Improved 8-Point Approximate DCT for Image and Video Compression Requiring Only 14 Additions

Article *in* Circuits and Systems I: Regular Papers, IEEE Transactions on · June 2014

DOI: 10.1109/TCSI.2013.2295022

Arjuna Madanayake
University of Akron
258 PUBLICATIONS **2,681** CITATIONS
SEE PROFILE

Fábio M. Bayer
Universidade Federal de Santa Maria
102 PUBLICATIONS **1,177** CITATIONS
SEE PROFILE

Improved 8-point Approximate DCT for Image and Video Compression Requiring Only 14 Additions

Uma Sadhvi Potluri* Arjuna Madanayake*
Renato J. Cintra† Fábio M. Bayer‡
Sunera Kulasekera* Amila Edirisuriya*

Abstract

Video processing systems such as HEVC requiring low energy consumption needed for the multimedia market has lead to extensive development in fast algorithms for the efficient approximation of 2-D DCT transforms. The DCT is employed in a multitude of compression standards due to its remarkable energy compaction properties. Multiplier-free approximate DCT transforms have been proposed that offer superior compression performance at very low circuit complexity. Such approximations can be realized in digital VLSI hardware using additions and subtractions only, leading to significant reductions in chip area and power consumption compared to conventional DCTs and integer transforms. In this paper, we introduce a novel 8-point DCT approximation that requires *only* 14 addition operations and no multiplications. The proposed transform possesses low computational complexity and is compared to state-of-the-art DCT approximations in terms of both algorithm complexity and peak signal-to-noise ratio. The proposed DCT approximation is a candidate for reconfigurable video standards such as HEVC. The proposed transform and several other DCT approximations are mapped to systolic-array digital architectures and physically realized as digital prototype circuits using FPGA technology and mapped to 45 nm CMOS technology.

Keywords Approximate DCT, low-complexity algorithms, image compression, HEVC, low power consumption

1 Introduction

Recent years have experienced a significant demand for high dynamic range systems that operate at high resolutions [1]. In particular, high-quality digital video in multimedia devices [2] and video-

*Uma Sadhvi Potluri, Arjuna Madanayake, Sunera Kulasekera and Amila Edirisuriya are with the Department of Electrical and Computer Engineering, The University of Akron, Akron, OH, USA (e-mail: arjuna@uakron.edu).

†Renato J. Cintra is with the Signal Processing Group, Departamento de Estatística, Universidade Federal de Pernambuco, Recife, PE, Brazil (e-mail: rjdsc@ieee.org).

‡Fábio M. Bayer is with the Departamento de Estatística and Laboratório de Ciências Espaciais de Santa Maria (LACESM), Universidade Federal de Santa Maria, Santa Maria, RS, Brazil (e-mail: bayer@ufsm.br).

over-Internet protocol networks [3] are prominent areas where such requirements are evident. Other noticeable fields are geospatial remote sensing [4], traffic cameras [5], automatic surveillance [1], homeland security [6], automotive industry [7], and multimedia wireless sensor networks [8], to name but a few. Often hardware capable of significant throughput is necessary; as well as allowable area-time complexity [8].

In this context, the discrete cosine transform (DCT) [9–11] is an essential mathematical tool in both image and video coding [8,11–15]. Indeed, the DCT was demonstrated to provide good energy compaction for natural images, which can be described by first-order Markov signals [10, 11, 13]. Moreover, in many situations, the DCT is a very close substitute for the Karhunen-Loève transform (KLT), which has optimal properties [9–11,13,14,16]. As a result, the two-dimensional (2-D) version of the 8-point DCT was adopted in several imaging standards such as JPEG [17], MPEG-1 [18], MPEG-2 [19], H.261 [20], H.263 [21, 22], and H.264/AVC [23, 24].

Additionally, new compression schemes such as the High Efficiency Video Coding (HEVC) employs DCT-like integer transforms operating at various block sizes ranging from 4×4 to 32×32 pixels [25–27]. The distinctive characteristic of HEVC is its capability of achieving high compression performance at approximately half the bit rate required by H.264/AVC with same image quality [25–27]. Also HEVC was demonstrated to be especially effective for high-resolution video applications [27]. However, HEVC possesses a significant computational complexity in terms of arithmetic operations [26–28]. In fact, HEVC can be 2–4 times more computationally demanding when compared to H.264/AVC [26]. Therefore, low complexity DCT-like approximations may benefit future video codecs including emerging HEVC/H.265 systems.

Several efficient algorithms were developed and a noticeable literature is available [10, 29–35]. Although fast algorithms can significantly reduce the computational complexity of computing the DCT, floating-point operations are still required [11]. Despite their accuracy, floating-point operations are expensive in terms of circuitry complexity and power consumption. Therefore, minimizing the number of floating-point operations is a sought property in a fast algorithm. One way of circumventing this issue is by means of approximate transforms.

The aim of this paper is two-fold. First, we introduce a new DCT approximation that possesses an extremely low arithmetic complexity, *requiring only 14 additions*. This novel transform was obtained by means of solving a tailored optimization problem aiming at minimizing the transform computational cost. Second, we propose hardware implementations for several 2-D 8-point approximate DCT. The approximate DCT methods under consideration are (i) the proposed transform; (ii) the 2008 Bouguezel-Ahmad-Swamy (BAS) DCT approximation [36]; (iii) the parametric transform for image compression [37]; (iv) the Cintra-Bayer (CB) approximate DCT based on the rounding-off function [38]; (v) the modified CB approximate DCT [39]; and (vi) the DCT approximation proposed in [40] in the context of beamforming. All introduced implementations are sought to be fully parallel time-multiplexed 2-D architectures for 8×8 data blocks. Additionally, the proposed designs are based on successive calls of 1-D architectures taking advantage of the separability property of the 2-D DCT kernel. Designs were thoroughly assessed and compared.

This paper unfolds as follows. In Section 2, we discuss the role of DCT-like fast algorithms for video CODECs while proposing some new possibilities for low-power video processing where rapid reconfiguration of the hardware realization is possible. In Section 3, we review selected approximate methods for DCT computation and describe associate fast algorithms in terms of matrix factorizations. Section 4 details the proposed transform and its fast algorithm based on matrix factorizations. Section 5 discusses the computational complexity of the approximate DCT techniques. Performance measures are also quantified and evaluated to assess the proposed approximate DCT as well as the remaining selected approximations. In Section 6 digital hardware architectures for discussed algorithms are supplied both for 1-D and 2-D analysis. Hardware resource consumptions using field programmable gate array (FPGA) and CMOS 45 nm application-specific integrated circuit (ASIC) technologies are tabulated. Conclusions and final remarks are in Section 7.

2 Reconfigurable DCT-like Fast Algorithms in Video CODECs

In current literature, several approximate methods for the DCT calculation have been archived [11]. While not computing the DCT exactly, such approximations can provide meaningful estimations at low-complexity requirements. In particular, some DCT approximations can totally eliminate the requirement for floating-point operations—all calculations are performed over a fixed-point arithmetic framework. Prominent 8-point approximation-based techniques were proposed in [14, 15,36–44]. Works addressing 16-point DCT approximations are also archived in literature [43,45,46].

In general, these approximation methods employ a transformation matrix whose elements are defined over the set $\{0, \pm 1/2, \pm 1, \pm 2\}$. This implies null multiplicative complexity, because the required operations can be implemented exclusively by means of binary additions and shift operations. Such DCT approximations can provide low-cost and low-power designs and effectively replace the *exact* DCT and other DCT-like transforms. Indeed, the performance characteristics of the low complexity DCT approximations appear similar to the exact DCT, while their associate hardware implementations are economical because of the absence of multipliers [14, 15, 36–43, 43–46]. As a consequence, some prospective applications of DCT approximations are found in real-time video transmission and processing.

Emerging video standards such as HEVC provide for reconfigurable operation on-the-fly which makes the availability of an ensemble of fast algorithms and digital VLSI architectures a valuable asset for low-energy high-performance embedded systems. For certain applications, low circuit complexity and/or power consumption is the driving factor, while for certain other applications, highest picture quality for reasonably low power consumption and/or complexity may be more important. In emerging systems, it may be possible to switch *modus operandi* based on the demanded picture quality vs available energy in the device. Such feature would be invaluable in high quality smart video devices demanding extended battery life. Thus, the availability of a suite of fast algorithms and implementation libraries for several efficient DCT approximation algorithms may be a welcoming contribution.

For example, in a future HEVC system, it may be possible to reconfigure the DCT engine to use a higher complexity DCT approximation which offers better signal-to-noise ratio (SNR) when the master device is powered by a remote power source, and then have the device seamlessly switch into a low complexity fast DCT algorithm when the battery storage falls below a certain threshold, for example [47]. Alternatively, the CODEC may be reconfigured in real-time to switch between different DCT approximations offering varying picture quality and power consumptions depending on the measured SNR of the incoming video stream, which would be content specific and very difficult to predict without resorting to real-time video metrics [48].

Furthermore, another possible application for a suite of DCT approximation algorithms in the light of reconfigurable video codecs is the intelligent intra-frame fast reconfiguration of the DCT core to take into account certain local frame information and measured SNR metrics. For example, certain parts of a frame can demand better picture quality (foreground, say) when compared to relatively unimportant part of the frame (background, say) [48]. In such a case, it may be possible to switch DCT approximations algorithms on an intra frame basis to take into account the varying demands for picture clarity within a frame as well as the availability of reconfigurable logic based digital DCT engines that support fast reconfiguration in real-time.

3 Review of Approximate DCT Methods

In this section, we review the mathematical description of the selected 8-point DCT approximations. All discussed methods here consist of a transformation matrix that can be put in the following format:

$$[\text{diagonal matrix}] \times [\text{low-complexity matrix}].$$

The diagonal matrix usually contains irrational numbers in the form $1/\sqrt{m}$, where m is a small positive integer. In principle, the irrational numbers required in the diagonal matrix would require an increased computational complexity. However, in the context of image compression, the diagonal matrix can simply be absorbed into the quantization step of JPEG-like compression procedures [15, 36–39, 42]. Therefore, in this case, the complexity of the approximation is bounded by the complexity of the low-complexity matrix. Since the entries of the low complexity matrix comprise only powers of two in $\{0, \pm 1/2, \pm 1, \pm 2\}$, null multiplicative complexity. is achieved.

In the next subsections, we detail these methods in terms of its transformation matrices and the associated fast algorithms obtained by matrix factorization techniques. All derived fast algorithms employ sparse matrices whose elements are the above-mentioned powers of two.

3.1 Bouguezel-Ahmad-Swamy Approximate DCT

In [36], a low-complexity approximate was introduced by Bouguezel *et al.* We refer to this approximate DCT as BAS-2008 approximation. The BAS-2008 approximation $\mathbf{C}_1$ has the following mathematical structure:

$$\mathbf{C_1} = \mathbf{D_1} \cdot \mathbf{T_1} = \mathbf{D_1} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{D}_1 = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}\right)$. A fast algorithm for matrix $\mathbf{T_1}$ can be derived by means of matrix factorization. Indeed, $\mathbf{T_1}$ can be written as a product of three sparse matrices having $\{0, \pm 1/2, \pm 1\}$ elements as shown below [36]: $\mathbf{T_1} = \mathbf{A_3} \cdot \mathbf{A_2} \cdot \mathbf{A_1}$, where $\mathbf{A_1} = \begin{bmatrix} \mathbf{I}_4 & \bar{\mathbf{I}}_4 \\ \bar{\mathbf{I}}_4 & -\mathbf{I}_4 \end{bmatrix}$,

$$\mathbf{A_2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{A_3} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrices $\mathbf{I}_n$ and $\bar{\mathbf{I}}_n$ denote the identity and counter-identity matrices of order n, respectively. It is recognizable that matrix $\mathbf{A_1}$ is the well-known decimation-in-frequency structure present in several fast algorithms [11].

3.2 Parametric Transform

Proposed in 2011 by Bouguezel-Ahmad-Swamy [37], the parametric transform is an 8-point orthogonal transform containing a single parameter a in its transformation matrix $\mathbf{C}^{(a)}$. In this work, we refer to this method as the BAS-2011 transform. It is given as follows:

$$\mathbf{C}^{(a)} = \mathbf{D}^{(a)} \cdot \mathbf{T}^{(a)} = \mathbf{D}^{(a)} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \end{bmatrix},$$

where $\mathbf{D}^{(a)} = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{4+4a^2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{4+4a^2}}\right)$. Usually the parameter a is selected as a small integer in order to minimize the complexity of $\mathbf{T}^{(a)}$. In [37], suggested values are $a \in \{0, 1/2, 1\}$. The value $a = 1/2$ will not be considered in our analyses because in hardware it represents a right-shift which may incur in computational errors. Another possible value that furnishes a low-complexity, error-free transform is $a = 2$. The matrix factorization of $\mathbf{T}^{(a)}$ that leads to its fast algorithm is [37]: $\mathbf{T}^{(a)} = \mathbf{P_1} \cdot \mathbf{Q}^{(a)} \cdot \mathbf{A_4} \cdot \mathbf{A_1}$, where $\mathbf{Q}^{(a)} = \operatorname{diag}\left(\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right], \left[\begin{smallmatrix} a & 1 \\ -1 & a \end{smallmatrix}\right], \mathbf{I}_4 \right)$, and $\mathbf{A_4} = \operatorname{diag}\left(\left[\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{smallmatrix}\right], \mathbf{I}_2, \left[\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right] \right)$. Matrix $\mathbf{P_1}$ performs the simple permutation (1)(2 5 6 4 8 7)(3), where cyclic notation is employed [49, p. 77]. This is a compact notation to denote permutation. In this particular case, it means that component indices are permuted according to $2 \to 5 \to 6 \to 4 \to 8 \to 7 \to 2$. Indices 1 and 3 are unchanged. Therefore, $\mathbf{P_1}$ represents no computational complexity.

3.3 CB-2011 Approximation

By means of judiciously rounding-off the elements of the exact DCT matrix, a DCT approximation was obtained and described in [38]. The resulting 8-point approximation matrix is orthogonal and contains only elements in $\{0, \pm 1\}$. Clearly, it possesses very low arithmetic complexity [38]. The matrix derived transformation matrix $\mathbf{C_2}$ is given by:

$$\mathbf{C_2} = \mathbf{D_2} \cdot \mathbf{T_2} = \mathbf{D_2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix},$$

where $\mathbf{D_2} = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right)$. An efficient factorization for the fast algorithm for $\mathbf{T_2}$ was proposed in [38] as described below: $\mathbf{T_2} = \mathbf{P_2} \cdot \mathbf{A_6} \cdot \mathbf{A_5} \cdot \mathbf{A_1}$, where $\mathbf{A_5} = \operatorname{diag}\left(\left[\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{smallmatrix}\right], \left[\begin{smallmatrix} -1 & 1 & -1 & 0 \\ -1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 1 \\ 0 & 1 & 1 & 1 \end{smallmatrix}\right] \right)$ and $\mathbf{A_6} = \operatorname{diag}\left(\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right], -1, \mathbf{I}_5 \right)$. Matrix $\mathbf{P_2}$ corresponds to the following permutation: (1)(2 5 8)(3 7 6 4).

3.4 Modified CB-2011 Approximation

The transform proposed in [39] is obtained by replacing elements of the CB-2011 matrix with zeros. The resulting matrix is given by:

$$\mathbf{C_3} = \mathbf{D_3} \cdot \mathbf{T_3} = \mathbf{D_3} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{D_3} = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}} \right)$. Matrix $\mathbf{T_3}$ can be factorized into $\mathbf{T_3} = \mathbf{P_2} \cdot \mathbf{A_6} \cdot \mathbf{A_7} \cdot \mathbf{A_1}$, where $\mathbf{A_7} = \operatorname{diag}\left(\left[\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{smallmatrix}\right], -\mathbf{I}_3, 1 \right)$. This particular DCT approximation has the distinction of requiring only 14 additions for its computation [39].

3.5 Approximate DCT in [40]

In [40], a DCT approximation tailored for a particular radio-frequency (RF) application was obtained in accordance with an exhaustive computational search. This transformation is given by

$$\mathbf{C_4} = \mathbf{D_4} \cdot \mathbf{T_4} = \mathbf{D_4} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix},$$

where $\mathbf{D_4} = \frac{1}{2} \cdot \operatorname{diag}\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{3}}\right)$. The fast algorithm for its computation consists of the following matrix factorization: $\mathbf{T_4} = \mathbf{P_3} \cdot \mathbf{A_9} \cdot \mathbf{A_8} \cdot \mathbf{A_1}$, where $\mathbf{A_9} = \operatorname{diag}\left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}, \mathbf{I}_4\right)$, $\mathbf{A_8} = \operatorname{diag}\left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 2 \\ -1 & -2 & 0 & 1 \\ 1 & 0 & -2 & 1 \\ -2 & 1 & -1 & 0 \end{bmatrix}\right)$, and matrix $\mathbf{P_3}$ denotes the permutation (1)(2 5)(3)(4 7 6)(8).

4 Proposed Transform

We aim at deriving a novel low-complexity approximate DCT. For such end, we propose a search over the 8 × 8 matrix space in order to find candidate matrices that possess low computation cost. Let us define the cost of a transformation matrix as the number of arithmetic operations required for its computation. One way to guarantee good candidates is to restrict the search to matrices whose entries do not require multiplication operations. Thus we have the following optimization problem:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \operatorname{cost}(\mathbf{T}), \tag{1}$$

where $\mathbf{T}^*$ is the sought matrix and $\operatorname{cost}(\mathbf{T})$ returns the arithmetic complexity of $\mathbf{T}$. Additionally, the following constraints were adopted:

1. Elements of matrix $\mathbf{T}$ must be in $\{0, \pm 1, \pm 2\}$ to ensure that resulting multiplicative complexity is null;

2. We impose the following form for matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 \\ a_0 & a_2 & a_4 & a_6 & -a_6 & -a_4 & -a_2 & -a_0 \\ a_1 & a_5 & -a_5 & -a_1 & -a_1 & -a_5 & a_5 & a_1 \\ a_2 & -a_6 & -a_0 & -a_4 & a_4 & a_0 & a_6 & -a_2 \\ a_3 & -a_3 & -a_3 & a_3 & a_3 & -a_3 & -a_3 & a_3 \\ a_4 & -a_0 & a_6 & a_2 & -a_2 & -a_6 & a_0 & -a_4 \\ a_5 & -a_1 & a_1 & -a_5 & -a_5 & a_1 & -a_1 & a_5 \\ a_6 & -a_4 & a_2 & -a_0 & a_0 & -a_2 & a_4 & -a_6 \end{bmatrix},$$

   where $a_i \in \{0, 1, 2\}$, for $i = 0, 1 \ldots, 6$;

3. All rows of $\mathbf{T}$ are non-null;

4. Matrix $\mathbf{T}\cdot\mathbf{T}^\top$ must be a diagonal matrix to ensure orthogonality of the resulting approximation [50].

Constraint 2) is required to preserve the DCT-like matrix structure. We recall that the *exact* 8-point DCT matrix is given by [35]:

$$\mathbf{C} = \frac{1}{2}\cdot\begin{bmatrix} \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 \\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0 \\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1 \\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2 \\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 \\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4 \\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5 \\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6 \end{bmatrix},$$

where $\gamma_k = \cos(2\pi(k+1)/32)$, $k = 0, 1, \ldots, 6$.

Above optimization problem is algebraically intractable. Therefore we resorted to exhaustive computational search. As a result, eight candidate matrices were found, including the transform matrix proposed in [39]. Among these minimal cost matrices, we separated the matrix that presents the best performance in terms of image quality of compressed images according the JPEG-like technique employed in [36–39, 41–44], and briefly reviewed in next Section 5.

An important parameter in the image compression routine is the number of retained coefficients in the transform domain. In several applications, the number of retained coefficients is very low. For instance, considering 8×8 image blocks, (i) in image compression using support vector machine, only the first 8–16 coefficients were considered [51]; (ii) Mandyam *et al.* proposed a method for image reconstruction based on only three coefficients; and Bouguezel *et al.* employed only 10 DCT coefficients when assessing image compression methods [41, 42]. Retaining a very small number of coefficients is also common for other image block sizes. In high speed face recognition applications, Pan *et al.* demonstrated that just 0.34%–24.26% out of 92×112 DCT coefficients are sufficient [52, 53]. Therefore, as a compromise, we adopted the number of retained coefficients equal to 10, as suggested in the experiments by Bouguezel *et al.* [41, 42].

The solution of (1) is the following DCT approximation:

$$\mathbf{C}^* = \mathbf{D}^*\cdot\mathbf{T}^* = \mathbf{D}^*\cdot\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix},$$

where $\mathbf{D}^* = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}\right)$. Matrix $\mathbf{T}^*$ has entries in $\{0, \pm 1\}$ and it can be given a sparse factorization according to: $\mathbf{T}^* = \mathbf{P_4}\cdot\mathbf{A_{12}}\cdot\mathbf{A_{11}}\cdot\mathbf{A_1}$, where $\mathbf{A_{11}} = \operatorname{diag}\left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \mathbf{I}_4\right)$, $\mathbf{A_{12}} = \operatorname{diag}\left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, -1, \mathbf{I}_5\right)$, and $\mathbf{P}_4$ is the permutation (1)(2 5 6 8 4 3 7).

5 Computational Complexity and Performance Analysis

The performance of the DCT approximations is often a trade-off between accuracy and computational complexity of a given algorithm [11]. In this section, we assess the computational complexity of the discussed methods and objectively compare them. Additionally, we separate several performance measures to quantify how "close" each approximation are; and to evaluate their performance as image compression tool.

5.1 Arithmetic Complexity

We adopt the arithmetic complexity as figure of merit for estimating the computational complexity. The arithmetic complexity consists of the number of elementary arithmetic operations (additions/subtractions, multiplications/divisions, and bit-shift operations) required to compute a given transformation. In other words, in all cases, we focus our attention to the low-complexity matrices: $\mathbf{T_1}$, $\mathbf{T}^{(a)}$, $\mathbf{T_2}$, $\mathbf{T_3}$, $\mathbf{T_4}$, and the proposed matrix $\mathbf{T}^*$. For instance, in the context of image and video compression, the complexity of the diagonal matrix can be absorbed into the quantization step [15, 36–39, 42]; therefore the diagonal matrix does not contribute towards an increase of the arithmetic complexity [38, 39].

Because all considered DCT approximations have null multiplicative complexity, we resort to comparing them in terms of their arithmetic complexity assessed by the number of additions/subtractions and bit-shift operations. Table 1 displays the obtained complexities. We also include the complexity of the *exact* DCT calculated (i) directly from definition [10] and (ii) according to Arai fast algorithm for the exact DCT [33].

We derived a fast algorithm for the proposed transform, employing *only 14 additions*. This is the same very low-complexity exhibited by the Modified CB-2011 approximation [39]. To the best of our knowledge these are DCT approximations offering the lowest arithmetic complexity in literature.

5.2 Comparative Performance

We employed three classes of assessment tools: (i) matrix proximity metrics with respect to the exact DCT matrix; (ii) transform-related measures; and (iii) image quality measures in image compression. For the first class of measures, we adopted the total error energy [38] and the mean-square error (MSE) [11,13]. For transform performance evaluation, we selected the transform coding gain (C_g) [11, 13] and the transform efficiency (η) [11, 54]. Finally, for image quality assessment we employed the peak SNR (PSNR) [55, 56] and the universal quality index (UQI) [57]. Next subsections furnish a brief description of each of these measures.

Table 1: Arithmetic complexity analysis

Method	Mult	Add	Shifts	Total
Exact DCT (Definition) [10]	64	56	0	120
Arai algorithm (exact) [33]	5	29	0	34
BAS-2008 [36]	0	18	2	20
BAS-2011 [37] with $a = 0$	0	16	0	16
BAS-2011 [37] with $a = 1$	0	18	0	18
BAS-2011 [37] with $a = 2$	0	18	2	20
CB-2011 [38]	0	22	0	22
Modified CB-2011 [39]	0	14	0	14
Approximate DCT in [40]	0	24	6	30
Proposed transform	0	14	0	14

5.2.1 Matrix Proximity Metrics

Let $\hat{\mathbf{C}}$ be an approximate DCT matrix and $\mathbf{C}$ be the exact DCT matrix. The total error energy is an energy-based measure for quantifying the "distance" between $\mathbf{C}$ and $\hat{\mathbf{C}}$. It is described as follows [38].

Let $H_m(\omega; \mathbf{T})$ is the transfer function of the m-th row of a given matrix $\mathbf{T}$ as shown below:

$$H_m(\omega; \mathbf{T}) = \sum_{n=1}^{8} t_{m,n} \exp\left\{-\jmath\,(n-1)\,\omega\right\}, \quad m = 1, 2, \ldots, 8,$$

where $\jmath = \sqrt{-1}$ and $t_{m,n}$ is the (m, n)-th element of $\mathbf{T}$. Then the row-wise error energy related to the difference between $\mathbf{C}$ and $\hat{\mathbf{C}}$ is furnished by:

$$D_m(\omega; \hat{\mathbf{C}}) \triangleq \left| H_m(\omega; \mathbf{C} - \hat{\mathbf{C}}) \right|^2, \quad m = 1, 2, \ldots, 8.$$

We note that, for each row m at any angular frequency $\omega \in [0, \pi]$ in radians per sample, $D_m(\omega; \mathbf{C} - \hat{\mathbf{C}})$ expression quantifies how discrepant a given approximation matrix $\hat{\mathbf{C}}$ is from the matrix $\mathbf{C}$. In this way, a total error energy departing from the *exact* DCT can be obtained by [38]:

$$\epsilon = \sum_{m=1}^{8} \int_0^{\pi} D_m(\omega; \hat{\mathbf{C}})\, \mathrm{d}\omega.$$

Above integral can be computed by numerical quadrature methods [58].

For the MSE evaluation, we assume that the input signal is a first-order Gaussian Markov process with zero-mean, unit variance, and correlation equal to 0.95 [11,13]. Typically images satisfy these requirements [11]. The MSE is mathematically detailed in [11, 13] and should be minimized to maintain the compatibility between the approximation and the *exact* DCT outputs [11].

5.2.2 Transform-related Measures

The transform coding gain is an important figure of merit to evaluate the coding efficiency of a given transform as a data compression tool. Its mathematical description can be found in [11, 13].

Another measure to evaluate the transform coding gain is the transform efficiency [11, 54]. The optimal KLT converts signals into completely uncorrelated coefficients that has transform efficiency equal to 100, whereas the DCT-II achieves a transform efficiency of 93.9911 for Markovian data at correlation coefficient of 0.95 [11].

5.2.3 Image Quality Measures in JPEG-like Compression

For quality analysis, images were submitted to a JPEG-like technique for image compression. The resulting compressed images are then assessed for image degradation in comparison to the original input image. Thus, 2-D versions of the discussed methods are required. An 8×8 image block $\mathbf{A}$ has its 2-D transform mathematically expressed by [59]:

$$\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^\top, \tag{2}$$

where $\mathbf{T}$ is a considered transformation. Input images were divided into 8×8 sub-blocks, which were submitted to the 2-D transforms. For each block, this computation furnished 64 coefficients in the approximate transform domain for a particular transformation. According to the standard zigzag sequence [60], only the $2 \leq r \leq 20$ initial coefficients in each block were retained and employed for image reconstruction [38]. This range of r corresponds to high compression. All the remaining coefficients were set to zero. The inverse procedure was then applied to reconstruct the processed image.

Subsequently, recovered images had their PSNR [55] and UQI [57] evaluated. The PSNR is a standard quality metric in the image processing literature [56], and the UQI is regarded as a more sophisticate tool for quality assessment, which takes into consideration structural similarities between images under analysis [38, 57].

This methodology was employed in [14], supported in [36, 37, 41–43], and extended in [38, 39]. However, in contrast to the JPEG-like experiments described in [36, 37, 41–43], the extended experiments considered in [38, 39] adopted the average image quality measure from a collection of representative images instead of resorting to measurements obtained from single particular images. This approach is less prone to variance effects and fortuitous input data, being more robust [61]. For the above procedure, we considered a set of 45 8-bit greyscale 512×512 standard images obtained from a public image bank [62].

Table 2: Accuracy measures of discussed methods

Method	ϵ	MSE ($\times 10^{-2}$)	C_g	η	PSNR	UQI
Exact DCT	0.000	0.000	8.826	93.991	28.336	0.733
BAS-2008 [36]	5.929	2.378	8.120	86.863	27.245	0.686
BAS-2011 [37] with $a = 0$	26.864	7.104	7.912	85.642	26.918	0.669
BAS-2011 [37] with $a = 1$	26.864	7.102	7.913	85.380	26.902	0.668
BAS-2011 [37] with $a = 2$	27.922	7.832	7.763	84.766	26.299	0.629
CB-2011 [38]	1.794	0.980	8.184	87.432	27.369	0.697
Modified CB-2011 [39]	8.659	5.939	7.333	80.897	25.224	0.563
Approximate DCT in [40]	0.870	0.621	8.344	88.059	27.567	0.701
Proposed transform	11.313	7.899	7.333	80.897	25.726	0.586

5.3 Performance Results

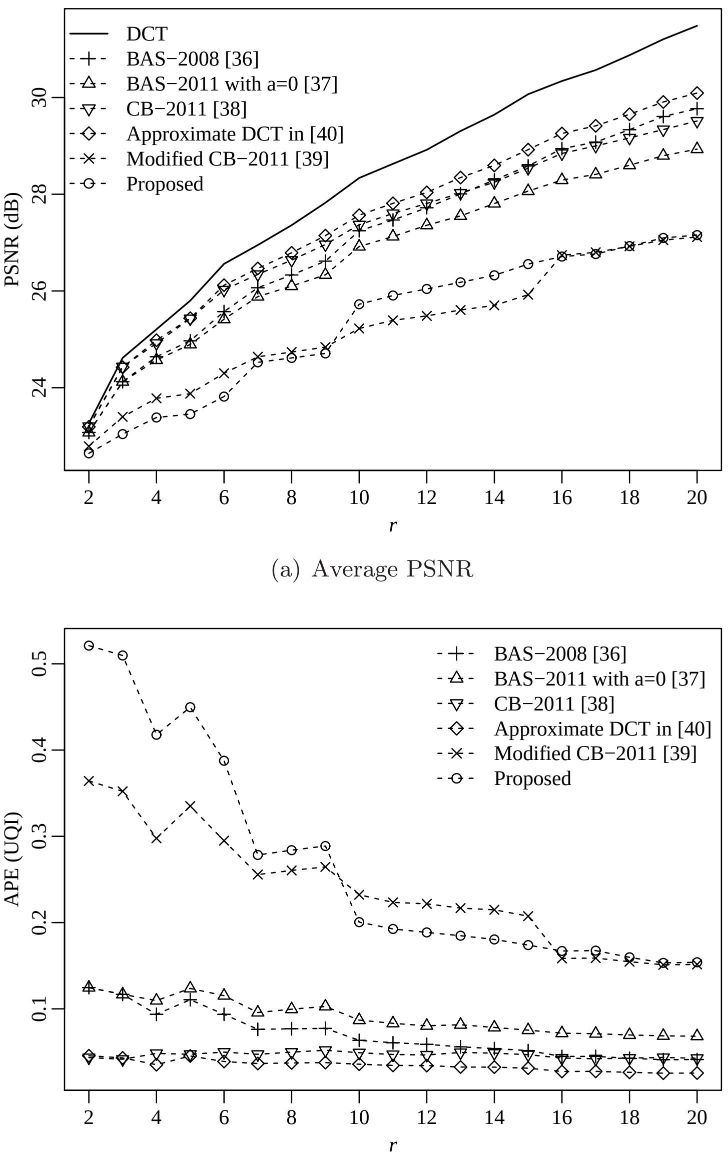
Figure 1 presents the resulting average PSNR and average UQI absolute percentage error (APE) relative to the DCT, for $r = 2, 3, \ldots, 20$, i.e., for high compression ratios [38]. The proposed transform could outperform the Modified CB-2011 approximation for $10 \leq r \leq 15$, i.e., when 84.38% to 76.56% of the DCT coefficients are discarded. Such high compression ratios are employed in several applications [41, 51–53, 63].

Table 2 shows the performance measures for the considered transforms. Average PSNR and UQI measures are presented for all considered images at a selected high compression ratio $r = 10$. The approximate transform proposed in [40] could outperform remaining methods in terms of proximity measures (total energy error and MSE) when compared to the *exact* DCT. It also furnished good image quality measure results (average PSNR = 27.567 dB). However, at the same time, it is the most expensive approximation measured by its computational cost as shown in Table 1.

On the other hand, the transforms with lowest arithmetic complexities are the Modified CB-2011 approximation and new proposed transform, both requiring only 14 additions. The new transform could outperform the Modified CB-2011 approximation as an image compression tool as indicated by the PSNR and UQI values.

A qualitative comparison based on the resulting compressed image Lena [62] obtained from the above describe procedure for $r = 10$ is shown in Fig. 2.

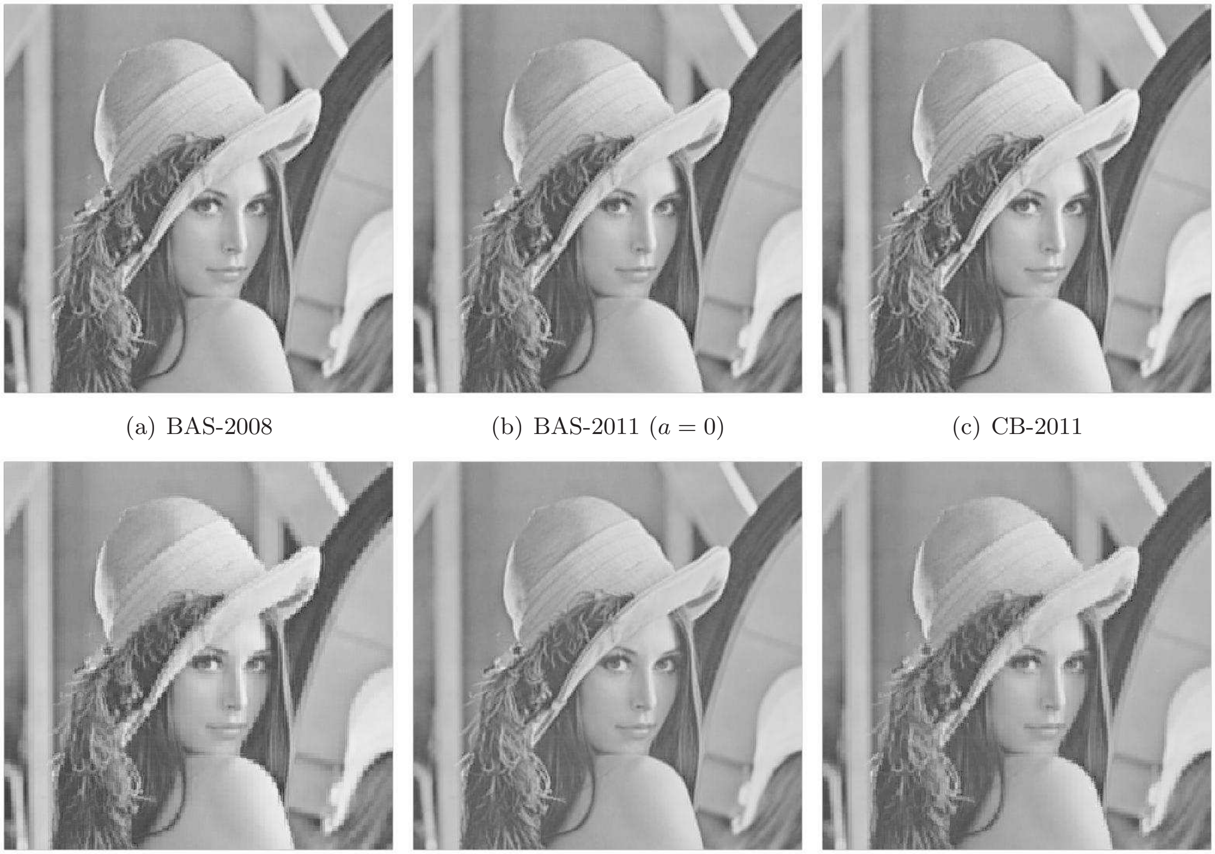
Fig. 1 and Table 2 illustrate the usual trade-off between computational complexity and performance. For instance, although BAS-2011 (for $a = 0$) could yield a better PSNR figure when compared with the proposed algorithm, it is computationally more demanding (about 14.3% more operations) and its coding gain and transform efficiency are improved in only 7.9% and 5%, respectively. In contrast, the proposed algorithm requires only 14 additions, which can lead to smaller, faster and more energy efficient circuitry designs. In the next section, we offer a comprehensive hardware analysis and comparison of the discussed algorithms with several implementation specific figures of merit.



(a) Average PSNR

(b) Average UQI absolute percentage error relative to the DCT

Figure 1: Image quality measures for several compression ratios.



(a) BAS-2008

(b) BAS-2011 ($a = 0$)

(c) CB-2011

(d) Modified CB-2011 [39]

(e) Approximate DCT in [40]

(f) Proposed transform

Figure 2: Compressed Lena image using several DCT approximations. Compression ratio is 84.375% ($r = 10$).

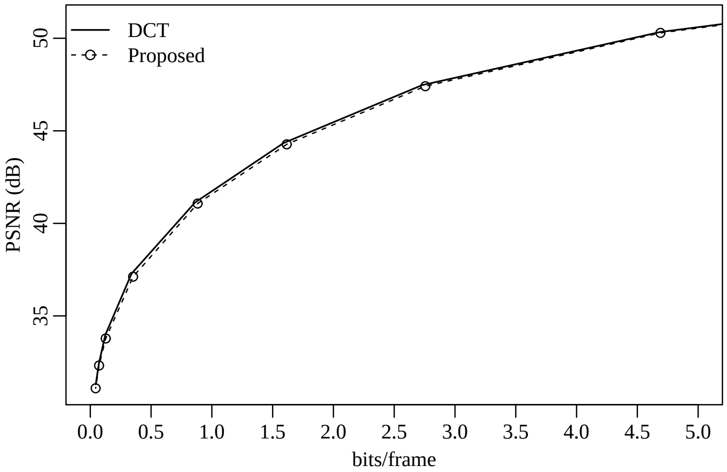


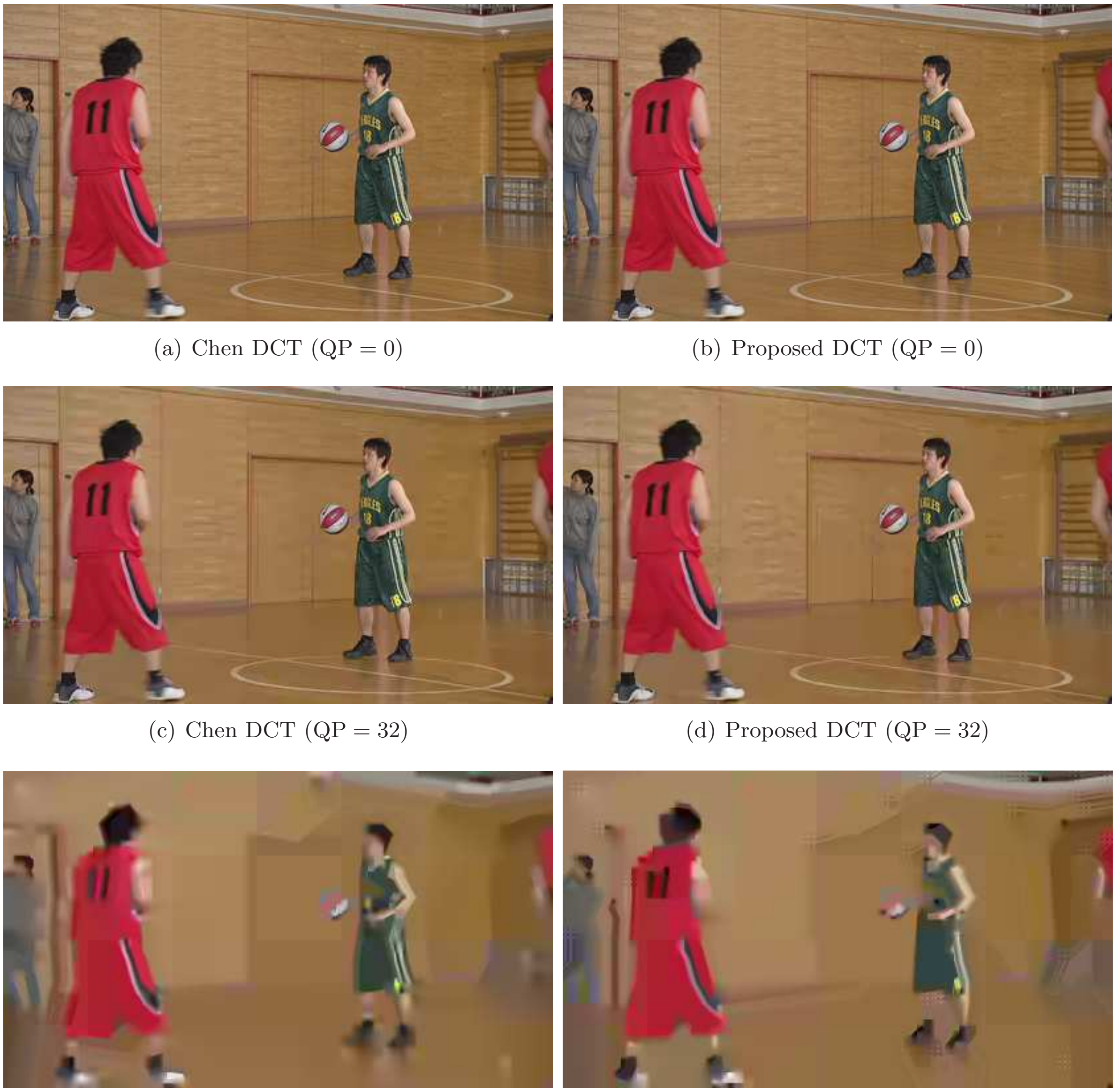
Figure 3: RD curves for 'BasketballPass' test sequence.

We also notice that although the proximity of the exact DCT—as measured by the MSE—is a good characteristic, it is not the defining property of a good DCT approximation, specially in image compression applications. A vivid example of this—seemingly counter-intuitive phenomenon—is the BAS series of DCT approximation. Such approximations possess comparatively large values of proximity measures (e.g., MSE) when compared with the exact DCT matrix. Nevertheless, they exhibit very good performance in image compression application. Results displayed in Table II illustrate this behavior.

5.4 Implementation in Real Time Video Compression Software

The proposed approximate DCT transform was embedded into an open source HEVC standard reference software [64] in order to assess its performance in real time video coding. The original integer transform prescribed in the selected HEVC reference software is a scaled approximation of Chen DCT algorithm [65], which employs 26 additions. For comparison, the proposed approximate DCT requires only 14 additions. Both algorithms were evaluated for their effect on the overall performance of the encoding process by obtaining rate-distortion (RD) curves for standard video sequences. The curves were obtained by varying the quantization point (QP) from 0 to 50 and obtaining the PSNR of the proposed approximate transform with reference to the Chen DCT implementation, which is already implemented in the reference software, along with the bits/frame of the encoded video. The PSNR computation was performed by taking the average PSNR obtained from the three channels YCbCr of the color image, as suggested in [66, p. 55]. Fig. 3 depicts the obtained RD curves for the 'BasketballPass' test sequence. Fig. 4 shows particular 416×240 frames for QP $\in \{0, 32, 50\}$ when the proposed approximate DCT and the Chen DCT are considered.

The RD curves reveals that the difference in the rate points of Chen DCT and proposed ap-



(a) Chen DCT (QP = 0)

(b) Proposed DCT (QP = 0)

(c) Chen DCT (QP = 32)

(d) Proposed DCT (QP = 32)

(e) Chen DCT (QP = 50)

(f) Proposed DCT (QP = 50)

Figure 4: Selected frames from ‘BasketballPass’ test video coded by means of the Chen DCT and the proposed DCT approximation for QP = 0 (a-b), QP = 32 (c-d), and QP = 50 (e-f).

proximation is negligible. In fact, the mean absolute difference was 0.1234 dB, which is very low. Moreover, the frames show that both encoded video streams using the above two DCT transforms are almost identical. For each QP value, the PSNR values between the resulting frames were 82.51 dB, 42.26 dB, and 36.38 dB, respectively. These very high PSNR values confirm the adequacy of the proposed scheme.

6 Digital Architectures and Realizations

In this section we propose architectures for the detailed 1-D and 2-D approximate 8-point DCT. We aim at physically implementing (2) for various transformation matrices. Introduced architectures were submitted to (i) Xilinx FPGA implementations and (ii) CMOS 45 nm application specific integrated circuit (ASIC) implementation up to the synthesis level.

This section explores the hardware utilization of the discussed algorithms while providing a comparison with the proposed novel DCT approximation algorithm and its fast algorithm realization. Our objective here is to offer digital realizations together with measured or simulated metrics of hardware resources so that better decisions on the choice of a particular fast algorithm and its implementation can be reached.

6.1 Proposed Architectures

We propose digital computer architectures that are custom designed for the real-time implementation of the fast algorithms described in Section 3. The proposed architectures employs two parallel realizations of DCT approximation blocks, as shown in Fig. 5.

The 1-D approximate DCT blocks (Fig. 5) implement a particular fast algorithm chosen from the collection described earlier in the paper. The first instantiation of the DCT block furnishes a row-wise transform computation of the input image, while the second implementation furnishes a column-wise transformation of the intermediate result. The row- and column-wise transforms can be any of the DCT approximations detailed in the paper. In other words, there is no restriction for both row- and column-wise transforms to be the same. However, for simplicity, we adopted identical transforms for both steps.

Between the approximate DCT blocks a real-time row-parallel transposition buffer circuit is required. Such block ensures data ordering for converting the row-transformed data from the first DCT approximation circuit to a transposed format as required by the column transform circuit. The transposition buffer block is detailed in Fig. 6.

The digital architectures of the discussed approximate DCT algorithms were given hardware signal flow diagrams as listed below:

1. Proposed novel algorithm and architecture shown in Fig. 7(a);

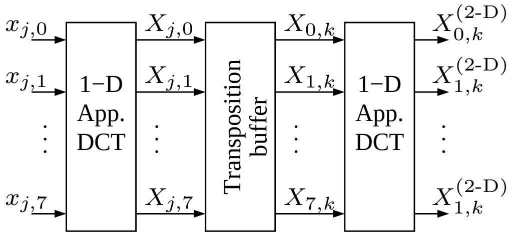


Figure 5: Two-dimensional approximate transform by means of 1-D approximate transform. Signal $x_{k,0}, x_{k,1}, \ldots$ corresponds to the rows of the input image; $X_{k,0}, X_{k,1}, \ldots$ indicates the transformed rows; $X_{0,j}, X_{1,j}, \ldots$ indicates the columns of the transposed row-wise transformed image; and $X_{0,j}^{(2\text{-D})}, X_{1,j}^{(2\text{-D})}, \ldots$ indicates the columns of the final 2-D transformed image. If $i = 0, 1, 2, 3, \ldots$, then indices j and k satisfy $j = i \pmod 8$ and $k = [(\downarrow 8)i]/8 \pmod 8$.

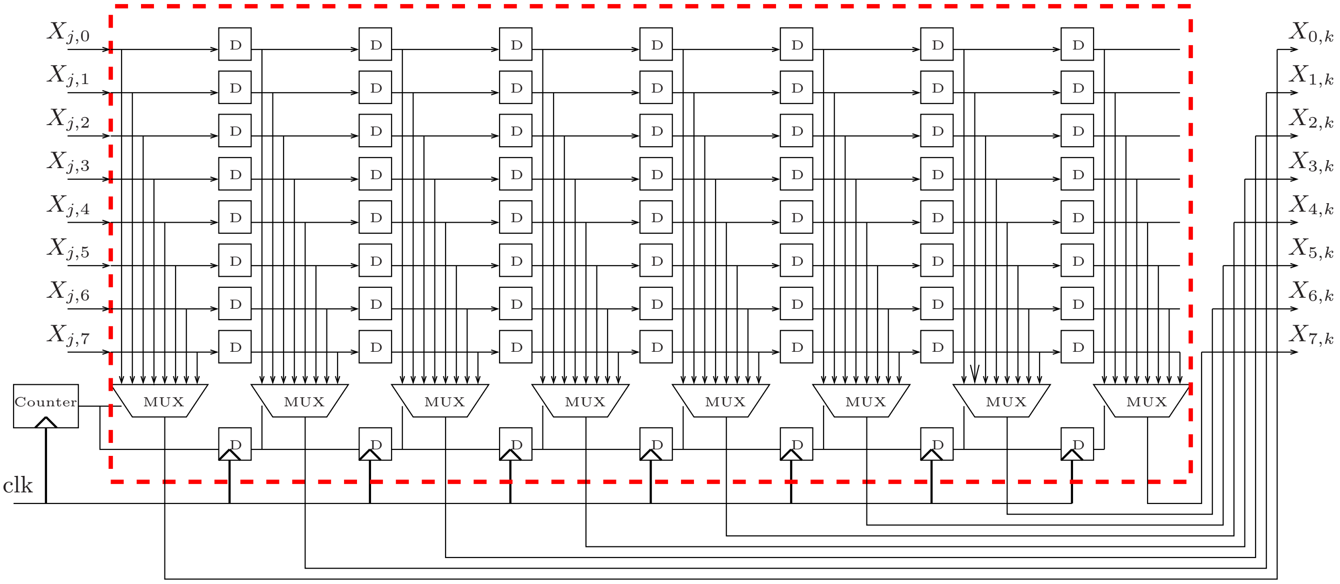


Figure 6: Details of the transposition buffer block.

2. BAS-2008 architecture shown in Fig. 7(b);
3. BAS-2011 architecture shown in Fig. 7(c);
4. CB-2011 architecture shown in Fig. 7(d);
5. Modified CB-2011 architecture shown in Fig. 7(e);
6. Architecture for the algorithm in [40] shown in Fig. 7(f).

The circuitry sections associated to the constituent matrices of the discussed factorizations are emphasized in the figures in bold or dashed boxes.

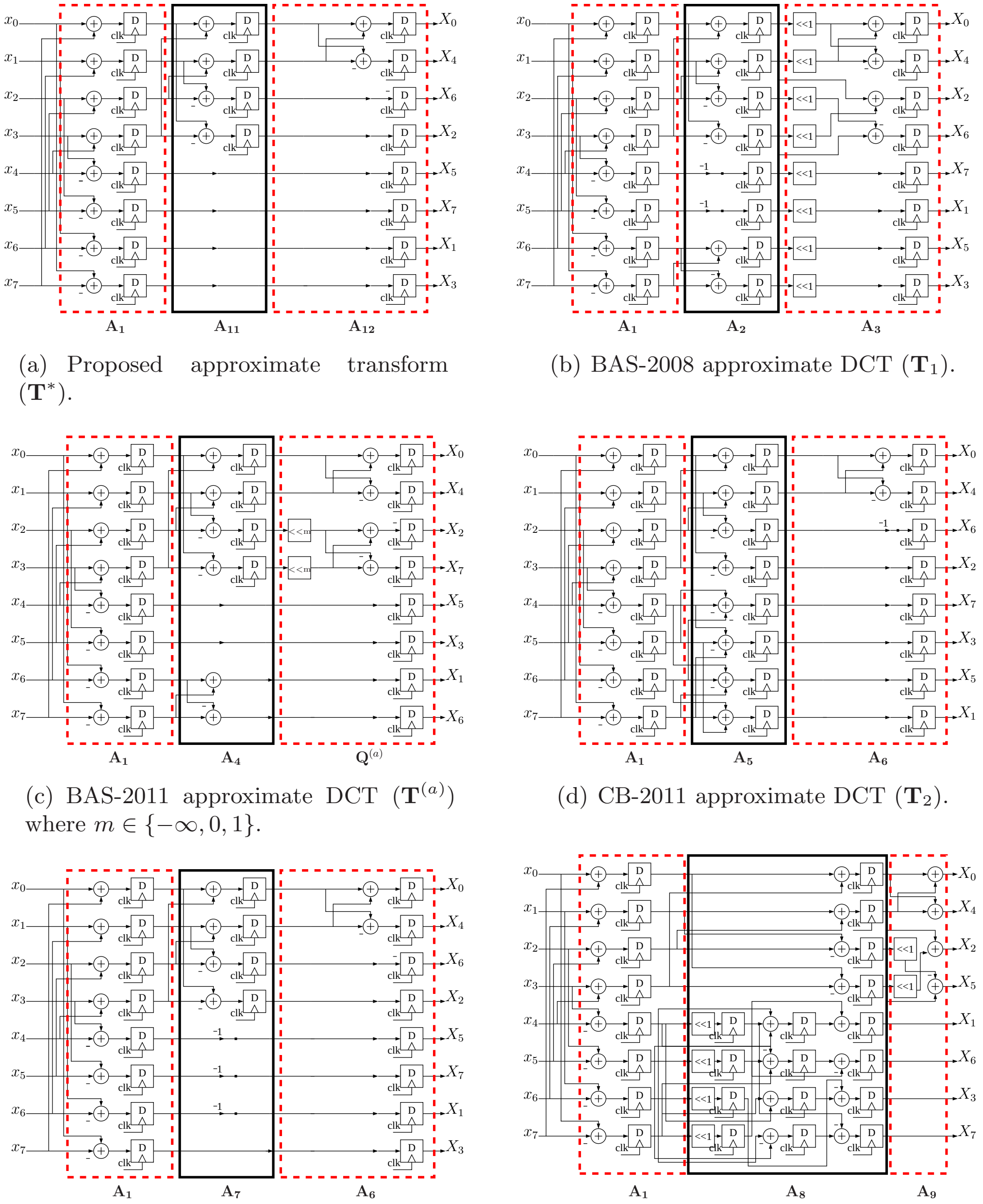
6.2 Xilinx FPGA Implementations

Discussed methods were physically realized on a FPGA based rapid prototyping system for various register sizes and tested using on-chip hardware-in-the-loop co-simulation. The architectures were designed for digital realization within the MATLAB environment using the Xilinx System Generator (XSG) with synthesis options set to generic VHDL generation. This was necessary because the auto-generated register transfer language (RTL) hardware descriptions are targeted on both FPGAs as well as custom silicon using standard cell ASIC technology.

The proposed architectures were physically realized on Xilinx Virtex-6 XC6VSX475T-2ff1156 device. The architectures were realized with fine-grain pipelining for increased throughput. Clocked registers were inserted at appropriate points within each fast algorithm in order to reduce the critical path delay as much as possible at a small cost to total area. It is expected that the additional logic overheard due to fine grain pipelining is marginal. Realizations were verified on FPGA chip using a Xilinx ML605 board at a clock frequency of 100 MHz. Measured results from the FPGA realization were achieved using stepped hardware-in-the-loop verification.

Several input precision levels were considered in order to investigate the performance in terms of digital logic resource consumptions at varied degrees of numerical accuracy and dynamic range. Adopting system word length $L \in \{4, 8, 12, 16\}$, we applied 10,000 random 8-point input test vectors using hardware co-simulation. The test vectors were generated from within the MATLAB environment and routed to the physical FPGA device using JTAG [67] based hardware co-simulation. JTAG is a digital communication standard for programming and debugging reconfigurable devices such as Xilinx FPGAs.

Then the measured data from the FPGA was routed back to MATLAB memory space. Each FPGA implementation was 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. The number of available CLBs and FFs were 297,600 and 595,200, respectively.



(a) Proposed approximate transform ($\mathbf{T}^*$).

(b) BAS-2008 approximate DCT ($\mathbf{T}_1$).

(c) BAS-2011 approximate DCT ($\mathbf{T}^{(a)}$) where $m \in \{-\infty, 0, 1\}$.

(d) CB-2011 approximate DCT ($\mathbf{T}_2$).

(e) Modified CB-2011 approximate DCT ($\mathbf{T}_3$).

(f) Approximate DCT in [40] ($\mathbf{T}_4$).

Figure 7: Digital architecture for considered DCT approximations.

Results are reported in Table 3. Quantities were obtained from the Xilinx FPGA synthesis and place-route tools by accessing the `xflow.results` report file for each run of the design flow. In addition, the static (Q_p) and dynamic power (D_p) consumptions were estimated using the Xilinx XPower Analyzer.

From Table 3 it is evident that the proposed transform and the modified CB-2011 approximation are faster than remaining approximations. Moreover, these two particular designs achieve the lowest consumption of hardware resources when compared with remaining designs.

6.3 CMOS 45 nm ASIC Implementation

The digital architectures were first designed using Xilinx System Generator tools within the Matlab/Simulink environment. Thereafter, the corresponding circuits were simulated using bit-true cycle-accurate models within the Matlab/Simulink software framework. The architectures were then converted to corresponding digital hardware description language designs using the auto-generate feature of the System Generator tool. The resulting hardware description language code led to physical implementation of the architectures using Xilinx FPGA technology, which in turn led to extensive hardware co-simulation on FPGA chip. Hardware co-simulation was used for verification of the hardware description language designs which were contained in register transfer language (RTL) libraries. Thus, the above mentioned verified RTL code for each of the 2-D architectures was ported to the Cadence RTL Compiler environment for mapping to application specific CMOS technology. To guarantee that the auto-generated RTL could seamlessly compile in the CMOS design environment, we ensured that RTL code followed a behavioral description which did not contain any FPGA specific (vendor specific) instructions. By adopting standard IEEE 1164 libraries and behavioral RTL, the resulting code was compatible with Cadence Encounter for CMOS standard cell synthesis.

For this purpose, we used FreePDK, a free open-source ASIC standard cell library at the 45 nm node [68]. The supply voltage of the CMOS realization was fixed at $V_{\text{DD}} = 1.1$ V during estimation of power consumption and logic delay. The adopted figures of merit for the ASIC synthesis were: area (A) in mm^2, critical path delay (T) in ns, area-time complexity (AT) in mm$^2 \cdot$ ns, dynamic power consumption in watts, and area-time-squared complexity (AT^2) in mm$^2 \cdot$ ns^2. Results are displayed in Table 4 and 5.

The AT complexity is an adequate metric when the chip area is more relevant than speed or computational throughput. On the other hand, AT^2 is employed when real-time speed is the most important driving force for the optimizations in the logic designs. In all cases, clear improvements in maximum real-time clock frequency is predicted over the same RTL targeted at FPGA technology.

Table 3: Hardware Resource Consumption using Xilinx Virtex-6 XC6VSX475T-2ff1156 device

L	CLB	FF	Q_p (W)	D_p (W)	T_{cpd}	$F_{\max}$
BAS-2008 Algorithm						
4	395	784	5.154	0.918	2.350	401.7
8	613	1123	5.168	1.105	2.573	367.1
12	821	1523	5.184	1.301	2.930	337.8
16	1029	1915	5.187	1.344	3.254	284.0
BAS-2011 for $a = 0$						
4	335	877	5.142	0.767	2.340	386.4
8	535	1276	5.161	1.015	2.600	356.2
12	728	1732	5.180	1.260	2.822	337.4
16	919	2187	5.198	1.486	2.981	325.2
BAS-2011 for $a = 1$						
4	387	1019	5.146	0.811	2.413	396.7
8	605	1453	5.165	1.065	2.513	361.4
12	813	1949	5.179	1.247	2.962	329.4
16	1021	2445	5.198	1.483	2.987	316.9
BAS-2011 for $a = 2$						
4	385	1019	5.146	0.818	2.371	402.9
8	603	1453	5.163	1.042	2.584	364.7
12	812	1950	5.190	1.378	2.618	353.1
16	1019	2445	5.201	1.527	3.006	326.5
CB-2011 Algorithm						
4	452	883	5.141	0.750	2.518	363.4
8	702	1257	5.151	0.876	3.065	303.1
12	950	1709	5.162	1.029	3.466	270.6
16	1198	2162	5.187	1.341	3.610	256.0
Approximate DCT in [40]						
4	513	1040	5.158	0.972	2.545	387.8
8	779	1471	5.173	1.170	2.769	351.0
12	1036	1968	5.181	1.262	2.945	314.9
16	1291	2463	5.200	1.514	3.205	298.0
Modified CB-2011 approximation						
4	297	652	5.153	0.903	2.384	399.7
8	481	961	5.177	1.214	2.523	391.2
12	657	1329	5.191	1.390	2.693	354.0
16	834	1698	5.219	1.752	2.829	345.5
Proposed Transform						
4	303	651	5.146	0.818	2.344	404.0
8	487	963	5.167	1.092	2.470	385.1
12	663	1329	5.185	1.322	2.524	353.7
16	839	1697	5.203	1.551	2.818	341.8

Table 4: Hardware resource consumption for CMOS 45nm ASIC implementation

L	ASIC Gates	Area	T_{cpd}	AT	AT^2	$F_{\max}$
BAS-2008 Algorithm						
4	27792	0.123	1.140	0.140	0.160	877.2
8	44654	0.192	1.204	0.231	0.278	830.6
12	61388	0.262	1.216	0.319	0.388	822.4
16	78281	0.332	1.236	0.411	0.508	809.1
BAS-2011 for $a = 0$						
4	26299	0.114	1.135	0.129	0.147	881.1
8	42313	0.182	1.147	0.209	0.239	871.8
12	58342	0.250	1.225	0.306	0.375	816.3
16	74062	0.317	1.310	0.415	0.544	763.4
BAS-2011 for $a = 1$						
4	25940	0.108	1.106	0.120	0.133	904.2
8	40330	0.166	1.125	0.187	0.210	888.9
12	53728	0.225	1.170	0.263	0.308	854.7
16	67860	0.283	1.200	0.339	0.407	833.3
BAS-2011 for $a = 2$						
4	25554	0.109	1.117	0.122	0.136	895.3
8	39321	0.167	1.132	0.189	0.214	883.4
12	53950	0.226	1.175	0.265	0.312	851.1
16	67979	0.284	1.201	0.341	0.409	832.6
CB-2011 Algorithm						
4	30319	0.132	1.167	0.154	0.180	856.9
8	48556	0.209	1.192	0.249	0.296	838.9
12	66956	0.285	1.221	0.348	0.425	819.0
16	85873	0.363	1.240	0.450	0.558	806.5
Approximate DCT in [40]						
4	35141	0.151	1.141	0.173	0.197	876.4
8	53624	0.230	1.211	0.278	0.337	825.8
12	73224	0.310	1.234	0.383	0.473	810.4
16	92697	0.391	1.242	0.486	0.603	805.2
Modified CB-2011 Approximation						
4	24777	0.107	1.105	0.119	0.131	905.0
8	40746	0.175	1.128	0.197	0.222	886.5
12	56644	0.242	1.164	0.282	0.328	859.1
16	73702	0.314	1.177	0.369	0.434	849.6
Proposed Transform						
4	24817	0.107	1.110	0.119	0.132	900.9
8	40705	0.175	1.129	0.197	0.223	885.7
12	56703	0.242	1.165	0.282	0.329	858.4
16	73906	0.314	1.174	0.368	0.432	851.8

Table 5: Power consumption for CMOS 45nm ASIC implementation

L	Q_p(mW)	D_p(W)	L	Q_p(mW)	D_p(W)
BAS-2008 Algorithm			CB-2011 Algorithm		
4	1.00	0.18	4	1.08	0.24
8	1.56	0.33	8	1.706	0.36
12	2.13	0.43	12	2.32	0.48
16	2.70	0.55	16	2.94	0.60
BAS-2011 for $a = 0$			Approximate DCT in [40]		
4	0.94	0.21	4	1.23	0.28
8	1.48	0.33	8	1.87	0.39
12	2.04	0.42	12	2.52	0.52
16	2.59	0.50	16	3.17	0.64
BAS-2011 for $a = 1$			Modified CB-2011		
4	0.88	0.24	4	0.88	0.20
8	1.34	0.36	8	1.42	0.32
12	1.81	0.40	12	1.98	0.43
16	2.28	0.48	16	2.55	0.55
BAS-2011 for $a = 2$			Proposed Transform		
4	0.89	0.20	4	0.88	0.20
8	1.35	0.30	8	1.42	0.32
12	1.82	0.39	12	1.98	0.43
16	2.29	0.48	16	2.55	0.55

7 Conclusion

In this paper, we proposed (i) a novel low-power 8-point DCT approximation that require only 14 addition operations to computations and (ii) hardware implementation for the proposed transform and several other prominent approximate DCT methods, including the designs by Bouguezel-Ahmad-Swamy. We obtained that all considered approximate transforms perform very close to the ideal DCT. However, the modified CB-2011 approximation and the proposed transform possess lower computational complexity and are faster than all other approximations under consideration. In terms of image compression, the proposed transform could outperform the modified CB-2011 algorithm. Hence the new proposed transform is the best approximation for the DCT in terms of computational complexity and speed among the approximate transform examined.

Introduced implementations address both 1-D and 2-D approximate DCT. All the approximations were digitally implemented using both Xilinx FPGA tools and CMOS 45 nm ASIC technology. The speeds of operation were much greater using the CMOS technology for the same function word size. Therefore, the proposed architectures are suitable for image and video processing, being candidates for improvements in several standards including the HEVC.

Future work includes replacing the FreePDK standard cells with highly optimized proprietary digital libraries from TSMC PDK [68] and continuing the CMOS realization all the way up to chip fabrication and post-fab test on a measurement system. Additionally, we intend to develop the approximate versions for the 4-, 16-, and 32-point DCT as well as to the 4-point discrete sine transform, which are discrete transforms required by HEVC.

8 Acknowledgments

This work was supported by the University of Akron, Ohio, USA; the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CPNq) and FACEPE, Brazil; and the Natural Science and Engineering Research Council (NSERC), Canada.

References

[1] H.-Y. Lin and W.-Z. Chang, "High dynamic range imaging for stereoscopic scene representation," in *16th IEEE International Conference on Image Processing (ICIP), 2009*, Nov. 2009, pp. 4305–4308.

[2] M. Rezaei, S. Wenger, and M. Gabbouj, "Video rate control for streaming and local recording optimized for mobile devices," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2005*, vol. 4, Sep. 2005, pp. 2284–2288.

[3] H. Zheng and J. Boyce, "Packet coding schemes for MPEG video over internet and wireless networks," in *IEEE Wireless Communications and Networking Confernce (WCNC), 2000*, vol. 1, 2000, pp. 191–195.

[4] E. Magli and D. Taubman, "Image compression practices and standards for geospatial information systems," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2003*, vol. 1, Jul. 2003, pp. 654–656.

[5] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, "Real-time video analysis on an embedded smart camera for traffic surveillance," in *10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004*, May 2004, pp. 174–181.

[6] S. Marsi, G. Impoco, S. C. A. Ukovich, and G. Ramponi, "Video enhancement and dynamic range control of HDR sequences for automotive applications," *Advances in Signal Processing*, vol. 2007, no. 80971, pp. 1–9, Jun. 2007.

[7] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer and Telecommunications Networking*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[8] A. Madanayake, R. J. Cintra, D. Onen, V. S. Dimitrov, N. Rajapaksha, L. T. Bruton, and A. Edirisuriya, "A Row-Parallel 8×8 2-D DCT Architecture Using Algebraic Integer-Based

Exact Computation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 915–929, Jun. 2012.

[9] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[10] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* San Diego, CA: Academic Press, 1990.

[11] V. Britanak, P. Yip, and K. R. Rao, *Discrete Cosine and Sine Transforms.* Academic Press, 2007.

[12] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards.* Boston: Kluwer Academic Publishers, 1997.

[13] J. Liang and T. D. Tran, "Fast multiplierless approximation of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[14] T. I. Haweel, "A new square wave transform based on the DCT," *Signal Processing*, vol. 81, no. 11, pp. 2309–2319, Nov. 2001.

[15] K. Lengwehasatit and A. Ortega, "Scalable variable complexity approximate forward DCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1236–1248, Nov. 2004.

[16] R. J. Clarke, "Relation between the Karhunen-Loève and cosine transforms," *IEE Proceedings F Communications, Radar and Signal Processing*, vol. 128, no. 6, pp. 359–360, Nov. 1981.

[17] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard.* New York, NY: Van Nostrand Reinhold, 1992.

[18] N. Roma and L. Sousa, "Efficient hybrid DCT-domain algorithm for video spatial downscaling," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 2, pp. 30–30, 2007.

[19] International Organisation for Standardisation, "Generic coding of moving pictures and associated audio information – Part 2: Video," ISO, ISO/IEC JTC1/SC29/WG11 - Coding of Moving Pictures and Audio, 1994.

[20] International Telecommunication Union, "ITU-T recommendation H.261 version 1: Video codec for audiovisual services at $p \times 64$ kbits," ITU-T, Tech. Rep., 1990.

[21] ——, "ITU-T recommendation H.263 version 1: Video coding for low bit rate communication," ITU-T, Tech. Rep., 1995.

[22] H. L. P. A. Madanayake, R. J. Cintra, D. Onen, V. S. Dimitrov, and L. T. Bruton, "Algebraic integer based 8×8 2-D DCT architecture for digital video processing," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2011*, May 2011, pp. 1247–1250.

[23] International Telecommunication Union, "ITU-T recommendarion H.264 version 1: Advanced video coding for generic audio-visual services,," ITU-T, Tech. Rep., 2003.

[24] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[25] M. T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?" *IEEE Consumer Electronics Magazine*, vol. 1, no. 3, pp. 36–46, Jul. 2012.

[26] J.-S. Park, W.-J. Nam, S.-M. Han, and S. Lee, "2-D large inverse transform (16x16, 32x32) for HEVC (High Efficiency Video Coding)," *Journal of Semiconductor Technology and Science*, vol. 2, pp. 203–211, 2012.

[27] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. W. it, "Comparison of the coding efficiency of video coding standards - including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[28] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[29] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, pp. 803–816, Aug. 1984.

[30] B. G. Lee, "A new algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, pp. 1243–1245, Dec. 1984.

[31] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, vol. 6, pp. 267–278, Aug. 1984.

[32] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 10, pp. 1455–1461, Oct. 1987.

[33] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Transactions of the IEICE*, vol. E-71, no. 11, pp. 1095–1097, Nov. 1988.

[34] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1D DCT algorithms with 11 multiplications," in *International Conference on Acoustics, Speech, and Signal Processing*, May 1989, pp. 988–991.

[35] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2174–2193, Sep. 1992.

[36] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-Complexity 8×8 Transform for Image Compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249–1250, 2008.

[37] ——, "A low-complexity parametric transform for image compression," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2011*, May 2011, pp. 2145–2148.

[38] R. J. Cintra and F. M. Bayer, "A DCT approximation for image compression," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 579–582, Oct. 2011.

[39] F. M. Bayer and R. J. Cintra, "DCT-like transform for image compression requires 14 additions only," *Electronics Letters*, vol. 48, no. 15, pp. 919–921, 2012.

[40] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, and N. Rajapaksha, "Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing," *Measurement Science and Technology*, vol. 23, no. 11, pp. 1–15, Nov. 2012.

[41] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A multiplication-free transform for image compression," in *2nd International Conference on Signals, Circuits and Systems, 2008*, Nov. 2008, pp. 1–4.

[42] ——, "A fast 8×8 transform for image compression," in *Proceedings of the Internation Conference on Microelectronics, 2009*, Dec. 2009.

[43] ——, "A novel transform for image compression," in *53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2010*, Aug. 2010, pp. 509–512.

[44] F. M. Bayer and R. J. Cintra, "Image compression via a fast DCT approximation," *Revista IEEE América Latina*, vol. 8, no. 6, pp. 708–713, Dec. 2010.

[45] F. M. Bayer, R. J. Cintra, A. Edirisuriya, and A. Madanayake, "A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications," *Measurement Science and Technology*, vol. 23, no. 11, 2012.

[46] A. Edirisuriya, A. Madanayake, R. J. Cintra, and F. M. Bayer, "A multiplication-free digital architecture for 16x16 2-D DCT/DST transform for HEVC," in *IEEE 27th Convention of Electrical Electronics Engineers in Israel, 2012*, Nov. 2012, pp. 1–5.

[47] M. Martuza and K. Wahid, "A cost effective implementation of 8×8 transform of HEVC from H.264/AVC," in *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, Apr. 2012, pp. 1–4.

[48] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, *IEEE Trans. Circuits Syst. Video Technol.*, no. 12, pp. 1669 –1684, Dec.

[49] I. N. Herstein, *Topics in Algebra*, 2nd ed., W. India, Ed. John Wiley & Sons, 1975.

[50] R. J. Cintra, "An integer approximation method for discrete sinusoidal transforms," *Journal of Circuits, Systems, and Signal Processing*, vol. 30, no. 6, pp. 1481–1501, 2011.

[51] J. Robinson and V. Kecman, "Combining support vector machine learning with the discrete cosine transform in image compression," *IEEE Trans. Neural Networks.*, vol. 14, pp. 950–958, 2003.

[52] Z. Pan and H. Bolouri, "High speed face recognition based on discrete cosine transforms and neural networks," Science and Technology Research Centre, University of Hertfordshire, Tech. Rep., 1999.

[53] Z. Pan, R. Adams, and H. Bolouri, "Image recognition using Discrete Cosine Transforms as dimensionality reduction," in *IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, 2001.

[54] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *IEE Proceedings I Communications, Speech and Vision*, vol. 136, no. 4, pp. 276–282, Aug. 1989.

[55] D. Salomon, *Data compression: The complete reference*, 4th ed. Springer, 2007.

[56] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, pp. 35–48, 2012.

[57] Z. Wang and A. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[58] R. Piessens, E. deDoncker-Kapenga, C. Uberhuber, and D. Kahaner, *Quadpack: a Subroutine Package for Automatic Integration.* Springer-Verlag, 1983.

[59] T. Suzuki and M. Ikehara, "Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2958–2965, Nov. 2010.

[60] I.-M. Pao and M.-T. Sun, "Approximation of calculations for forward discrete cosine transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 264–268, Jun. 1998.

[61] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, ser. Prentie Hall Signal Processing Series, A. V. Oppenheim, Ed. Upper Saddle River, NJ: Prentice Hall, 1993, vol. 1.

[62] "The USC-SIPI image database," http://sipi.usc.edu/database/, 2012, University of Southern California, Signal and Image Processing Institute. [Online]. Available: http://sipi.usc.edu/database/

[63] G. Mandyam, N. Ahmed, and N. Magotra, "Lossless image compression using the discrete cosine transform," *Journal of Visual Communication and Image Representation*, vol. 8, no. 1, pp. 21–26, 1997.

[64] Joint Collaborative Team on Video Coding (JCT-VC), "HEVC references oftware documentation," 2013, Fraunhofer Heinrich Hertz Institute. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[65] W.-H. Chen, C. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *Communications, IEEE Transactions on*, vol. 25, no. 9, pp. 1004–1009, 1977.

[66] N. Semary, *Image coloring Techniques and Applications.* GRIN Verlag, 2012.

[67] D. Rahul, *Introduction to Embedded System Design Using Field Programmable Gate Arrays.* Springer-Verlag, 2009.

[68] J. E. Stine, I. Castellanos, M. Wood, J. Henson, F. Love, W. R. Davis, P. D. Franzon, M. Bucher, S. Basavarajaiah, J. Oh, and R. Jenkal, "FreePDK: An open-source variation-aware design kit," in *IEEE International Conference on Microelectronic Systems Education, 2007*, Jun. 2007, pp. 173–174.