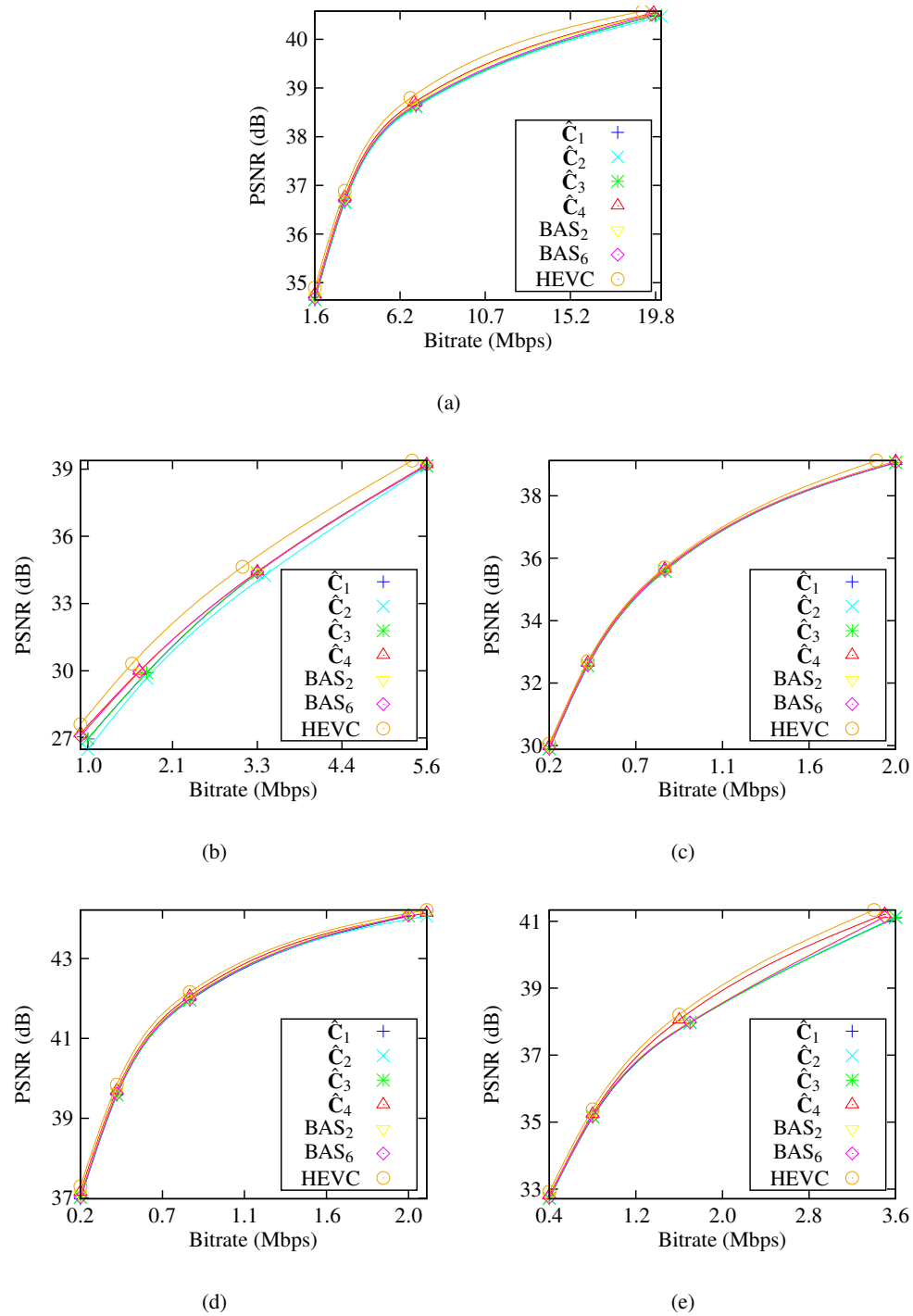


Figure 9. Rate-distortion curves of the modified HEVC software in LDB mode for test sequences: (a) ‘BasketballDrive’, (b) ‘RaceHorses’, (c) ‘BlowingBubbles’, (d) ‘KristenAndSara’, and (e) ‘BasketbalDrillText’.

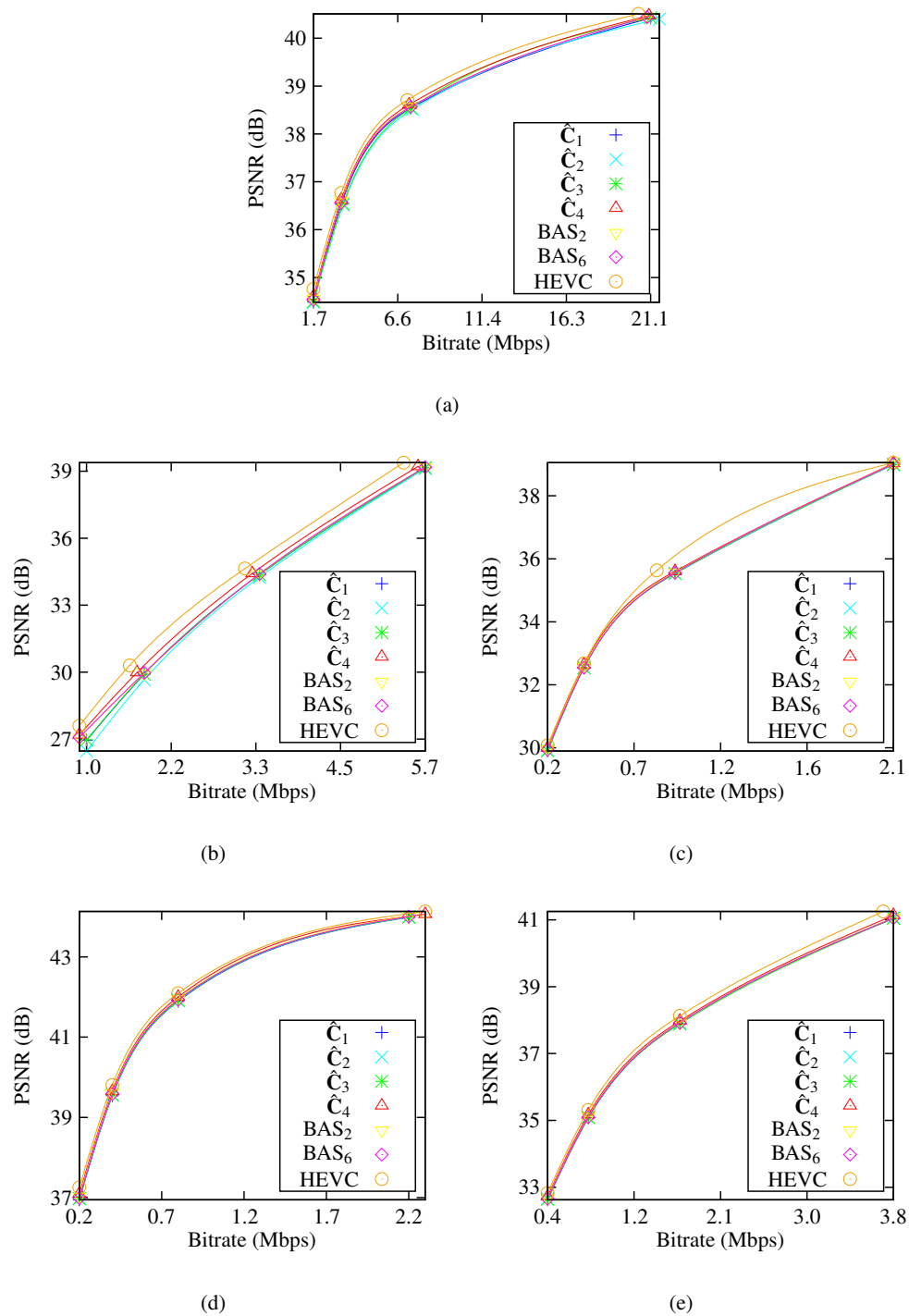


Figure 10. Rate-distortion curves of the modified HEVC software in LDP mode for test sequences: (a) 'BasketballDrive', (b) 'RaceHorses', (c) 'BlowingBubbles', (d) 'KristenAndSara', and (e) 'BasketbalDrillText'.

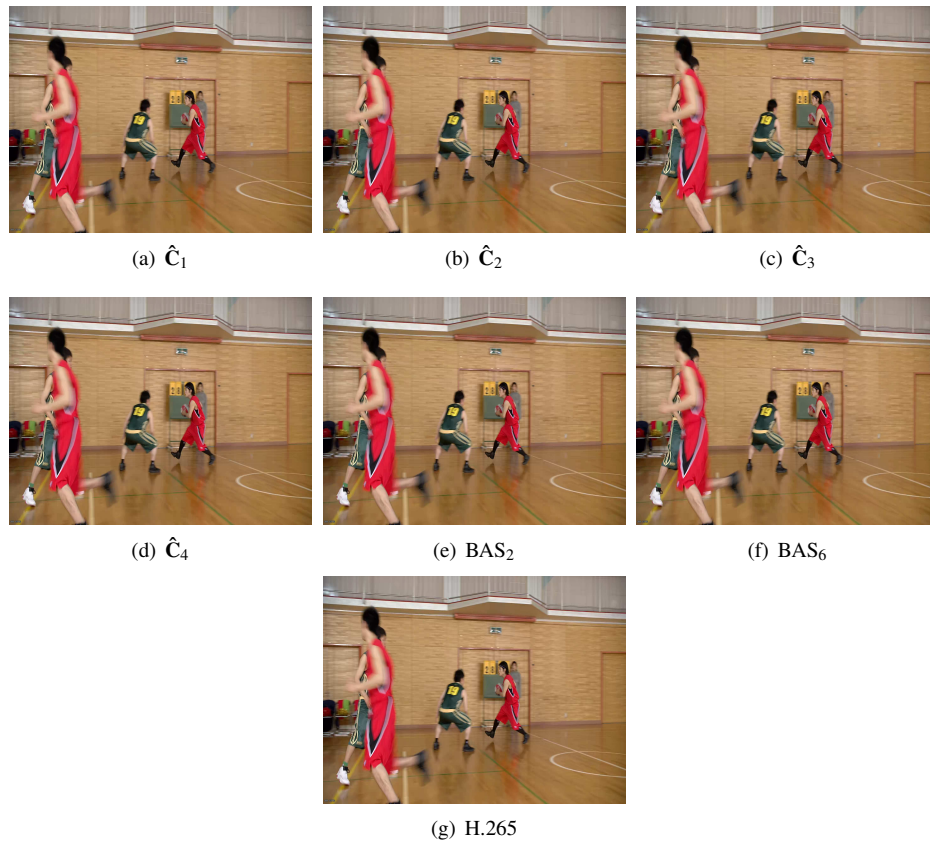


Figure 11. First frame of the H.265/HEVC-compressed ‘BasketballDrive’ sequence with $QP = 32$.

6. FPGA Implementation

To compare the hardware-resource consumption of the discussed approximations, they were initially modeled and tested in Matlab Simulink and then physically realized on FPGA. The FPGA used was a Xilinx Virtex-6 XC6VLX240T installed on a Xilinx ML605 prototyping board. FPGA realization was tested with 100,000 random 8-point input test vectors using hardware cosimulation. Test vectors were generated from within the Matlab environment and routed to the physical FPGA device using a JTAG-based hardware cosimulation. Then, measured data from the FPGA was routed back to Matlab memory space.

We separated the BAS_6 approximation for comparison with efficient approximations because its performance metrics lay on the Pareto frontier of the plots in Figure 2. The associated FPGA implementations were evaluated for hardware complexity and real-time performance using metrics such as configurable logic blocks (CLB) and flip-flop (FF) count, critical path delay (T_{cpd}) in ns, and maximum operating frequency (F_{max}) in MHz. Values were obtained from the Xilinx FPGA synthesis and place-route tools by accessing the `xfLOW.results` report file. In addition, static (Q_p) and dynamic power (D_p in mW/GHz) consumption were estimated using the Xilinx XPower Analyzer. We also reported area-time complexity (AT) and area-time-squared complexity (AT^2). Circuit area (A) was estimated using the CLB count as a metric, and time was derived from T_{cpd} . Table 7 lists the FPGA hardware resource and power consumption for each algorithm.

Considering the circuit complexity of the discussed approximations and BAS_6 [5], as measured from the CBL count for the FPGA synthesis report, it can be seen from Table 7 that T_1 is the smallest option in terms of circuit area. When considering maximum speed, matrix T_5 showed the best performance on the Vertex-6 XC6VLX240T device. Alternatively, if we consider the normalized dynamic power consumption, the best performance was again measured from T_1 .

Table 7. Hardware-resource consumption and power consumption using a Xilinx Virtex-6 XC6VLX240T 1FFG1156 device.

Approximation	CLB	FF	T_{cpd} (ns)	F_{max} (MHz)	D_p (mW/GHz)	Q_p (W)	AT	AT^2
T_1	108	368	2.783	359	2.93	3.455	300.56	836.47
T_3	114	386	2.805	357	3.56	3.462	319.77	896.95
T_5	125	442	2.280	439	3.65	3.472	285	649.8
T_6	185	626	2.699	371	4.18	3.470	499.32	1347.65
BAS_6 [5]	116	316	2.740	365	4.05	3.468	317.84	870.88

7. Conclusions and Final Remarks

In this paper, we introduced a mathematical framework for the design of 8-point DCT approximations. The Loeffler algorithm was parameterized and a class of matrices was derived. Matrices with good properties, such as low-complexity, invertibility, orthogonality or near-orthogonality, and closeness to the exact DCT, were separated according to a multicriteria optimization problem aiming at Pareto efficiency. The DCT approximations in this class were assessed, and the optimal transformations were separated.

The obtained transforms were assessed in terms of computational complexity, proximity, and coding measures. The derived efficient solutions constitute DCT approximations capable of good properties when compared to existing DCT approximations. At the same time, approximation requires extremely low computation costs: only additions and bit-shifting operations are required for their evaluation.

We demonstrated that the proposed method is a unifying structure for several approximations scattered in the literature, including the well-known approximation by Lengwehasatit–Ortega [31], and they share the same matrix factorization scheme. Additionally, because all discussed approximations have a common matrix expansion, the SFG of their associated fast algorithms are identical except for the parameter values. Thus, the resulting structure paves the way to performance-selectable transforms according to the choice of parameters.

Moreover, approximations were assessed and compared in terms of image and video coding. For images, a JPEG-like encoding simulation was considered, and for videos, approximations were embedded in the H.264/AVC and H.265/HEVC video-coding standards. Approximations exhibited good coding performance compared to exact DCT-based JPEG compression and the obtained frame video quality was very close to the results shown by the H.264/AVC and H.265/HEVC standards.

Extensions for larger blocklengths can be achieved by considering scalable approaches such as the one suggested in Reference [71]. Alternatively, one could employ direct parameterization of the 16-point Loeffler DCT algorithm described in Reference [27].

Author Contributions: conceptualization, D.F.G.C. and R.J.C.; methodology, D.F.G.C. and R.J.C.; software, T.L.T.S., P. M. and R.S.O.; hardware, S.K., R.S.O., and A.M.; validation, D.F.G.C., R.J.C., T.L.T.S., P.M. and R.S.O.; writing—original draft preparation, D.F.G.C. and R.J.C.; writing, reviewing, and editing, D.F.G.C., R.J.C., T.L.T.S., R.S.O., F.M.B. and S.K.; supervision, R.J.C.; funding acquisition, R.J.C. and V.S.D.

Funding: Partial support from CAPES, CNPq, FACEPE, and FAPERGS.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DCT	Discrete cosine transform
AVC	Advanced video coding
HEVC	High-efficiency video coding
FPGA	Field-programmable gate array
KLT	Karhunen–Loève transform

DWT	Discrete wavelet transform
JPEG	Joint photographic experts group
MPEG	Moving-picture experts group
MSE	Mean-squared error
SSIM	Structural similarity index metric
PSNR	Peak signal-to-noise ratio
WHT	Walsh–Hadamard transform
QP	Quantization parameter
BD	Bjontegaard delta
CIF	Common intermediate format
CLB	Configurable logic block
FF	Flip-flop

References

1. Ahmed, N.; Rao, K.R. *Orthogonal Transforms for Digital Signal Processing*; Springer: New York, NY, USA, 1975.
2. Britanak, V.; Yip, P.; Rao, K.R. *Discrete Cosine and Sine Transforms*; Academic Press: Orlando, FL, USA, 2007.
3. Kolaczyk, E.D. Methods for analyzing certain signals and images in astronomy using Haar wavelets. In Proceedings of the Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 2–5 November 1997; Volume 1, pp. 80–84.
4. Qiang, Y. Image denoising based on Haar wavelet transform. In Proceedings of the International Conference on Electronics and Optoelectronics, Dalian, China, 29–31 July 2011; Volume 3.
5. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. Binary Discrete Cosine and Hartley Transforms. *IEEE Trans. Circuits Syst. I* **2013**, *60*, 989–1002. [[CrossRef](#)]
6. Martucci, S.A.; Mersereau, R.M. New Approaches To Block Filtering of Images Using Symmetric Convolution and the DST or DCT. In Proceedings of the International Symposium on Circuits and Systems, Chicago, IL, USA, 3–6 May 1993; pp. 259–262.
7. Oppenheim, A.V.; Schaffer, R.W.; Buck, J.R. *Discrete-Time Signal Processing*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1999; Volume 1.
8. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2001.
9. Parfieniuk, M. Using the CS decomposition to compute the 8-point DCT. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 24–27 May 2015; pp. 2836–2839.
10. Coutinho, V.D.A.; Cintra, R.J.; Bayer, F.M.; Kulasekera, S.; Madanayake, A. Low-complexity pruned 8-point DCT approximations for image encoding. In Proceedings of the International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 25–27 February 2015; pp. 1–7.
11. Potluri, U.S.; Madanayake, A.; Cintra, R.J.; Bayer, F.M.; Kulasekera, S.; Edirisuriya, A. Improved 8-point Approximate DCT for Image and Video Compression Requiring Only 14 Additions. *IEEE Trans. Circuits Syst. I* **2014**, *61*, 1727–1740. [[CrossRef](#)]
12. Goebel, J.; Paim, G.; Agostini, L.; Zatt, B.; Porto, M. An HEVC multi-size DCT hardware with constant throughput and supporting heterogeneous CUs. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 2202–2205.
13. Naiel, M.A.; Ahmad, M.O.; Swamy, M.N.S. Approximation of feature pyramids in the DCT domain and its application to pedestrian detection. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 2711–2714.
14. Lalitha, N.V.; Prasad, P.V.; Rao, S.U. Performance analysis of DCT and DWT audio watermarking based on SVD. In Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–5.
15. Masera, M.; Martina, M.; Masera, G. Adaptive Approximated DCT Architectures for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 2714–2725. [[CrossRef](#)]
16. Zhao, M.; Liu, H.; Wan, Y. An improved Canny edge detection algorithm based on DCT. In Proceedings of the IEEE International Conference on Progress in Informatics and Computing (PIC), Nanjing, China, 18–20 December 2015; pp. 234–237.

17. Wang, Y.; Shi, M.; You, S.; Xu, C. DCT Inspired Feature Transform for Image Retrieval and Reconstruction. *IEEE Trans. Image Process.* **2016**, *25*, 4406–4420. [[CrossRef](#)] [[PubMed](#)]
18. Bhaskaran, V.; Konstantinides, K. *Image and Video Compression Standards*; Kluwer Academic Publishers: Norwell, MA, USA, 1997.
19. Wallace, G.K. The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.* **1992**, *38*, xviii–xxxiv. [[CrossRef](#)]
20. Roma, N.; Sousa, L. Efficient hybrid DCT-domain algorithm for video spatial downscaling. *EURASIP J. Adv. Signal Process.* **2007**, *2007*, 057291. [[CrossRef](#)]
21. Wiegand, T.; Sullivan, G.J.; Bjøntegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [[CrossRef](#)]
22. Pourazad, M.T.; Dautre, C.; Azimi, M.; Nasiopoulos, P. HEVC: The New Gold Standard for Video Compression: How Does HEVC Compare with H.264/AVC? *IEEE Consum. Electron. Mag.* **2012**, *1*, 36–46. [[CrossRef](#)]
23. Horn, D.R. Lepton Image Compression: Saving 22% Losslessly from Images at 15MB/s. 2016. Available online: <https://blogs.dropbox.com/tech/2016/07/lepton-image-compression-saving-22-losslessly-from-images-at-15mbs/> (accessed on 10 November 2018).
24. Lee, B. A New Algorithm to Compute the Discrete Cosine Transform. *IEEE Trans. Acoust. Speech Signal Process.* **1984**, *32*, 1243–1245.
25. Arai, Y.; Agui, T.; Nakajima, M. A fast DCT-SQ scheme for images. *IEICE Trans.* **1988**, *E71*, 1095–1097.
26. Feig, E.; Winograd, S. Fast Algorithms for the Discrete Cosine Transform. *IEEE Trans. Signal Process.* **1992**, *40*, 2174–2193. [[CrossRef](#)]
27. Loeffler, C.; Ligtenberg, A.; Moschytz, G.S. Practical Fast 1-D DCT Algorithms with 11 Multiplications. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Glasgow, UK, 23–26 May 1989; Volume 2; pp. 988–991.
28. Duhamel, P.; H'Mida, H. New 2ⁿ DCT algorithms suitable for VLSI implementation. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, USA, 6–9 April 1987; Volume 12, pp. 1805–1808.
29. Heideman, M.T.; Burrus, C.S. *Multiplicative Complexity, Convolution, and the DFT*; Springer: New York, NY, USA, 1988.
30. Haweel, T.I. A new square wave transform based on the DCT. *Signal Process.* **2001**, *81*, 2309–2319. [[CrossRef](#)]
31. Lengwehasatit, K.; Ortega, A. Scalable Variable Complexity Approximate Forward DCT. *IEEE Trans. Circuits Syst. Syst. Video Technol.* **2004**, *14*, 1236–1248. [[CrossRef](#)]
32. Cintra, R.J.; Bayer, F.M. A DCT approximation for image compression. *IEEE Signal Process. Lett.* **2011**, *18*, 579–582. [[CrossRef](#)]
33. Bayer, F.M.; Cintra, R.J. DCT-like transform for image compression requires 14 additions only. *Electron. Lett.* **2012**, *48*, 919–921. [[CrossRef](#)]
34. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. A multiplication-free transform for image compression. In Proceedings of the 2nd International Conference on Signals, Circuits and Systems, Monastir, Tunisia, 7–9 November 2008; pp. 1–4.
35. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. Low-complexity 8x8 transform for image compression. *Electron. Lett.* **2008**, *44*, 1249–1250. [[CrossRef](#)]
36. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. A novel transform for image compression. In Proceedings of the 53rd IEEE International Midwest Symposium on Circuits and Systems, Seattle, WA, USA, 1–4 August 2010; pp. 509–512.
37. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. A low-complexity parametric transform for image compression. In Proceedings of the IEEE International Symposium on Circuits and Systems, Rio de Janeiro, Brazil, 15–18 May 2011; pp. 2145–2148.
38. Cintra, R.J. An Integer Approximation Method for Discrete Sinusoidal Transforms. *Circuits Syst. Signal Process.* **2011**, *30*, 1481–1501. [[CrossRef](#)]
39. Cintra, R.J.; Bayer, F.M.; Tablada, C.J. Low-complexity 8-point DCT approximations based on integer functions. *Signal Process.* **2014**, *99*, 201–214. [[CrossRef](#)]
40. Zhang, D.; Lin, S.; Zhang, Y.; Yu, L. Complexity controllable {DCT} for real-time H.264 encoder. *J. Vis. Commun. Image Represent.* **2007**, *18*, 59–67. [[CrossRef](#)]

41. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. A fast 8×8 transform for image compression. In Proceedings of the International Conference on Microelectronics, Columbus, OH, USA, 4–7 August 2009; pp. 74–77.
42. Chen, Y.J.; Oraintara, S.; Tran, T.D.; Amaratunga, K.; Nguyen, T.Q. Multiplierless approximation of transforms using lifting scheme and coordinate descent with adder constraint. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, USA, 13–17 May 2002, Volume 3.
43. Liang, M.J.; Tran, T.D. Approximating the DCT with the lifting scheme: Systematic design and applications. In Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 29 October–1 November 2000; Volume 1; pp. 192–196.
44. Liang, M.J.; Tran, T.D. Fast multiplierless approximations of the DCT with the lifting scheme. *IEEE Trans. Signal Process.* **2001**, *49*, 3032–3044. [[CrossRef](#)]
45. Malvar, H. Fast computation of discrete cosine transform through fast Hartley transform. *Electron. Lett.* **1986**, *22*, 352–353. [[CrossRef](#)]
46. Malvar, H.S. Fast computation of the discrete cosine transform and the discrete Hartley transform. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1484–1485. [[CrossRef](#)]
47. Malvar, H. Erratum: Fast computation of discrete cosine transform through fast Hartley transform. *Electron. Lett.* **1987**, *23*, 608. [[CrossRef](#)]
48. Kouadria, N.; Doghmane, N.; Messadeg, D.; Harize, S. Low complexity DCT for image compression in wireless visual sensor networks. *Electron. Lett.* **2013**, *49*, 1531–1532. [[CrossRef](#)]
49. Manassah, J.T. *Elementary Mathematical and Computational Tools for Electrical and Computer Engineers Using Matlab®*; CRC Press: New York, NY, USA, 2001.
50. Blahut, R.E. *Fast Algorithms for Digital Signal Processing*; Cambridge University Press: Cambridge, UK, 2010.
51. Halmos, P.R. *Finite Dimensional Vector Spaces*; Springer: New York, NY, USA, 2013.
52. Graham, R.L.; Knuth, D.E.; Patashnik, O. *Concrete Mathematics*; Addison-Wesley Publishing Company: Reading, MA, USA, 1989.
53. Shores, T.S. *Applied Linear Algebra and Matrix Analysis*; Springer: Lincoln, NE, USA, 2007.
54. Tablada, C.J.; Bayer, F.M.; Cintra, R.J. A Class of DCT Approximations Based on the Feig-Winograd Algorithm. *Signal Process.* **2015**, *113*, 38–51. [[CrossRef](#)]
55. Strang, G. *Linear Algebra and Its Applications*, 4th ed.; Thomson Learning; Massachusetts Institute of Technology: Cambridge, MA, USA, 2005.
56. Katto, J.; Yasuda, Y. Performance evaluation of subband coding and optimization of its filter coefficients. *J. Vis. Commun. Image Represent.* **1991**, *2*, 303–313. [[CrossRef](#)]
57. Ehrgott, M. *Multicriteria Optimization*, 2 ed.; Springer: Berlin, Germany, 2005.
58. Barichard, V.; Ehrgott, M.; Gandibleux, X.; T'Kindt, V. *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, 1 ed.; Springer: Berlin, Germany, 2009.
59. Kay, S.M. *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1993; Volume 1.
60. Kay, S.M. *Fundamentals of Statistical Signal Processing: Detection Theory*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1998; Volume 2.
61. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
62. x264 Team. VideoLAN Organization. Available online: <http://www.videolan.org/developers/x264.html> (accessed on 10 November 2018).
63. Richardson, I. *The H.264 Advanced Video Compression Standard*, 2nd ed.; John Wiley and Sons: Hoboken, NJ, USA, 2010.
64. Gordon, S.; Marpe, D.; Wiegand, T. *Simplified Use of 8×8 Transform—Updated Proposal and Results*; Technical Report; Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG: Munich, Germany, 2004.
65. Xiph.Org Foundation. Xiph.org Video Test Media. 2014. Available online: <https://media.xiph.org/video/derf/> (accessed on 10 November 2018).
66. Bjøntegaard, G. Calculation of Average PSNR Differences between RD-curves. In Proceedings of the 13th VCEG Meeting, Austin, TX, USA, 2–4 April 2001.
67. Hanhart, P.; Ebrahimi, T. Calculation of average coding efficiency based on subjective quality scores. *J. Vis. Commun. Image Represent.* **2014**, *25*, 555–564. [[CrossRef](#)]

68. Tan, T.K.; Weerakkody, R.; Mrak, M.; Ramzan, N.; Baroncini, V.; Ohm, J.R.; Sullivan, G.J. Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 76–90. [CrossRef]
69. Bossen, F. 12th Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. Available online: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7281 (accessed on 8 November 2018).
70. Joint Collaborative Team on Video Coding (JCT-VC). *HEVC References Software Documentation*; Fraunhofer Heinrich Hertz Institute: Berlin, Germany, 2013.
71. Jridi, M.; Alfalou, A.; Meher, P.K. A Generalized Algorithm and Reconfigurable Architecture for Efficient and Scalable Orthogonal Approximation of DCT. *IEEE Trans. Circuits Syst. I* **2015**, *62*, 449–457. [CrossRef]
72. Bossen, F. *Common Test Conditions and Software Reference Configurations*; Document JCT-VC L1100; 2013. Available online: http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7281 (accessed on 8 November 2018).
73. Naccari, M.; Mrak, M. Chapter 5—Perceptually Optimized Video Compression. In *Academic Press Library in Signal Processing Image and Video Compression and Multimedia*; Theodoridis, S., Chellappa, R., Eds.; Elsevier: Amsterdam, Netherlands, 2014; Volume 5, pp. 155–196.
74. Wien, M. *High Efficiency Video Coding*; Signals and Communication Technology, Springer: Berlin/Heidelberg, Germany, 2015; p. 314.
75. Budagavi, M.; Fuldseth, A.; Bjøntegaard, G.; Sze, V.; Sadafale, M. Core Transform Design in the High Efficiency Video Coding (HEVC) Standard. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 1029–1041. [CrossRef]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).