Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, Biomedical, and Smart Antenna Applications

Arjuna Madanayake, Renato J. Cintra, Vassil Dimitrov, Fábio Mariano Bayer, Khan A. Wahid, Sunera Kulasekera, Amila Edirisuriya, Uma Sadhvi Potluri, Shiva Kumar Madishetty, and Nilanka Rajapaksha

Abstract

The DCT and the DWT are used in a number of emerging DSP applications, such as, HD video compression, biomedical imaging, and smart antenna beamformers for wireless communications and radar. Of late, there has been much interest on fast algorithms for the computation of the above transforms using multiplier-free approximations because they result in low power and low complexity systems. Approximate methods rely on the trade-off of accuracy for lower power and/or circuit complexity/chip-area. This paper provides a detailed review of VLSI architectures and CAS implementations for both DCT/DWTs, which can be designed either for higher-accuracy or for low-power consumption. This article covers both recent theoretical advancements on discrete transforms in addition to an overview of existing VLSI architectures. The paper also discusses error free VLSI architectures that provides high accuracy systems and approximate architectures that offer high computational gain making them highly attractive for real-world applications that are subject to constraints in both chip-area as well as power. The methods discussed in the paper can be used in the design of emerging low-power digital systems having lowest complexity at the cost of a loss in accuracy—the optimal trade-off of computational accuracy for lowest possible complexity and power. A complete synopsis of available techniques, algorithms and FPGA/VLSI realizations are discussed in the paper.

Digital Object Identifier 10.1109/MCAS.2014.2385553
Date of publication: 17 February 2015

I. Introduction

Explosive growth in the digital VLSI sector, especially in deep submicron CMOS nanoelectronics, is fueled by new innovations in physics at the device level with technological advancements such as multi-gate FinFETs, space-time FPGA fabrics, 3-D VLSI, deep subthreshold low-power devices, and asynchronous logic being driving factors, to name a few. In parallel to technology developments, systems performance is rapidly improving at the algorithms level by sophisticate designs which increasingly take into non-conventional mathematical techniques and optimization schemes that are specifically tailored for high-performance digital VLSI devices. In particular, the application of number theoretic and approximation techniques for the design of fast algorithms having lower complexity, lower power consumption, and/or higher accuracy is of critical interest [1]–[3]. In this review article, we concentrate on a key aspects of this vast area of research and offer a synopsis of our recent work pertaining to (i) highly-accurate/exact computation of mathematical transforms [1]; (ii) multiplier-less implementation of approximate fast algorithms for the realization of image/video processing functions [4], [5]; (iii) the application of multiplier-less fast trigonometric algorithms in new application domains especially at radio-frequencies leading to new capabilities for imaging and wireless communications [5], (iv) the VLSI realization of number theoretic transforms with emerging applications in secure image processing [2]–[4], and (v) the application of low-complexity high-accuracy algorithms for power efficient imaging with applications in biomedical signal processing [3], [6], [7]. Examples are shown in Fig. 1.

Of the many different trigonometric and wavelet transforms used in signal processing, the DCT plays an especially important role for image and video compression applications [1], [2], [4], [5]. For instance, in video over Internet protocol (Video over IP), which is one of the fastest growing sectors of the global Internet, the DCT and it variants are highly important tools providing low-power multimedia devices with increasing better video quality.

Because the DCT is a widely used mathematical tool in image and video compression, it follows that it is a core component in contemporary media standards such as JPEG and MPEG [8]. Indeed, the DCT is known for its properties of decorrelation, energy compaction, separability, symmetry, and orthogonality [9]. The DCT enjoys its prominent role in compression because of its excellent energy compaction property, which is very close to the statistically optimal Karhunen–Loève transform. Low-complexity approximations for the DCT, whether this is achieved via integer approximations or other fast algorithms, are of immense importance in the digital video industry.

II. Discrete Transforms

A. Discrete Cosine Transform

The discrete transforms is an essential tool in digital signal processing [10], [11]. In particular, the DCT is a major tool in image processing. This is mainly because the DCT is asymptotically equivalent to the Karhunen-Loève transform (KLT), which possesses optimal decorrelation and energy compaction properties [10]–[15]. Such good characteristics are attained when high correlated first-order Markov signals are considered [10], [11]. Importantly natural images belong to this particular class of signals [14]. In particular, the DCT has been adopted in several image and video coding schemes [16], such as JPEG [17],

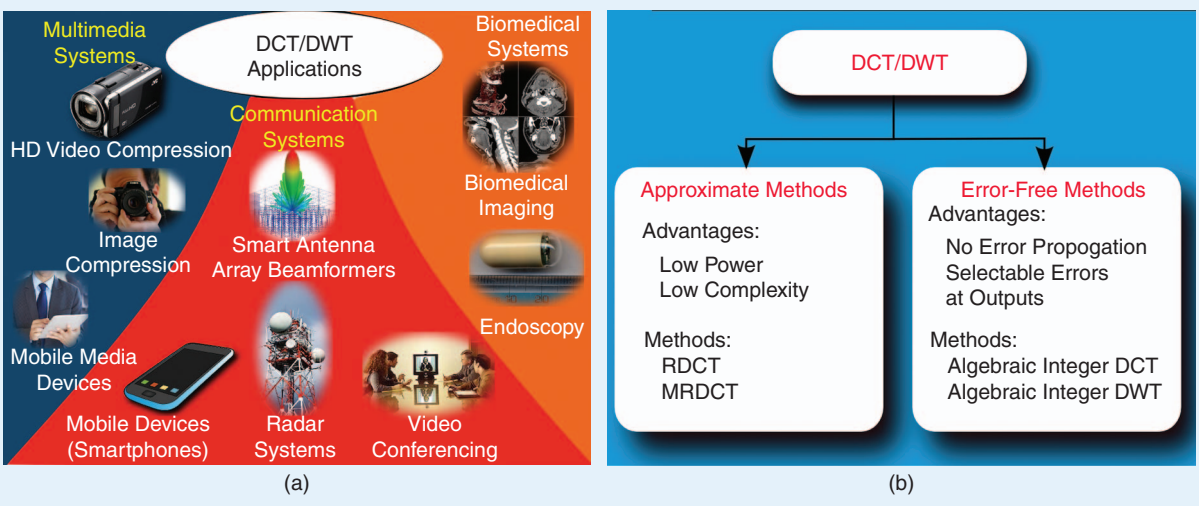

Figure 1. (a) Applications of the DCT and DWT in various domains, and (b) The approximate methods and error-free architectures for the DCT and DWT transforms.

MPEG-1 [18], MPEG-2 [19], H.261 [20], H.263 [21], H.264 [22], and the recent HEVC [23]. For instance, H.264 and HEVC standards employ 8-point DCT algorithms [23]–[30] among other different blocklenghts, such as 4, 16, and 32 points [31], [32]. In [33], the 8-point DCT stage of the HECV was optimized. Because of this increasing demand, several algorithms for the efficient computation of the 8-point DCT have been proposed [34], [35]. In [10], [11], comprehensive surveys on DCT algorithms are detailed.

Noteworthy DCT methods include the following procedures: Wang factorization [36], Lee DCT for power-of-two block lengths [37], Arai DCT scheme [38], Loeffler algorithm [39], and Feig-Winograd factorization [40]. All these algorithms are classical results in the field and have been considered for practical applications [18], [41], [42]. For instance, the Arai DCT scheme was employed in various recent hardware implementations of the DCT [1], [2], [43].

The N-point DCT is algebraically represented by the $N\times N$ transformation matrix $\boldsymbol{C}_N$ whose elements are given by [10], [11]

$$c_{m,n}=\frac{1}{\sqrt{N}}\beta_{m-1}\cos\left(\frac{\pi(m-1)(2n-1)}{2N}\right),$$

where $m,n=1,2,\ldots,N$, $\beta_0=1$, and $\beta_k=\sqrt{2}$, for $k\neq 0$. Let $\boldsymbol{x}=[x_0\ \ x_1\ \ \cdots\ \ x_{N-1}]^\top$ be an input vector, where the superscript $^\top$ denotes the transposition operation. The one-dimensional (1-D) DCT transform of $\boldsymbol{x}$ is the N-point vector $\boldsymbol{X}=[X_0\ \ X_1\ \ \cdots\ \ X_{N-1}]^\top$ given by $\boldsymbol{X}=\boldsymbol{C}_N\cdot\boldsymbol{x}$. Because $\boldsymbol{C}_N$ is an orthogonal matrix, the inverse transformation can be written according to $\boldsymbol{x}=\boldsymbol{C}_N^\top\cdot\boldsymbol{X}$. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be square matrices of size N. For two-dimensional (2-D) signals, we have the following expressions that relate the forward and inverse 2-D DCT operations, respectively:

$$\boldsymbol{B}=\boldsymbol{C}_N\cdot\boldsymbol{A}\cdot\boldsymbol{C}_N^\top\ \text{ and }\ \boldsymbol{A}=\boldsymbol{C}_N^\top\cdot\boldsymbol{B}\cdot\boldsymbol{C}_N. \tag{1}$$

Although the procedures described in this work can be applied to any blocklength, we focus exclusively on the 8-point DCT. Thus, for simplicity, the 8-point DCT matrix is denoted as $\boldsymbol{C}$ and is given by:

$$\boldsymbol{C}\triangleq\boldsymbol{C}_8=\frac{1}{2}\cdot\begin{bmatrix}\gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3\\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0\\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1\\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2\\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3\\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4\\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5\\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6\end{bmatrix},$$

where $\gamma_k=\cos(2\pi(k+1)/32)$, $k=0,1,\ldots,6$. These quantities are algebraic integers explicitly given by [44]:

$$\gamma_0=\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}\approx 0.9808\ldots,$$
$$\gamma_1=\frac{\sqrt{2+\sqrt{2}}}{2}\approx 0.9239\ldots,$$
$$\gamma_2=\frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}\approx 0.8315\ldots,$$
$$\gamma_3=\frac{\sqrt{2}}{2}\approx 0.7071\ldots,$$
$$\gamma_4=\frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}\approx 0.5556\ldots,$$
$$\gamma_5=\frac{\sqrt{2-\sqrt{2}}}{2}\approx 0.3827\ldots,$$
$$\gamma_6=\frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}\approx 0.1951\ldots$$

In this work, we adopt the following terminology. A matrix $\boldsymbol{A}$ is orthogonal if $\boldsymbol{A}\cdot\boldsymbol{A}^\top$ is a diagonal matrix. In particular, if $\boldsymbol{A}\cdot\boldsymbol{A}^\top$ is the identity matrix, then $\boldsymbol{A}$ is said to be orthonormal.

B. Discrete Wavelet Transform

In recent years, there has been a great deal of interest in wavelet transforms, especially in the field of image and video processing [45], [46]. Wavelet transforms are also a requirement for image compression in the JPEG 2000 standard [47]. There have been several wavelet filters proposed for applications in image processing, but in this paper we will restrict ourselves to Daubechies wavelets. This class of wavelets includes members ranging from highly localized

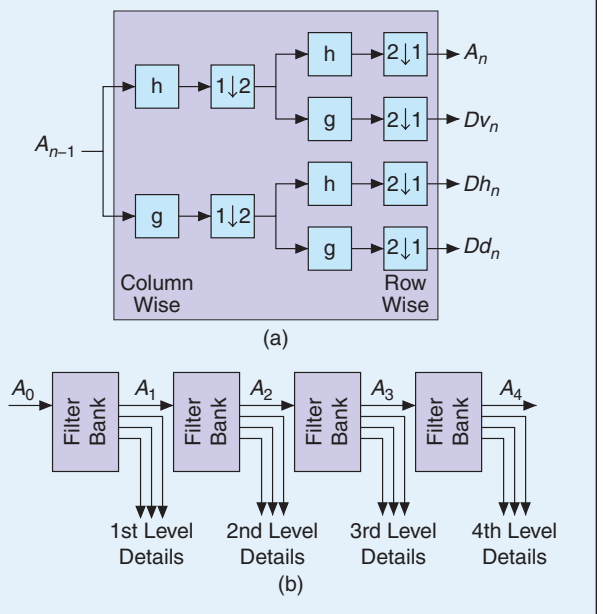

Figure 2. (a) Diagram of a single application of the 2-D wavelet filter bank, (b) Recursive application of the 2-D wavelet filter bank.

to highly smooth and can provide excellent performance in image compression applications [48].

Wavelet decomposition of input image data can be accomplished by sub-band coding. A 2-D finite impulse response (FIR) filter bank processes the input data resulting in an approximation and detail sub-images. The input image $\boldsymbol{A}_{n-1}$ is of resolution $N \times N$ pixels; and it is input to a pair of low-pass (approximation) and high-pass (detail) filters $\boldsymbol{h}$ and $\boldsymbol{g}$, respectively. The filters operate column-wise on the image followed by dyadic down-sampling, i.e., only one of every two columns are retained. Then the same process is applied row-wise. The outputs are four sub-images $\boldsymbol{A}_n$, $\boldsymbol{Dv}_n$, $\boldsymbol{Dh}_n$, and $\boldsymbol{Dd}_n$, which represent the 2-D wavelet coefficients for the coarse approximation, vertical details, horizontal details, and diagonal details, respectively. This process is shown in Fig. 2 (a) for one-level wavelet analysis via filter banks. Symbols $2 \downarrow 1$ and $1 \downarrow 2$ are used to denote the column-wise and row-wise down-sampling. respectively [49, pp. 6-26]. The resultant sub-images are all of size $N/2 \times N/2$, because of dyadic down-sampling.

These operations can be performed recursively [50]. The resulting approximation $\boldsymbol{A}_n$ can be re-submitted to the signal flow architecture shown in Fig. 2. As a result, after each iteration a coarser approximation can be achieved. Let the original image to be analysed be denoted by $\boldsymbol{A}_0$. The topmost branch of the signal flow shown in Fig. 2 (a) computes the approximation data. Detail data $\boldsymbol{Dv}_n$, $\boldsymbol{Dh}_n$, and $\boldsymbol{Dd}_n$ are normally discarded or thresholded in data compression applications [50]. Fig. 2 (b) shows the pyramidal diagram of the multi-level wavelet analysis. After each set of filter banks, a coarser approximation $\boldsymbol{A}_n, n = 1, 2, 3, 4,$ is furnished. Each level also produces the detail information. In this work, we focus on the computation of the coarser approximations $\boldsymbol{A}_n, n \geq 1$.

III. Traditional Fixed-Point VLSI Circuits

The computational complexity of the DCT operation imparts a significant burden in VLSI circuits for real time applications. Several algorithms have been proposed to reduce the complexity of DCT circuits by exploiting its mathematical properties [38], [51], [52]. Integer transforms are employed in video standards such as H.264. However, we emphasize that these methods are approximations, which are inherently inexact and may introduce a computational error floor to the DCT evaluation.

A unified distributed-arithmetic parallel architecture for the computation of DCT and the DST was proposed in [53]. A direct-connected 3-D VLSI architecture for the 2-D prime-factor DCT that does not need a transpose memory (buffer) is available in [54]. A pioneering implementation at a clock of 100 MHz on 0.8 μm CMOS technology for the 2-D DCT with block-size 8×8 which is suitable for HDTV applications is available in [55].

An efficient VLSI linear-array for both N-point DCT and IDCT using a subband decomposition algorithm that results in computational-and hardware-complexity of $O(5N/8)$ with FPGA realization is reported in [56]. Recently, VLSI linear-array 2-D architectures and FPGA realizations having computation complexity $O(5N/8)$ (for forward DCT) was reported in [57].

An efficient adder-based 2-D DCT core on 0.35 μm CMOS using cyclic convolution is described in [58]. A high-performance video transform engine employing a space-time scheduling scheme for computing the 2-D DCT in real-time has been proposed and implemented in 0.18 μm CMOS [59]. A systolic-array algorithm using a memory based design for both the DCT and the discrete sine transform which is suitable for real-time VLSI realization was proposed in [60]. An FPGA-based system-on-chip realization of the 2-D DCT for 8×8 block size that operates at 107 MHz with a latency of 80 cycles is available in [61]. A low-complexity IP core for quantized $8 \times 8/4 \times 4$ DCT combined with MPEG4 codecs and FPGA synthesis is available in [62]. "New distributed-arithmetic (NEDA)" based low-power 8×8 2-D DCT is reported in [63]. A reconfigurable processor on TSMC 0.13 μm CMOS technology operating at 100 MHz is described in [64] for the calculation of the fast Fourier transform and the 2-D DCT. A high-speed 2-D transform architecture based on NEDA technique and having unique kernel for multi-standard video processing is described in [65].

IV. Error-Free Architectures

In this section, the idea is to obtain highest possible accuracy in a digital computing system for several popular discrete transforms and filters. The approach is to use algebraic integer (AI) based encoding that yields exactly computed results at the algorithm output. The methods trade computational complexity and power consumption for highest possible accuracy in the VLSI computing system.

The AI encoding was originally proposed for digital signal processing systems by Cozzens and Finkelstein [66]. Since then it has been adapted for the VLSI implementation of the 1-D DCT and other trigonometric transforms by Julien *et al.* in [67]–[71], leading to a 1-D bivariate encoded Arai DCT algorithm by Wahid and Dimitrov [52], [72]–[74]. Recently, subsequent contributions by Wahid *et al.* (using bivariate encoded 1-D Arai DCT blocks for row and column transforms of the 2-D DCT) has led to practical area-efficient VLSI video processing circuits with low-power consumption [75]–[77].We now briefly summarize the state-of-the-art in both 1-D and 2-D DCT VLSI cores based on conventional fixed-point arithmetic as well as on AI encoding.

A. AI-Based VLSI Circuits

The following AI-based realizations of 2-D DCT computation relies on the row-and column-wise application of

1-D DCT cores that employ AI quantization [67]–[71]. The architectures proposed by Wahid *et al.* rely on the low-complexity Arai Algorithm and lead to low-power realizations [72]–[76]. However, these realizations also are based on repeated application along row and columns of an fundamental 1-D DCT building block having an FRS section at the output stage. Here, 8×8 2-D DCT refers to the use of bivariate encoding in the AI basis and not to the a true AI-based 2-D DCT operation.

A 4×4 approximate 2-D-DCT using AI quantization is reported in [78]. Both FPGA implementation and ASIC synthesis on 90 nm CMOS results are provided. Although [78] employs AI encoding, it is not an error-free architecture. The low complexity of this architecture makes it suitable for H.264 realizations. Table 1 provides a comparison between different algebraic integer based 2-D DCT implementations.

B. Mathematical Background for AI Representation

In order to prevent quantization noise, we adopt the AI representation. Such representation is based on a mapping function that links input numbers to integer arrays.

This topic is a major and classic field in number theory. A famous exposition is due to Hardy and Wright [80, Chap. XI and XIV], which is widely regarded as masterpiece on this subject for its clarity and depth. Pohst also brings a didactic explanation in [81] with emphasis on computational realization. In [82, p. 79], Pollard and Diamond devote an entire chapter to the connections between algebraic integers and integral basis. In the following, we furnish an overview focused on the practical aspects of AI, which may be useful for circuit designers.

***Definition** 1:* A real or complex number is called an algebraic integer if it is a root of a monic polynomial with integer coefficients [80], [83].

The set of algebraic integers have useful mathematical properties. For instance, they form a commutative ring, which means that addition and multiplication operations are commutative and also satisfies distribution over addition.

Table 1.
Comparison of the algebraic integer based 2-D DCT implementation with existing algebraic integer implementations.

				Madanayake et al. [1]		Madanayake et al. [79]	
	Nandi et al. [78]	**Fu et al. [70]**	**Wahid et al. [77]**	**(12, 5, 13)**	**(437, 181, 473)**	**(12, 5, 13)**	**(437, 181, 473)**
Measured results	No	No	No	Yes	Yes	Yes	Yes
Structure	Single 1-D DCT +Mem. bank	Two 1-D DCT +Dual port RAM	Two 1-D DCT +TMEM	Five 1-D DCT +TMEM	Five 1-D DCT +TMEM	Two 1-D DCT +TMEM	Two 1-D DCT +TMEM
Multipliers	0	0	0	0	0	0	0
Exact 2D AI computation	No	No	No	Yes	Yes	Yes	Yes
Operating frequency (MHz)	N/A	75	194.7	300.39	307.78	949	946
8×8 blocks per clock cycle	1/128	1/64	1/64	1/8	1/8	1/8	1/8
8×8 block rate ($\times 10^6 s^{-1}$)	7.8125	1.171	3.042	37.55	38.47	118.625	118.25
Pixel rate ($\times 10^6 s^{-1}$)	125	75	194.7	1158.95	1187.35	7592	7568
Implementation technology	Xilinx XC5VLX30	0.18 μm CMOS	0.18 μm CMOS	Xilinx XCVLX240T	Xilinx XCVLX240T	45 nm CMOS	45 nm CMOS
Coupled quantization noise	Yes	Yes	Yes	No	No	No	No
Independantly adjustable precision	No	No	No	Yes	Yes	Yes	Yes
FRS between row-column stages	No	Yes	Yes	No	No	No	No

A general AI encoding mapping has the following format

$$f_{\text{enc}}(x\,;\boldsymbol{z}) = \boldsymbol{a},$$

where $\boldsymbol{a}$ is a multidimensional array of integers and $\boldsymbol{z}$ is a fixed multidimensional array of algebraic integers. It can be shown that there always exist integers such that any real number can be represented with arbitrary precision [66]. Also there are real numbers that can be represented *without* error.

Decoding operation is furnished by

$$f_{\text{dec}}(\boldsymbol{a}\,;\boldsymbol{z}) = \boldsymbol{a} \bullet z,$$

where the binary operation $\bullet$ is the generalized inner product—a component-wise inner product of multidimensional arrays. The elements of $\boldsymbol{z}$ constitute the AI basis. In hardware, decoding is often performed by an FRS block, where the AI basis $\boldsymbol{z}$ is represented as precisely as required.

As an example, let the AI basis be such that $\boldsymbol{z} = [1 \;\; z_1]^T$, where z_1 is the algebraic integer $\sqrt{2}$ and the superscript ^{T} denotes the transposition operation. Thus, a possible AI encoding mapping is $f_{\text{enc}}(x;\boldsymbol{z}) = \boldsymbol{a} = [a_0 \;\; a_1]^T$, where a_0 and a_1 are integers. Encoded numbers are then represented by a 2-point vector of integers. Decoding operation is simply given by the usual inner product: $x = \boldsymbol{a} \bullet \boldsymbol{z} = a_0 + a_1 z_1$. For example, the number $1 - 2\sqrt{2}$ has the following encoding:

$$f_{\text{enc}}\left(1 - 2\sqrt{2}; \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}\right) = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

which is an *exact* representation.

In principle, any number can be represented in an arbitrarily high precision [66], [84]. However, within a limited dynamic range for the employed integers, not all numbers can be exactly encoded. For instance, considering the real number $\sqrt{3}$, we have $f_{\text{enc}}(\sqrt{3};[1 \;\; \sqrt{2}]^T) = [88 \;\; -61]^T$, where integers were limited to be 8-bit long. Although very close, the representation is not exact:

$$f_{\text{dec}}\left(\begin{bmatrix} 88 \\ -61 \end{bmatrix}; \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}\right) - \sqrt{3} \approx 9.21 \times 10^{-4}.$$

In a similar way, the multipliers required by the DCT could be encoded into 2-point integer vectors: $f_{\text{enc}}(c[n];\boldsymbol{z}) = [a_0[n] \;\; a_1[n]]^T$. Given that the DCT constants are algebraic integers [83], an exact AI representation can be derived [44]. Thus, the integer sequences $a_0[n]$ and $a_1[n]$ can be easily realized in VLSI hardware.

The multiplication between two numbers represented over an AI basis may be interpreted as a modular polynomial multiplication with respect to the monic polynomial that defines the AI basis. In the above particular illustrative example, consider the multiplication of the following pair of numbers $a_0 + a_1 z_1$ with $b_0 + b_1 z_1$, where b_0 and b_1 are integers. This operations is equivalent to the computation of the following expression:

$$(a_0 + a_1 x) \cdot (b_0 + b_1 x) \pmod{x^2 - 2}.$$

Thus, existing algorithms for fast polynomial multiplication may be of consideration [85, p. 311].

In practical terms, a good AI representation possesses a basis such that: (i) the required constants can be represented *without* error; (ii) the integer elements provided by the representation are sufficiently small to allow a simple architecture design and fast signal processing; and (iii) the basis itself contains few elements to facilitate simple encoding-decoding operations.

Other AI procedures allow the constants to be approximated, yielding much better options for encoding, at the cost of introducing error within the transform (before the FRS) [83].

C. Bivariate AI Encoding

Depending on the DCT algorithm employed, only the cosine of a few arcs are in fact required. We adopted the Arai DCT algorithm [38]; and the required elements for this particular 1-D DCT method are only [52], [72], [73]:

$$\begin{aligned} c[4] = \cos\frac{4\pi}{16}, \;\; c[6] = \cos\frac{6\pi}{16}, \\ c[2] - c[6] = \cos\frac{2\pi}{16} - \cos\frac{6\pi}{16}, \\ c[2] + c[6] = \cos\frac{2\pi}{16} + \cos\frac{6\pi}{16}. \end{aligned}$$

These particular values can be conveniently encoded as follows. Considering $z_1 = \sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}}$ and $z_2 = \sqrt{2+\sqrt{2}} - \sqrt{2-\sqrt{2}}$, we adopt the following 2-D array for AI encoding:

$$\boldsymbol{z} = \begin{bmatrix} 1 & z_1 \\ z_2 & z_1 z_2 \end{bmatrix}.$$

This leads to a 2-D encoded coefficients of the form (scaled by 4):

$$f_{\text{enc}}(x;\boldsymbol{z}) = \boldsymbol{a} = \begin{bmatrix} a_{0,0} & a_{1,0} \\ a_{0,1} & a_{1,1} \end{bmatrix}.$$

Such encoding is referred to as bivariate. For this specific AI basis, the required cosine values possess an error-free and sparse representation as given in Table II [52], [72], [73]. Also we note that this representation utilizes very small integers and therefore is suitable for fast arithmetic computation. Moreover, these employed integers are powers of two, which require no hardware components other than wired-shifts, being cost-free.

Encoding an arbitrary real number can be a sophisticated operation requiring the usage of look-up tables and greedy algorithms [86]. Essentially, an exhaustive search is required to obtain the most accurate representation. However, integer numbers can be encoded effortlessly:

$$f_{\text{enc}}(m; \boldsymbol{z}) = \begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}, \tag{2}$$

where m is an integer. In this case, the encoding step is unnecessary. Our proposed design takes advantage of this property.

For a given encoded number $\boldsymbol{a}$, the decoding operation is simply expressed by:

$$f_{\text{dec}}(\boldsymbol{a}; \boldsymbol{z}) = \boldsymbol{a} \bullet \boldsymbol{z} = a_{0,0} + a_{1,0}z_1 + a_{0,1}z_2 + a_{1,1}z_1z_2.$$

In terms of circuitry design, this operation is usually performed by the FRS.

In order to reduce and simplify the employed notation, hereafter a superscript notation is used for identifying the bivariate AI encoded coefficients. For a given real x, we have the following representation

$$\begin{bmatrix} x^{(a)} & x^{(b)} \\ x^{(c)} & x^{(d)} \end{bmatrix} \equiv x = x^{(a)} + x^{(b)}z_1 + x^{(c)}z_2 + x^{(d)}z_1z_2, \tag{3}$$

where superscripts $^{(a)}$, $^{(b)}$, $^{(c)}$, and $^{(d)}$ indicate the encoded integers associated to basis elements 1, z_1, z_2, and z_1z_2, respectively. We denote this basis as $\boldsymbol{z}_4 = \{1, z_1, z_2, z_1z_2\}$.

It is worth to emphasize that in the 2-D AI encoding the equivalence between the algebraic integer multiplication and the polynomial modular multiplication does not hold true. Thus, a tailored computational technique to handle this operation must be developed.

D. DCT Architectures

An area efficient VLSI architecture has been proposed for the real-time implementation of bivariate AI-encoded 2-D DCT for video processing. The proposed architecture computes 8×8 2-D DCT transform based on the Arai DCT algorithm in a row parallel format. In [79], a fast algorithm for AI based 1-D DCT computation was recently proposed, accompanied by a single channel 2-D DCT architecture, aimed at improving the 4-channel AI DCT architecture in [1]. The improvements were achieved by reducing the number of integer channels to one and the number of 8-point 1-D DCT cores from 5 down to 2. The latest architecture in [1], [79] offers *exact computation* of 8×8 blocks of the 2-D DCT coefficients all the way up to the FRS. Recall that the FRS is the part of the algorithm that is used to convert the error free coefficients from the AI representation to fixed-point format using methods such as expansion factors.

The 1-D DCT algorithm described in [1] was derived from the 1-D AI Arai DCT [1], [72], using algebraic manipulations.

Table 2.
2-D AI encoding of arai DCT constants.

$c[4]$	$c[6]$	$c[2] - c[6]$	$c[2] + c[6]$
$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$

The arithmetic complexity consists of 20 additions. No multiplications or shift operations are required. This provides an improvement in terms of hardware resources when compared with the algorithm proposed in [1] and [72], which requires 21 additions and 2 shift operations. The proposed method as well as its competitors are highly optimized procedures. Thus one may not expect enormous gains in terms of computational complexity.

Let $\boldsymbol{A}$ denote the matrix transformation associated to the 1-D Arai DCT algorithm defined over the AI structure and $\boldsymbol{B}$ denote the matrix related to the operations in the FRS. Matrix $\boldsymbol{A}$ is represented in terms of signal flow diagram in the dashed block of Fig. 3(a). Matrix $\boldsymbol{B}$ is simply represents the AI presentation, ensuring that the resulting calculation furnishes AI as shown in (3).

Therefore, the complete 1-D AI DCT is given by

$$\boldsymbol{X}_{1\text{D}} = \boldsymbol{B} \cdot \boldsymbol{A} \cdot \boldsymbol{x}_{1\text{D}},$$

where $\boldsymbol{x}_{1\text{D}}$ and $\boldsymbol{X}_{1\text{D}}$ are 8-point column vectors for the input and output sequences, respectively, and where

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & 0 & 0 & 1 & 1 & 0 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & z_1z_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -z_1z_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -z_2 & -z_1z_2 & -z_1 & 1 \\ 0 & 0 & 0 & 0 & z_2 & -z_1z_2 & z_1 & 1 \\ 0 & 0 & 0 & 0 & -z_1 & z_1z_2 & z_2 & 1 \\ 0 & 0 & 0 & 0 & z_1 & z_1z_2 & -z_2 & 1 \end{bmatrix}. \tag{4}$$

The multiplications present in $\boldsymbol{B}$ can be efficiently implemented by means of the expansion factor method as described in [43]. The expansion factor method employs a factor α which scales the required multiplicands z_1, z_2, and z_1z_2 into quantities that are close to integers governed by the following relationship:

$$\alpha \cdot [z_1 \;\; z_2 \;\; z_1 z_2] \approx [m_1 \;\; m_2 \;\; m_3], \tag{5}$$

where m_1, m_2, and m_3 are integers. This approach entail a reduction in the overall number of multiplications required by the FRS [43].

Let $\boldsymbol{x}_{2D}$ and $\boldsymbol{X}_{2D}$ be 8×8 input and output data of the 2-D DCT, respectively. The 2-D DCT of $\boldsymbol{X}_{2D}$ can be obtained after (i) applying the 1-D DCT to its columns; (ii) transposing the resulting matrix; and (iii) applying the 1-D DCT to the rows of the transposed matrix. Mathematically, above procedure is given by:

$$\boldsymbol{X}_{2D} = \boldsymbol{B} \cdot \boldsymbol{A} \cdot \boldsymbol{x}_{2D} \cdot (\boldsymbol{B} \cdot \boldsymbol{A})^{\top} = \boldsymbol{B} \cdot \boldsymbol{A} \cdot \boldsymbol{x}_{2D} \cdot \boldsymbol{A}^{\top} \cdot \boldsymbol{B}^{\top}. \tag{6}$$

Matrix multiplications by $\boldsymbol{B}$ (reconstruction step) should be the final computation stage. Otherwise, an earlier multiplication by $\boldsymbol{B}$ represents an intermediate reconstruction stage, which may reintroduce numerical representation errors. Such errors would then be propagate to subsequent 1-D DCT calls. This may render the purpose of AI encoding ineffective.

In [1] an AI based architecture, where the reconstruction step occurs only at the end of the computation, was proposed. In other words, the reconstruction step was placed after both row- and column-wise transforms. In a recent contribution [1], an architecture which requires a single column-wise DCT has been made available. The block diagram of the 2-D DCT architecture in [1] is given in Fig. 3(b).

Several prototype circuits corresponding to FRS blocks based on two expansion factors have been realized, tested, and verified using FPGA technology employing a Xilinx Virtex-6 XC6VLX240T device. Post place-and-route results show a 20% reduction in terms of area compared to the 2-D DCT architecture requiring five 1-D AI cores. The area-time and area-time^2 complexity metrics are also reduced by 23% and 22% respectively for designs with 8-bit input word length. The above-mentioned digital realizations have been successfully simulated up to place and route for ASICs using 45 nm CMOS standard cells from the FreePDK library [87]. The maximum estimated clock rate is 951 MHz for the CMOS realizations indicating $7.608 \cdot 10^9$ pixels/seconds and a 8×8 block rate of 118.875 MHz. Several hardware platforms shown in Fig. 4, including Berkeley Emulation Engine (BEE3) [a], Xilinx ML405 [b], Xilinx ML605 [c], Xilinx XtremeDSP Kit-4 [d] and Achronix SPD60 FPGA [e] were used in the rapid prototyping and hardware evaluation of the algorithms.

AI DCT is implemented and tested on asynchronous quasi delay-insensitive logic, using Achronix SPD60 FPGA, which leads to lower complexity, higher speed of operation, and insensitivity to process-voltage-temperature variations. Results indicate a 31% improvement over the integer DCT in the number of transform coefficients having error within 1%. The performance of the 65 nm asynchronous hardware in terms of speed of operation was investigated and compared with the 65 nm synchronous Xilinx FPGA. Considering word lengths of 5 and 6 bits, a speed increase of 230% and 199% was reported in [2].

E. Wavelet Transform Architectures

The field of discrete wavelet transforms (DWT) has recently attracted much interest. The DWT has enjoyed such attention in part due to the fact that wavelet analysis is capable of decomposing a signal into a particular set of basis functions equipped with good spectral properties [88]–[91]. In fact, wavelet analysis is used to detect system non-linearities by making use of its localization feature [92]. DWT-based multi-resolution analysis leads to both time and frequency localization [91], [93]–[96].

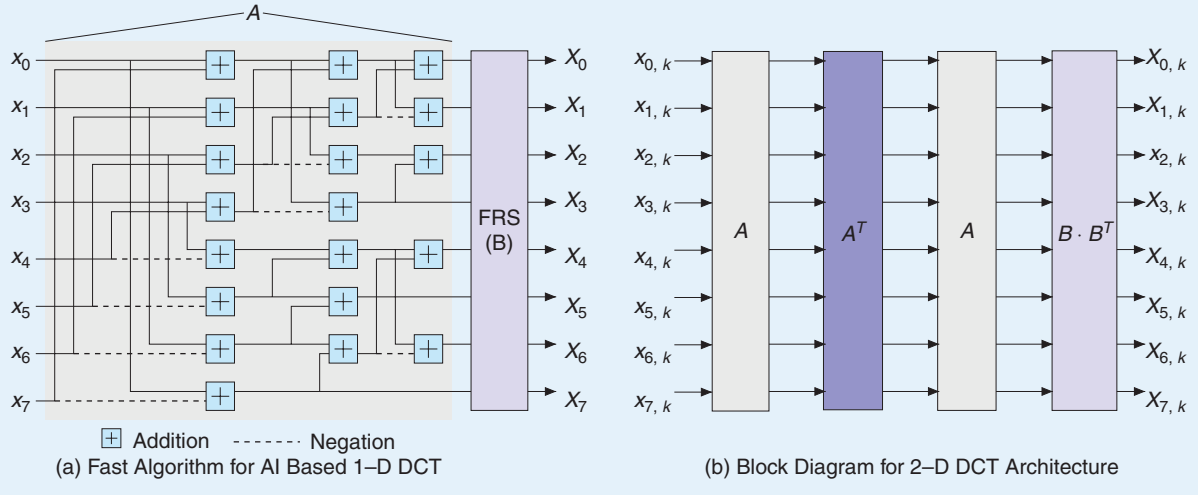

Figure 3. Fast algorithm for AI based 1-D DCT and the block diagram for 2-D DCT architecture.

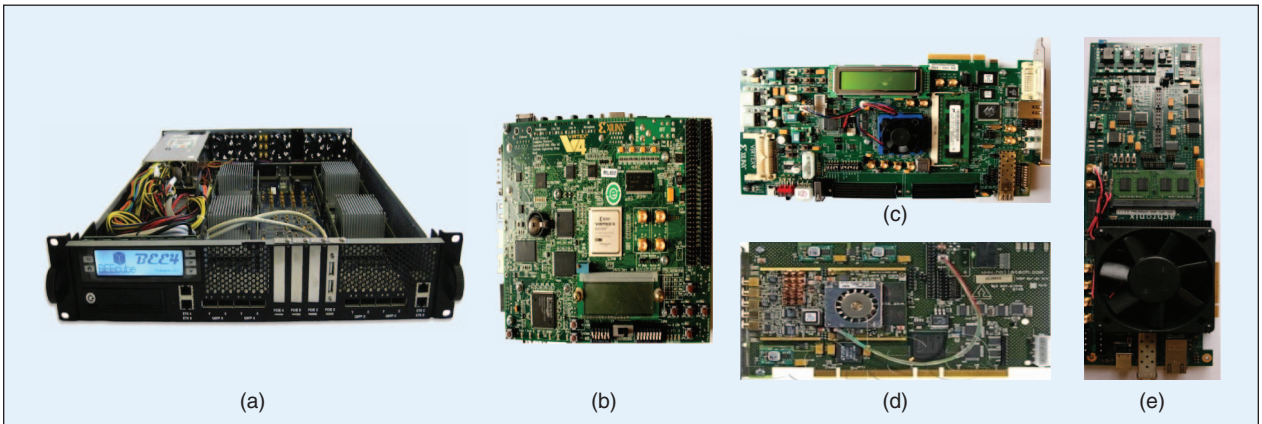

Figure 4. Different devices used in rapid prototyping and hardware evaluation of the algorithms.

DWT filter banks establish a strong support for digital signal/image processing systems [97]. For example, wavelets can be used in numerical analysis [98], [99], real-time processing [98], image/video compression and reconstruction [90], [100]–[104], pattern recognition [98], biomedicine [99], approximation theory, computer graphics [105] and various multimedia coding standards (H.265) [6], [46]. Following the use of the bi-orthogonal 2.2 DWT filters in JPEG2000 [47], [90], [93], [100], significant effort has been expended on research on how to reduce the computational and circuit complexities of such DWT hardware architectures in VLSI systems standpoint [89], [98], [100], [106], [107].

A particularly important class of DWT are the Daubechies wavelets [50] as they are indeed suited and extensively utilized for image compression applications [48], [90], [108]. We will refer to the Daubechies wavelets generated from 4-and 6-tap filter banks as Daub-4 and -6 wavelets, respectively, for brevity and simplicity. In particular, whereas the Daub-4 wavelets are often employed in applications where the signals are smooth and slowly varying, the Daub-6 wavelets are used for signals bearing abrupt changes, spikes, and having high undesired noise levels [98]. Daub-4 wavelets can be highly localized to smooth [48], [89] and Daub-6 wavelets have found applications in medical imaging, such as wireless capsule endoscopy where images of fine details are regarded important [7], [50], [98].

Because wavelets can be associated to specific filter banks, practical wavelet analysis is achieved by means of sub-band coding [50], [109]–[111]. Sub-band coding is a fundamental filtering principle which is used for decomposing a given signal in to frequency bands for subsequent encoding [112]. Especially important is the case of 2-D multi-resolution analysis, which is obtained via sub-band coding [50], [109].

In [3], a new multi-encoding technique that achieves exact computation of multi-level 2-D Daubechies wavelet transforms using algebraic integer (AI) encoding was described. When compared to traditional AI designs in the literature [6], [88]–[90], [93], [98], [100], the design proposed in [3] can compute wavelet image approximations entirely over integer fields and with a *single* FRS in a purely AI based 2-D architecture. The design [3] therefore avoids the need of intermediate reconstruction steps.

Moreover, this newly designed architecture is sought to be multiplier-free. Such design facilitate accuracy, speed, relatively smaller area on chip as well as cost of design. The new design is multi-encoded and multi-rate, operating over AI with no intermediate reconstruction steps. Error-free computations is performed until the final FRS. The architecture emphasizes on quality of output image and speed by trading complexity and power consumption for accuracy.

In [3], the authors focused on the computation of the coarser approximations $\boldsymbol{A}_n, n \geq 1$. The topmost branch of the signal flow shown in Fig. 5(a) computes the approximation data.

Let the low-pass filter associate to these filter banks be denoted as $\boldsymbol{h}^{(\mathrm{Daub}-4)}$ and $\boldsymbol{h}^{(\mathrm{Daub}-6)}$, respectively. These particular filters possess irrational quantities as shown below [50], [89], [90], [93], [113]:

$$\boldsymbol{h}^{(\mathrm{Daub}-4)} = \frac{1}{4\sqrt{2}}[1+\sqrt{3} \quad 3+\sqrt{3} \quad 3-\sqrt{3} \quad 1-\sqrt{3}]^{\top},$$

$$\boldsymbol{h}^{(\mathrm{Daub}-6)} = \frac{1}{16\sqrt{2}}\begin{bmatrix} 1+\sqrt{10}+\sqrt{5+2\sqrt{10}} \\ 5+\sqrt{10}+3\sqrt{5+2\sqrt{10}} \\ 10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}} \\ 10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}} \\ 5+\sqrt{10}-3\sqrt{5+2\sqrt{10}} \\ 1+\sqrt{10}-\sqrt{5+2\sqrt{10}} \end{bmatrix},$$

where the superscript $^{\top}$ denotes transposition. Comparisons are provided in Table 3 and 4 between Daubechies-4 and -6 designs in terms of SNR, PSNR, hardware

structure, and power consumptions, for different word lengths. SNR and PSNR improvements of approximately 30% were observed in favour of AI-based systems, when compared to 8-bit fixed-point schemes (six fractional bits). Filter structures for Daub-4 and Daub-6 are shown in Fig. 6 and Fig. 7 displays images resulted from the Xilinx FPGA design as well as from the fixed point design for the Daub-4 and -6 filter banks. Further, FRS designs based on canonical signed digit representation and on expansion factors are proposed. The Daubechies-4 and -6 4-level VLSI architectures are prototyped on a Xilinx Virtex-6 vcx240t-1ff1156 FPGA device at 282 MHz and 146 MHz, respectively, with dynamic power consumption of 164 mW and 339 mW, respectively, and verified on FPGA chip using an ML605 FPGA based rapid prototyping platform.

V. Approximate Computation Architectures

A. Background

The high-definition video industry is exploding with both technological enhancements as well as content. In addition to standard high-definition video formats such as H.264/AVC and modern improvements such as H.265/HEVC standard, there is a emerging trend for better and better

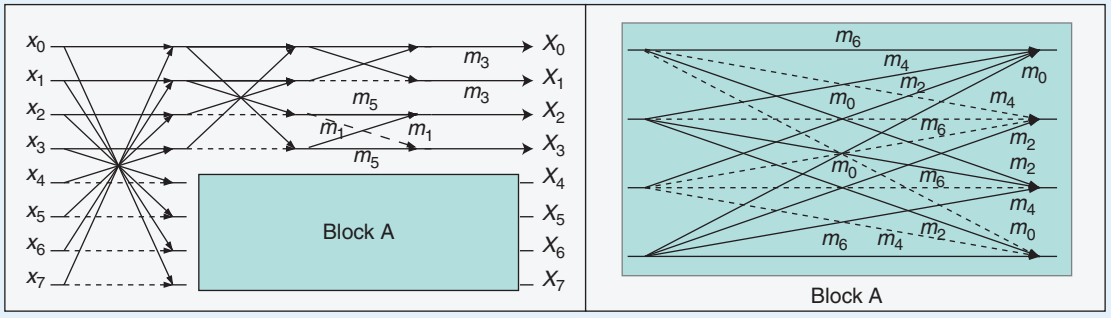

Figure 5. General signal flow graph for DCT transformations. Input data $x_n, n = 0,1,\ldots,7$, relates to output $X_k, k = 0,1,\ldots,7$, according to $\boldsymbol{X} = \boldsymbol{T} \cdot \boldsymbol{x}$. Dashed arrows represent multiplications by –1.

Table 3.
Comparison of AI based DWT architectures.

	Wahid et al, [88]	Wahid et al, [93]	Wahid et al, [100]	Wahid et al, [90]	Madishetty et al, [3]	Wahid et al, [98]	Wahid et al, [114]	Madishetty et al, [115]
LR[1]	No	No	No	No	Yes[2]	No	No	Yes[2]
Wavelet	Daub-4/6	Daub-4/6	Daub-6	Daub-4/6	Daub-4/6	Daub-4/6	Daub-4	Daub-4
Architecture	1-D/2-D	1-D	1-D	1-D	2-D	1-D	1-D	2-D
IRS[3]	Yes	Yes	Yes	Yes	No	Yes	Yes	No
HC[4]	10/18	10/25	25	N/A	10/21	9/18	9	7
Registers[5]	200/494	N/A	N/A	N/A	258/765	115/196	200	236
Logic cells[5]	248/680	N/A	N/A	N/A	426/1040	106/254	N/A	483
PSNR (dB)[6]	38† (Daub-4) 39† (Daub-6)	50.49 (Daub-4) 49.87 (Daub-6)	N/A	22.26	54.64 (Daub-4) 57.12 (Daub-4)	N/A	44.90	64.78
FRS[7]	CSD	CSD	Booth	Booth	CSD/EFM[8]	CSD	CSD	CSD
Throughput	1 ip/op	1 ip/op	N/A	N/A	1 ip/op	2 ip/op	1 ip/op	1 ip/op
DP[9] (mW)[5]	16.94/22.29	N/A	N/A	N/A	38/57	4.51‡	N/A	46§
Technology	Xilinx VirtexE	N/A	N/A	N/A	Xilinx xc6vcx240t	CMOS 0.18 um	Xilinx VirtexE	Xilinx xc6vcx240t
MF[10] (MHz)	148.21/119.57	N/A	N/A	N/A	282.50/146.42	100	148	263.15

[1]Locally Reproduced (LR). [2]Measured. [3]Intermediate Reconstruction Step (IRS). [4]Hardware Complexity (HC) (# adders). [5]For single level decomposition.
[6]For 8-bit Lena Image. [7]Final Reconstruction Step (FRS). [8]Expansion Factor Method (EFM). [9]Dynamic Power (DP). [10]Maximum Frequency (MF).
†Taken from PSNR plots in [88, p. 1264]. ‡At 50 MHz [98]. §At 362:18 MHz. *At 238:74 MHz.

picture quality achieved via higher screen resolutions, known as 2160p and 4320p. HEVC is now frozen and aimed at consumer electronics. It is noted that the HEVC standard refers to particular types of integer transforms that must be present in the encoders in order to confirm to standard decoders. There is continuous proliferation in the video sector, especially H.265/HEVC. Nevertheless, we continue to explore novel transform algorithms and architectures for scientific merit and interest from an engineering design perspective. The knowledge and experience gathered by trying better transforms and algorithms over those frozen in the standard will help researchers create better innovations for future systems. For example, emerging super high vision (SHV) technology, also known as 8K UHD, improves the HEVC resolution of 1920×1080 significantly, to yield screen resolutions up to 7680×4320 at the standard aspect radio of 19 : 6. The high complexity of encoding 8K UHD video content in real-time is partially addressed by adopting high degrees of parallelism. For example, in the first VLSI implementation of 8K UHD by

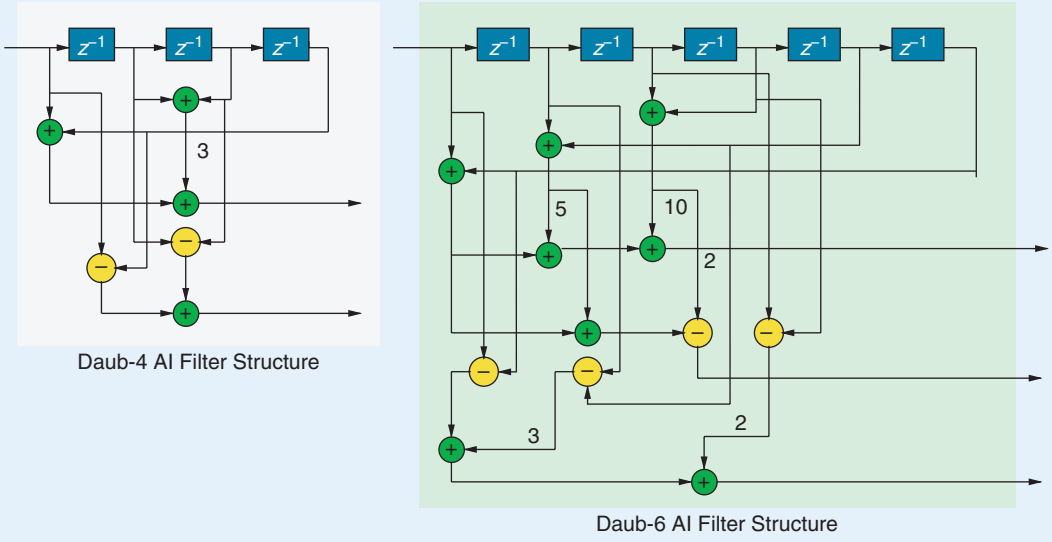

Figure 6. Daub-4 and Daub-6 filter structures.

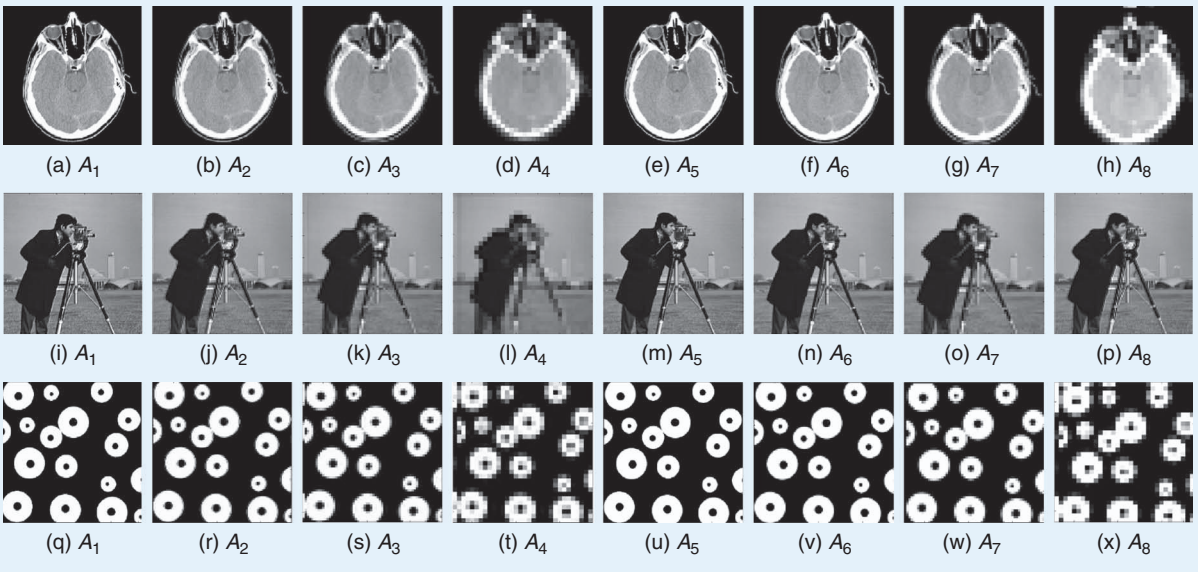

Figure 7. Approximation sub-images $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, and $\boldsymbol{A}_4$ (on left) obtained from on-chip physical verification on a Virtex-6 vcx240t-1ff1156 and sub-images $\boldsymbol{A}_5$, $\boldsymbol{A}_6$, $\boldsymbol{A}_7$, and $\boldsymbol{A}_8$ obtained from FP scheme (8-bit word length and 6 fractional bits) Daubechies 4- and 6-tap wavelet filters.

Table 4.
Comparison of AI based DWT architectures in literature.

	Wahid *et al.* [88]	Wahid *et al.* [93]	Wahid *et al.* [100]	Wahid *et al.* [90]	Wahid *et al.* [98]	Gustafsson *et al.* [116]	Wahid *et al.* [114]	Madishetty *et al.* [3]	Madishetty *et al.* [117]
LR[1]	No	No	No	No	No	No	No	Yes[2]	Yes[2]
Wavelet	Daub-4/6	Daub-4/6	Daub-6	Daub-4/6	Daub-4/6	Daub-6	Daub-6	Daub-4/6	Daub-6
Architecture	1-D/2-D	1-D	1-D	1-D	1-D	1-D	1-D	2-D	1-D/2-D
IRS[3]	Yes	Yes	Yes	Yes	Yes	N/A	Yes	No	No
HC[4]	10/18	10/25	25	N/A	9/18	18	18	10/21	17*/18**
Registers[5]	200/ 494	N/A	N/A	N/A	115/196	N/A	494	258/765	177◊/593◊◊
Logic cells[5]	248/680	N/A	N/A	N/A	106/254	N/A	340	426/ 1040	248◊/ 867◊◊
PSNR (dB)[6]	38 †(Daub-4) 39 †(Daub-6)	50.49 (Daub-4) 49.87 (Daub-4)	N/A	22.26	N/A	N/A	43.20	66.82 (Daub-4) 68.12 (Daub-6)	70.54
FRS[7]	CSD	CSD	Booth	Booth	CSD	CSD	CSD	CSD/EFM[8]	CSD
Throughtput	1 ip/op	1 ip/op	N/A	N/A	2 ip/op	1 ip/op	1 ip/op	1 ip/op	1 ip/op
DP[9](mW)[5]	15.94/22.29	N/A	N/A	N/A	4.51 ‡	N/A	N/A	38 § /57*	61 ††
Technology	Xilinx VirtexE	N/A	N/A	N/A	CMOS 0.18um	N/A	Xilinx Virtex E	Xilinx Virtex 6 vcx240t	Xilinx Virtex 6/ CMOS 45 nm
MF[10] (MHz)	148.21 ◊ 119.57 ◊◊	N/A	N/A	N/A	100	N/A	120 (Daub-6)	282.50◊◊ (Daub-4) 146.42◊◊ (Daub-6)	168.83 A 306.15 B

[1]Locally Reproduced (LR). [2]Measured. [3]Intermediate Reconstruction Step (IRS). [4]Hardware Complexity (HC) (# adders) for an AI Daub-6 filter. [5]For single level decomposition. [6]For 8-bit Lena Image. [7]Final Reconstruction Step (FRS). [8]Expansion Factor Method (EFM). [9]Dynamic Power (DP). [10]Maximum Frequency (MF). †Taken from PSNR plots in [88, p. 1264]. ‡At 50 MHz [98]. §At 442.47 MHz. [A]Using a Xilinx Virtex-6 xc6vcx240t-1ff1156 FPGA device. [B]Using CMOS 45 nm implementation generated by Encounter® RTL compiler. *At 274.72 MHz. ††At 374.25 MHz. ‡‡At 346.58 MHz. ◊For 1-D Design. ◊◊For 2-D Design.

Japan's NHK in 2013, the picture screen was divided into 17 strips, each of size 7680×256 pixels, and each strip was encoded using a separate HEVC encoder [118], [119]. Such an encoding architecture is illustrated in Fig. 8. The expanding industry of high-definition video content and its efficient delivery over IP networks calls for better compression algorithms following HEVC/H.265, of which a major component lies in the macro-blocking and energy compaction operation.

After the introduction of the DCT by Ahmed *et al.* [12] in 1974, designing efficient DCT algorithms has been a major scientific effort in the circuits, systems, and signal processing community with tremendous applications for image and video compression. Because of such intense research in the field, the current *exact* methods are very close to the theoretical DCT complexity [14], [38], [39], [79], [120]. Therefore, it may be unrealistic to expect that new exact algorithms could offer dramatic computational gains for such a fundamental and deeply investigated mathematical operation.

In this scenario, *approximate methods offer an alternative way to further reduce the computational complexity* of the DCT [11], [15], [121]–[123]. While not computing the DCT exactly, such approximations can provide meaningful estimations at very low-complexity computational requirements. In this sense, literature has been populated with approximate methods for the efficient computation of the 8-point DCT. A comprehensive list of approximate methods for the DCT is found in [11]. Prominent techniques include the signed DCT (SDCT) [15], the level 1 approximation by Lengwehasatit-Ortega [124], the Bouguezel-Ahmad-Swamy (BAS) series of algorithms [122], [123], [125]–[128], the DCT round-off approximation [129], the modified DCT round-off approximation [121], and the multiplier-free DCT approximation for RF imaging [5].

Although several other types of approximations are available, in general, *very low-complexity* approximation matrices have their entries defined on the set $\{0, \pm 1/2, \pm 1, \pm 2\}$ [15], [121], [123], [128], [129]. Thus, such transformations possess null multiplicative complexity, because the required arithmetic operations can be implemented exclusively by means of binary additions and bit-shifting operations. Indeed, DCT approximations can replace the *exact* DCT in hardware implementation and

high-speed computation/processing [11], while having low hardware costs and low power demands [43]. Effectively, DCT approximations have been considered for applications in real-time video transmission and processing [130], [131], satellite communication systems [11], portable computing applications [11], radio-frequency smart antenna array [5], and wireless image sensor networks [132].

The proposed methods archived in literature for generating *very* low complexity approximations include: (i) crude approximations [15], [128], [129]; (ii) inspection [122], [126], [127]; (iii) variations of previous approximations via a single-parameter matrix [123]; and (iv) optimization procedures based on the DCT structure [5]. Thus, the existing approximations appear as *isolate* cases without an unifying mathematical formalism. Therefore, the discovery of a unified approach to creating low-complexity approximations and corresponding fast algorithms is a very important research challenge.

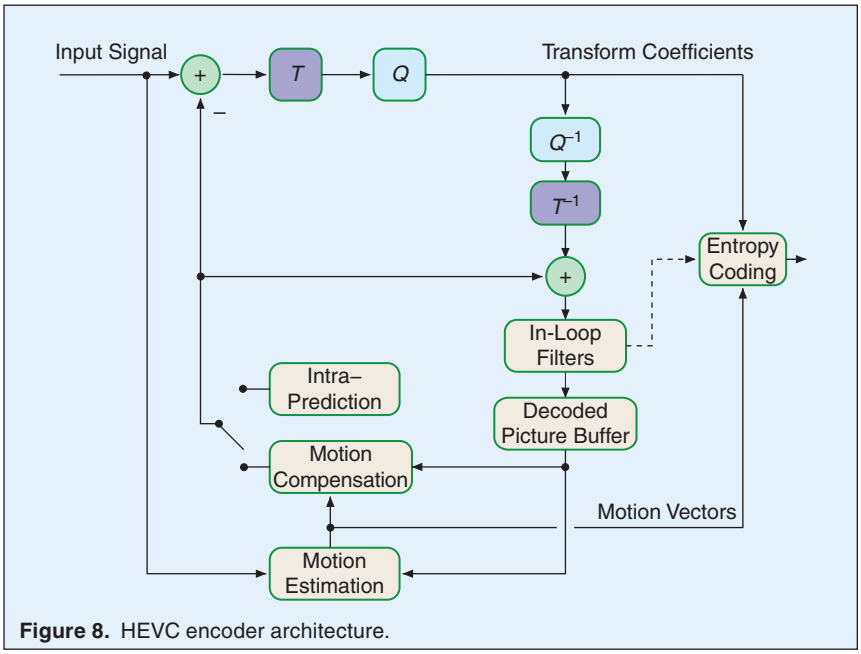

Figure 8. HEVC encoder architecture.

The aims of this line of research are: (i) the introduction of a general theory aiming at automatic generation of multiplier-free classes of DCT approximations; (ii) the proposal of new DCT approximations with excellent mathematical properties with respect to computational complexity and transform coding efficiency; and (iii) the derivation of very low-complexity fast algorithms suitable for the newest technology of image and video codecs.

VI. Exact and Approximate DCT

Generally, a DCT approximation is a transformation $\hat{\boldsymbol{C}}$ that—according to some specified metric—behaves similarly to the *exact* DCT matrix $\boldsymbol{C}$. An approximation matrix $\hat{\boldsymbol{C}}$ is usually based on a transformation matrix $\boldsymbol{T}$ of low computational complexity. Indeed, matrix $\boldsymbol{T}$ is the key component of a given DCT approximation.

Often the elements of the transformation matrix $\boldsymbol{T}$ possess null multiplicative complexity. For instance, this property can be satisfied by restricting the entries of $\boldsymbol{T}$ to the set of powers of two $\{0, \pm 1, \pm 2, \pm 4, \pm 8, \ldots\}$. In fact, multiplications by such elements are trivial and require only bit-shifting operations.

Approximations for the DCT can be classified into two categories depending on whether $\hat{\boldsymbol{C}}$ is orthonormal or not. In principle, given a low-complexity matrix $\boldsymbol{T}$ it is possible to derive an orthonormal matrix $\hat{\boldsymbol{C}}$ based on $\boldsymbol{T}$ by means of the polar decomposition [133], [134]. Indeed, if $\boldsymbol{T}$ is a full rank real matrix, then the following factorization is uniquely determined:

$$\hat{\boldsymbol{C}} = \boldsymbol{S} \cdot \boldsymbol{T}, \tag{7}$$

where $\boldsymbol{S}$ is a symmetric positive definite matrix [134, p. 348]. Matrix $\boldsymbol{S}$ is explicitly related to $\boldsymbol{T}$ according to the following relation:

$$\boldsymbol{S} = \sqrt{(\boldsymbol{T} \cdot \boldsymbol{T}^{\top})^{-1}},$$

where $\sqrt{\cdot}$ denotes the matrix square root operation [135], [136]. Being orthonormal, such type of approximation satisfies $\hat{\boldsymbol{C}}^{-1} = \hat{\boldsymbol{C}}^{\top}$. Therefore, we have that

$$\hat{\boldsymbol{C}}^{-1} = \boldsymbol{T}^{\top} \cdot \boldsymbol{S}.$$

As a consequence, the inverse transformation $\hat{\boldsymbol{C}}^{-1}$ inherits the same computational complexity of the forward transformation.

From the computational point of view, it is desirable that $\boldsymbol{S}$ be a diagonal matrix. In this case, the computational complexity of $\hat{\boldsymbol{C}}$ is the same as that of $\boldsymbol{T}$, except for the scale factors in the diagonal matrix $\boldsymbol{S}$. Moreover, depending on the considered application, even the constants in $\boldsymbol{S}$ can be disregarded in terms of computational complexity assessment. This occurs when the involved constants are trivial multiplicands, such as the powers of two. Another more practical possibility for neglecting the complexity of $\boldsymbol{S}$ arises when it can be absorbed into other sections of a larger procedure. This is the case in JPEG-like compression, where the quantization step is present [17]. Thus, matrix $\boldsymbol{S}$ can be incorporated into the quantization matrix [4], [121]–[124], [126], [129]. In terms of the inverse transformation, it is also beneficial that $\boldsymbol{S}$ is diagonal, because the complexity of $\hat{\boldsymbol{C}}^{-1}$ becomes essentially that of $\boldsymbol{T}^{\top}$.

In order that $\boldsymbol{S}$ be a diagonal matrix, it is sufficient that $\boldsymbol{T}$ satisfies the orthogonality condition:

$$\boldsymbol{T}\cdot\boldsymbol{T}^{\top}=\boldsymbol{D}, \tag{8}$$

where $\boldsymbol{D}$ is a diagonal matrix [133].

If (8) is not satisfied, then $\boldsymbol{S}$ is not a diagonal and the advantageous properties of the resulting DCT approximation are in principle lost. In this case, the off-diagonal elements contribute to a computational complexity increase and the absorption of matrix $\boldsymbol{S}$ cannot be easily done. However, at the expense of not providing an orthogonal approximation, one may consider approximating $\boldsymbol{S}$ itself by replacing the off-diagonal elements of $\boldsymbol{D}$ by zeros. Thus, the resulting matrix $\hat{\boldsymbol{S}}$ is given by:

$$\hat{\boldsymbol{S}}=\sqrt{[\operatorname{diag}(\boldsymbol{T}\cdot\boldsymbol{T}^{\top})]^{-1}},$$

where $\operatorname{diag}(\cdot)$ returns a diagonal matrix with the diagonal elements of its matrix argument. Thus, the non-orthogonal approximation is furnished by:

$$\tilde{\boldsymbol{C}}=\hat{\boldsymbol{S}}\cdot\boldsymbol{T}.$$

Matrix $\tilde{\boldsymbol{C}}$ can be a meaningful approximation if $\hat{\boldsymbol{S}}$ is, in some sense, close to $\boldsymbol{S}$; or, alternatively, if $\boldsymbol{T}$ is almost orthogonal.

From the algorithm designing perspective, proposing non-orthogonal approximations may be a less demanding task, since (8) is not required to be satisfied. However, since $\tilde{\boldsymbol{C}}$ is not orthogonal, the inverse transformation must be cautiously examined. Indeed, the inverse transformation does not employ directly the low-complexity matrix $\boldsymbol{T}$ and is given by

$$\tilde{\boldsymbol{C}}^{-1}=\boldsymbol{T}^{-1}\cdot\hat{\boldsymbol{S}}^{-1}.$$

Even if $\boldsymbol{T}$ is a low-complexity matrix, it is not guaranteed that $\boldsymbol{T}^{-1}$ also possesses low computational complexity figures. Nevertheless, it is possible to obtain non-orthogonal approximations whose both direct and inverse transformation matrices have low computational complexity. Two non-orthogonal examples are the SDCT [15] and the BAS approximation described in [122].

In this scenario, some classical examples of the low-complexity approximate methods for 8-point DCT are the SDCT [15] and the level 1 approximation in [124]. Indeed, recent literature in DSP presents the following prominent DCT approximations: the round-off DCT approximation (RDCT) [129], the modified RDCT (MRDCT) [121], the DCT approximation for RF imaging [5] and the improved DCT approximation [33]. In addition, in [137] a collection of DCT approximation are proposed based on integer functions that we selected an orthogonal and a non-orthogonal case for comparisons.

The low-complexity matrix $\boldsymbol{T}$ for each above transforms can be factorized following the same structure:

$$\boldsymbol{T}=\boldsymbol{P}\cdot\boldsymbol{K}\cdot\boldsymbol{B}_1\cdot\boldsymbol{B}_2\cdot\boldsymbol{B}_3,$$

where $\boldsymbol{P}$ is a permutation matrix, $\boldsymbol{K}$ is a multiplicative matrix, and $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, and $\boldsymbol{B}_3$ are additive matrices. These matrices are given by:

$$\boldsymbol{P}=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&0&0&-1&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&-1&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&-1\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\end{bmatrix},$$

$$\boldsymbol{K}=\begin{bmatrix}m_3&0&0&0&0&0&0&0\\0&m_3&0&0&0&0&0&0\\0&0&m_5&m_1&0&0&0&0\\0&0&-m_1&m_5&0&0&0&0\\0&0&0&0&m_4&-m_6&m_2&m_0\\0&0&0&0&-m_0&m_4&-m_6&m_2\\0&0&0&0&-m_2&-m_0&m_4&-m_6\\0&0&0&0&m_6&-m_2&-m_0&m_4\end{bmatrix},$$

$$\boldsymbol{B}_1=\begin{bmatrix}1&1&0&0&0&0&0&0\\1&-1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&0&-1&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&-1&0&0\\0&0&0&0&-1&0&0&0\end{bmatrix},$$

$$\boldsymbol{B}_2=\begin{bmatrix}1&0&0&1&0&0&0&0\\0&1&1&0&0&0&0&0\\1&0&0&-1&0&0&0&0\\0&1&-1&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix},$$

$$\boldsymbol{B}_3=\begin{bmatrix}1&0&0&0&0&0&0&1\\0&1&0&0&0&0&1&0\\0&0&1&0&0&1&0&0\\0&0&0&1&1&0&0&0\\1&0&0&0&0&0&0&-1\\0&1&0&0&0&0&-1&0\\0&0&1&0&0&-1&0&0\\0&0&0&1&-1&0&0&0\end{bmatrix},$$

where constants $m_i, i=0,1,\ldots,6,$ depend on the particular choice of transformation matrix $\boldsymbol{T}$. As a consequence, all transformations presented in [5], [15], [33], [121], [124], [129], [137] also share the same fast algorithm and signal flow structure, as presented in Fig. 5(b). In

Table 5.
Constants required for the fast algorithm.

Approximation	m_0	m_1	m_2	m_3	m_4	m_5	m_6
SDCT [15]	1	1	1	1	1	1	1
Level 1 [124]	1	1	1	1	1	½	0
RDCT [129]	1	1	1	1	1	0	0
MRDCT [121]	1	1	0	1	0	0	0
Approximation for RF [5]	2	2	1	1	1	1	0
Improved approximation [33]	0	1	1	1	0	0	0
Orthogonal selected in [137]	1	1	1	1	1	1	0
Nonorthogonal selected in [137]	1	1	1	1	0	0	0

Table 5, the constants are listed for each of the discussed transformations.

For each approximation, we could assess the performance in terms of computational complexity and similarity with the exact DCT. Classically, the computational complexity of the fast algorithms are assessed by the arithmetic complexity, as measured by multiplication, addition, and bit-shifting operation counts. Indeed, we assess the similarity measure by total error energy measure introduced in [129]. Taking the basis vectors of the exact DCT and a given approximation as filter coefficients, the total error energy [129], measures the spectral proximity between the corresponding transfer functions [15]. Invoking Parseval theorem [138], the total error energy can be evaluated according to:

$$\epsilon(\hat{\boldsymbol{C}}) = \pi \cdot \| \boldsymbol{C} - \hat{\boldsymbol{C}} \|_F^2,$$

where $\| \cdot \|_F$ is the Frobenius norm [139]. Table 6 shows the performance assessment, presenting arithmetic complexity and total error energy.

The minimization of total error energy indicates proximity to the exact DCT matrix. Table 6 shows strong similarities between the spectral characteristics of the DCT and the considered approximations. These similarities demonstrate the good energy related properties of the approximation algorithms. The best results are presented for the level 1 [124] and approximation for RF [5]. However, these transforms present the higher arithmetic complexity, with 24 additions each one. In the other hand, the MRDCT [121] and the improved approximation presented in [33] are the fastest algorithms archived in literature. The MRDCT presents better approximation to the DCT, with error energy equals to 8.66, than the transform presented in [33]. But, as mentioned in [33], the improved approximation [33] has superior performance when compared with MRDCT in image compression for high compression ratios. In this trade-off between complexity and similarity the RDCT highlights among all considered approximations. In non-orthogonal cases, the transform selected in [137] shows better performance compared with classical SDCT, possessing the same error energy at the same time possess lower computational cost.

Table 6.
Performance assessment.

Approximation	Mult	Add	Shift	Error Energy
Exact DCT [39]	11	29	0	0
SDCT [15]	0	24	0	3.32
Level 1 [124]	0	24	2	0.87
RDCT [129]	0	22	0	1.79
MRDCT [121]	0	14	0	8.66
Approximation for RF [5]	0	24	6	0.87
Improved approximation [33]	0	14	0	11.31
Orthogonal selected in [137]	0	24	0	1.79
Nonorthogonal selected in [137]	0	18	0	3.32

As a qualitative analysis, Fig. 9 shows the compressed Lena image with compression ratio equals to 84.37%, retained just the first ten coefficients following the JPEG-like experiments described in [15], [33], [121], [129], [137].

VII. Applications

In this section, the idea is to obtain multiplierless fast algorithms having the lowest complexity towards achieving well-know discrete computations such as DCT and DST at the cost of a loss in accuracy.

The emergence of standards for reconfigurable video (HEVC) has lead to renewed interest in both lower-complexity as well as in higher accuracy transform coders. The integer version of the DCT has found its place in many modern video codecs including HEVC. To enhance the on-chip transcoding capability for smart devices, the need was

Figure 9. Compressed Lena image using all considered transforms.

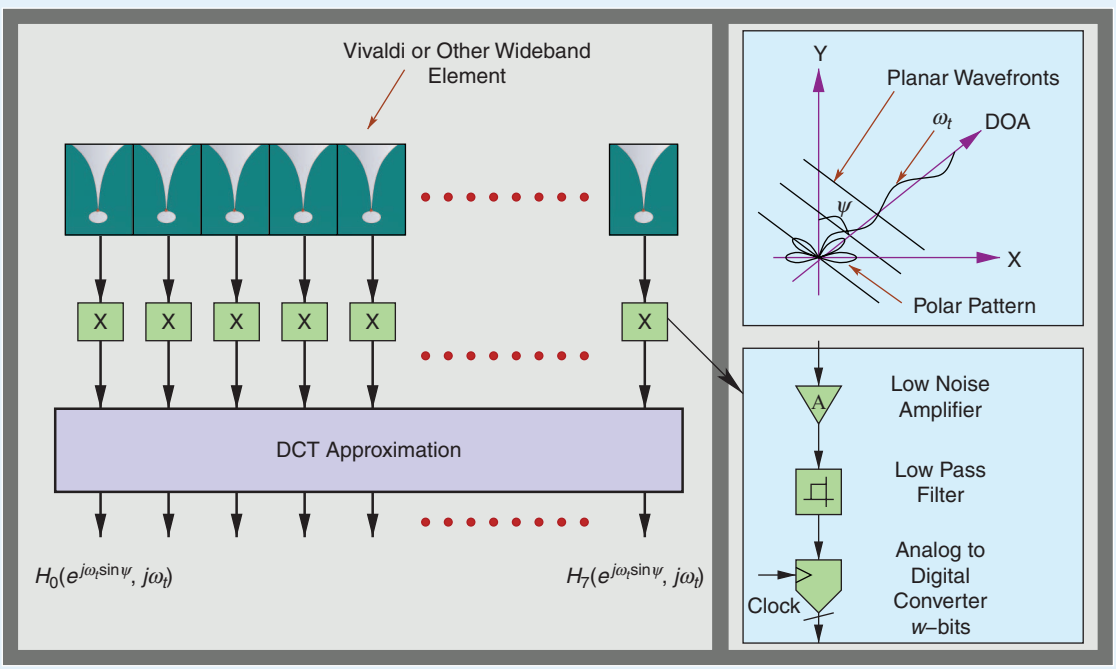

Figure 10. Application of DCT approximations in multi-beamforming.

felt to support more than one encoder into one single platform. Multi-codecs are capable of better picture quality at lower power consumption for a wide range of devices and technologies. Several multi-codec architectures that compute the inverse integer DCT using matrix decomposition and circuit-share strategy are discussed. Both algorithmic details as well as digital VLSI implementation details at silicon level will be included. In addition to integer DCTs and accurate realizations of the exact DCT using number theoretic approaches, the ultra-low complexity realization

of multiplierless DCT approximations is also of high interest [5]. As such, this paper discusses recent progress in the field of fast algorithms where well-established mathematical operations, such as the DCT and the discrete sine transform (DST), can be approximated using multiplierless methods. Such approximate DCTs often possess transformation matrix whose entries are in $\{0, \pm 1/2, \pm 1, \pm 2\}$. Thus the use of multiplierless architectures imply only register delays and adders are required in circuit realizations. This property leads to very lower power consumption. A general numerical optimization framework is derived and several examples of fast algorithms are scrutinized in terms of VLSI implementations and computational noise power. Complete algorithmic details as well VLSI circuit implementation methods and metrics are provided. Standard images from public databases are employed for obtaining a comparison of performance between exact trigonometric transforms and the low-complexity approximations in the context of image coding and compression. Because of the significantly lower circuit complexity, new applications of the proposed multiplierless transforms enable unconventional applications, such as RF beamforming [5].

Multi-beamforming requires operations at low-complexity for radio-frequency (RF) aperture arrays. Numerous applications exist for RF multi-beamformers. For example, emerging radio telescopes such as the Square Kilometer Array (SKA) require simultaneous independent RF beams for Big-Science astronomy and space physics experiments of the future. Another important example of RF multi-beamforming is in remote sensing, wireless communications, microwave imaging, signal intelligence, and radar. Traditional algorithms are based on phased-array technology which can be realized using FIR filterbanks and/or the FFT. In our recent work, we proposed the application of high-speed approximate DCT fast algorithms as an alternative for the DFT enabling their applications in RF multi-beamforming. The side-lobe level of the proposed beamformers are greater than those of DFT based methods. However, the multi-RF beams are achieved at very low complexity compared to traditional methods. Typical RF beams in the polar domain are shown in Fig. 11 where four of the eight possible beam directions for an 8-point DCT approximation has been provided. A null multiplicative complexity is achieved with digital architectures having greatly reduced complexity in the arithmetic circuits which consist of networks of parallel adders. A comparative study of several low complexity multi-beamforming with performance close to the ideal DCT [1], [2], [4], [5] will be discussed.

The realization of highly precise numerical computations, while dealing with finite computational resources has a prominent role on the circuit implementation of fast algorithms for multimedia applications. Many of discrete transforms such as the DCT or the DFT regularly encounter irrational numbers which invariably incur numerical errors when approximated using traditional fixed-point number representation schemes such as the extremely well

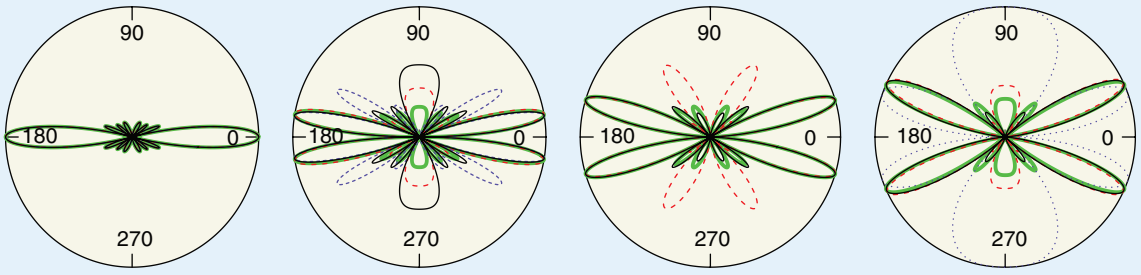

Figure 11. Polar plots of transfer functions of DCT fast algorithms.

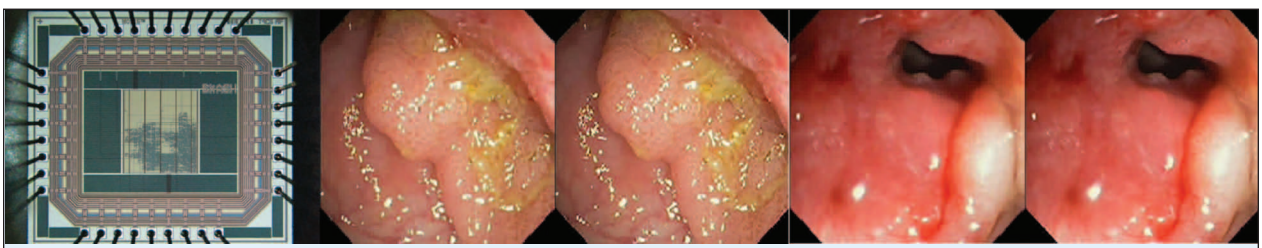

Figure 12. Left: Chip micrograph: "Dual Daubechies Wavelet Transform Processor for Image Compression," Size: 1.76 mm × 1.52 mm (CMOSP18), 7,533 transistors, 100MHz, 4.51 mW, [98], Middle and Right: Two reconstructed images (simulated). In both cases, AI (on-right) produced 2-3 dB better results and 4-5% higher compression than its integer DCT counterpart [7].

known two's complement arithmetic encoding technique. Because numerical error leads to a penalty in the signal to noise ratio of a signal processing systems, high performance systems increasingly demand better accuracy than what can be achieved using traditional fixed-point methods at reasonable power consumption or circuit complexity. Therefore, the fixed point precision can in theory be increased indefinitely to achieve any required level of accuracy. However, this comes as a cost of circuit complexity which translates to silicon real estate; lower operational speeds, due to increased critical path latency; and higher power consumption. We positively demonstrate that better arithmetic encoding techniques based on number theoretic tools can furnish required accuracy while maintaining acceptable circuit complexity and low power consumption. In particular, we consider number representation schemes based on algebraic integer, where irrational quantities can be given exact representations over an integer arithmetic.

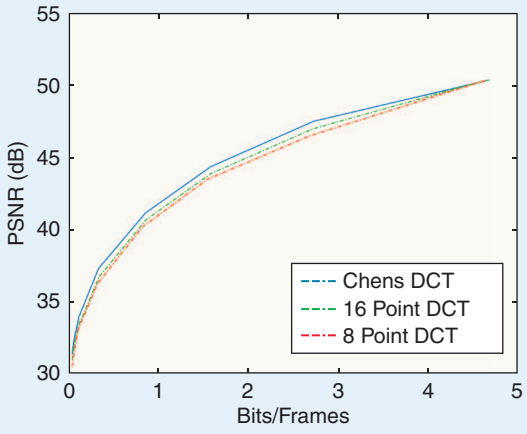

Figure 13. RD curves for 'BasketballPass' test sequence.

We summarize several low-cost DCT-based implementations based on algebraic integer quantization aimed at low-power biomedical applications that not only reduce power and area but also improve image quality that is critical to accurate diagnosis [1], [3], [6], [7].

A. AI-Based DCT for Low-Power Biomedical Video

The work in [7] presents an area and power-efficient implementation of an image compressor for wireless capsule endoscopy application. The architecture uses a direct

Figure 14. Selected frames from 'BasketballPass' test video coded by means of the Chen DCT and the 16-point and 8-point DCT approximations for QP = 0 (a-b-c), QP = 32 (d-e-f), and QP = 50 (g-h-i).

mapping to compute the 2-D DCT, which eliminates the need of transpose operation and results in reduced area and low processing time. The algorithm has been modified to comply with the JPEG standard and the corresponding quantization tables have been developed and the architecture is implemented using the CMOS 0.18 μm technology. The processor [76] costs less than 3.5 k cells, runs at a maximum frequency of 150 MHz, and consumes 10 mW of power. In this work, several GI test color images of 256×256 resolutions have been used for simulation. These images are first transformed and quantized using both the conventional floating point and the JPEG-like AI algorithms. The images are then reconstructed after the de-quantization and the inverse transform. The results and the reconstructed images are shown in Fig. 12. The test results of several endoscopic color images show that higher compression ratio (over 85%) can be achieved with high quality image reconstruction (over 30 dB).

B. Approximate DCT for HEVC

In order to assess the performance of the DCT approximations in real time video coding, the 8 point and the 16 point DCT approximations were embedded into an open source HEVC [140], [141] standard reference software by the Fraunhofer Heinrich Hertz Institute [142]. The original transform prescribed in the selected HEVC reference software is the scaled approximation of Chen DCT algorithm [143] and the software can process image block sizes of 4×4, 8×8, 16×16 and 32×32. Our methodology consists of replacing the 8×8 and the 16×16 DCT algorithms of the reference software by the respective approximate algorithm. The 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 quantization point (QP) was varied from 0 to 50 to obtain the curves and the resulting PSNR values along with the bits/frame values were recorded for both approximation algorithm and Chen DCT algorithm which was already implemented in the reference software.

Fig. 13 depicts the obtained RD curves for the 'BasketballPass' test sequence. Fig. 14 shows particular 416×240 frames for the test video sequence 'BasketballPass' with QP $\in \{0, 32, 50\}$ when the 8-point approximate DCT, 16-point approximate DCT, and the Chen DCT are considered.

VIII. Conclusion

The DCT and DWT are extremely useful transforms having a plethora of applications in image, video, biomedical, and RF signal processing. For example, the superior energy compaction properties of DCT makes it a highly-suitable choice for compression applications including video standards such as HEVC. In this article, we discussed several recent approaches to the digital computation of both DCT and DWT for multiple applications. The review covered mathematical and algorithmic innovations where techniques were sought to reduce the computational complexity, power consumption, and increase throughput, for certain applications. Techniques based algebraic integer and matrix approximation were pursued to furnish a comprehensive set of methods for transform evaluation. The paper covers very low-complexity algorithms that utilize approximation theory as a tool for trading accuracy for circuit complexity and energy efficiency. Various recently proposed VLSI architectures for the DCT and the DWT that are designed for low power applications were briefly covered. Based on algebraic integer representation, the paper also discussed the idea of exact computation. The concept was extended to furnish exact computation for both multi-dimensional as well as multi-rate DCT/DWT structures.

Acknowledgment

The authors at the University of Akron are grateful to the College of Engineering (CoE) for financial support. Dr. Madanayake thanks Xilinx for donating the FPGAs installed in the BEE3 device. He also thanks Dr. Virantha Ekanayake and Greg Martin, both at Achronix Seminconductor, Sujata Mahidara, and Dr. Chen Chang (BeeCube), for technical support and software donations. Dr. Cintra thanks FACEPE and CNPq for partial financial support. Dr. Bayer thanks FAPERGS for partial financial support. Dr. Wahid and Dr. Dimitrov thanks NSERC for financial support. Dr Wahid is grateful to the Canada Foundation for Innovation (CFI) and NSERC.

Arjuna Madanayake (M'03) is a tenure-track assistant professor at the Department of Electrical and Computer Engineering at the University of Akron. He completed both MSc (2004) and PhD (2008) Degrees, in Electrical Engineering, from the University of Calgary, Canada. Dr. Madanayake obtained a BSc in Electronic and Telecommunication Engineering (with First Class Honors) from the University of Moratuwa in Sri Lanka, in 2002. His research interests include multidimensional signal processing, analog/digital and mixed-signal electronics, FPGA systems, and VLSI for fast algorithms.

Renato J. Cintra (SM'2010) received the B.Sc., M.Sc., and D.Sc. degrees in electrical engineering from the Universidade Federal de Pernambuco (UFPE), Recife, Brazil, in 1999, 2001, and 2005, respectively. He joined the Department of Statistics, UFPE, in 2005. In 2008-2009, he spent a sabbatical year with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB, Canada,

as a Visiting Research Fellow. He is associate editor of the IEEE Geoscience and Remote Sensing Letters. He is also a member of AMS and SIAM. His long-term topics of research include theory and methods for digital signal processing, statistical methods, communication systems, and applied mathematics.

Vassil Dimitrov received a PhD degree in applied mathematics from the Mathematical Institute of the Bulgarian Academy of Sciences in 1995. After holding several postdoctoral positions at the University of Windsor, Canada, RST Corporation, USA and Helsinki University of Technology, Finland, between 1996 and 2000, he hold an Associate Professor position at the University of Windsor from 2000 to 2001 and Professor position at the University of Calgary, Canada, since July 2001. His research interests include the design of efficient algorithms for signal and image processing applications, efficient implementations of cryptographic protocols, number theoretic algorithms and related topics. His latest interests include the development of fast parallel algorithms for linear algebra problems.

Fábio Mariano Bayer earned his B.Sc. in Mathematics, M.Sc. in Industrial Engineering from the Federal University of Santa Maria (UFSM), Santa Maria, Brazil, and D.Sc. in Statistics from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2006, 2008, and 2011, respectively. He is currently with the Department of Statistics and the Laboratory of Space Sciences of Santa Maria (LACESM), UFSM. His research interests include statistical computing, parametric inference, and digital signal processing.

Khan A. Wahid (S'02–GS'05–M'07–SM'13) received the B.Sc. degree from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, and the M.Sc. and Ph.D. degrees from the University of Calgary, Calgary, AB, Canada, in 2000, 2003, and 2007, respectively. He has been with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada, since 2007. He has authored and coauthored two book chapters and over 100 peer-reviewed journal and international conference papers in the field of FPGA-based digital system design, real-time video and image processing, real-time embedded systems, and biomedical imaging systems. He is currently a registered Professional Engineer in the Province of Saskatchewan. Dr. Wahid was a recipient of numerous prestigious awards and scholarships, including the Most Distinguished Killam Scholarship and the NSERC Canada Graduate Scholarship for his doctoral research.

Sunera Kulasekera is a MSc. student at the Department of Electrical and Computer Engineering at the University of Akron. He obtained his BSc. degree in Computer Science and Engineering from University of Moratuwa, Sri Lanka. After the bachelor's degree he worked as a Business Analyst at Millennium Information Technologies. His research interests are in the field of digital architectures.

Amila Edirisuriya earned his MSc. in Electrical and Computer Engineering from the University of Akron in 2013. He obtained his B.Sc. degree in Electronic and Telecommunication Engineering with first class honors from University of Moratuwa, Sri Lanka in 2008. His research interests were in the field of digital architectures and analog/digital VLSI design.

Uma Sadhvi Potluri received the B.S. degree in electronics and communications from Jawaharlal Nehru Technological University, Hyderabad, India in 2010. She has completed her M.S in electrical and computer engineering at the University of Akron, Akron, OH. Her research interests include fast algorithms and architectures for DCT, DST and DFT approximations.

Shiva Kumar Madishetty received his masters degree in Electrical Engineering from The University of Akron, OH, USA in 2013 and bachelors degree in Electronics & Communication from Kakatiya University, Warangal, India in 2010. Madishetty is currently working for Laird Technologies, PA, USA. His research interests include Wavelets, DWT, Subband coding, algebraic integers based hardware resource optimizations.

Nilanka Rajapaksha received the B.S. degree (first class hons.) in electronic and telecommunications engineering from the University of Moratuwa, Moratuwa, Sri Lanka. He is currently pursuing his Ph.D. with the Department of Electrical and Computer Engineering, the University of Akron, Akron, OH. He worked as a software tools staff engineer with Tier Logic, Inc. which is a FPGA startup based in Santa Clara, CA. His current research

interests include wave-digital architectures for multidimensional (MD) circuits and systems, and MD array signal processing and architectures for fast algorithm implementation.

References

[1] 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, June 2012.

[2] N. Rajapaksha, A. Edirisuriya, A. Madanayake, R. J. Cintra, D. Onen, I. Amer, and V. S. Dimitrov. (2013). Asynchronous realization of algebraic integer-based 2D DCT using Achronix Speedster SPD60 FPGA. *J. Electr. Comput. Eng.* [Online]. *2013*, pp. 1–9, 2013. Available: http://www.hindawi.com/journals/jece/2013/834793/

[3] S. K. Madishetty, A. Madanayake, R. J. Cintra, V. S. Dimitrov, and D. H. Mugler, "VLSI architectures for the 4-tap and 6-tap 2-D Daubechies wavelet filters using algebraic integers," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 6, pp. 1455–1468, June 2013.

[4] 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," *Measure. Sci. Technol.*, vol. 23, no. 8, p. 114010, 2012.

[5] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, and N. Rajapaksha. (2012). Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. *Measure. Sci. Technol.* [Online]. *23(11)*, p. 114003. Available: http://stacks.iop.org/0957-0233/23/i=11/a=114003

[6] K. A. Wahid, M. A. Islam, and S. Ko, "Lossless implementation of Daubechies 8-tap wavelet transform," in *Proc. IEEE Int. Symp. Circuits Systems (ISCAS)*, 2011, pp. 2157–2160.

[7] K. Wahid, S. Ko, and D. Teng, "Efficient hardware implementation of an image compressor for wireless capsule endoscopy applications," in *Proc. IEEE Int. Joint Conf. Neural Networks, IJCNN (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2761–2765.

[8] C. Lian, Y. Huang, H. Fang, Y. Chang, and L. Chen, "JPEG, MPEG-4, and H.264 codec IP development," in *Proc. Design, Automation, Test in Europe*, Mar. 2005, vol. 2, pp. 1118–1119.

[9] J. F. Blinn, "What's that deal with the DCT?" *IEEE Comput. Graph. Appl.*, vol. 13, no. 4, pp. 78–83, July 1993.

[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*. San Diego, CA: Academic Press, 2007.

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

[13] R. J. Clarke, "Relation between the Karhunen-Loève and cosine transforms," *IEEE Proc. F Commun., Radar Signal Process.*, vol. 128, no. 6, pp. 359–360, Nov. 1981.

[14] J. Liangand T. D. Tran, "Fast multiplierless approximation of the DCT with the lifting scheme," *IEEE Trans. Signal Processing*, vol. 49, pp. 3032–3044, 2001.

[15] T. I. Haweel, "A new square wave transform based on the DCT," *Signal Process.*, vol. 82, pp. 2309–2319, 2001.

[16] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards*. Boston, MA: Kluwer, 1997.

[17] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, no. 1, pp. 18–34, 1992.

[18] N. Roma and L. Sousa, "Efficient hybrid DCT-domain algorithm for video spatial downscaling," *EURASIP J. Adv. Signal Process.*, 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] International Telecommunication Union, "ITU-T recommendation H.263 version 1: Video coding for low bit rate communication," ITU-T, Tech. Rep., 1995.

[22] 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, July 2003.

[23] 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 Electron. Mag.*, vol. 1, no. 3, pp. 36–46, July 2012.

[24] I. E. Richardson, *The H.264 Advanced Video Compression Standard*. Hoboken, NJ: Wiley, 2011.

[25] J. Lee and H. Kalva, *The VC-1 and H.264 Video Compression Standards for Broadband Video Services*. New York: Springer, 2008.

[26] M. Mehrabi, F. Zargari, and M. Ghanbari, "Fast and low complexity method for content accessing and extracting DC-pictures from H.264 coded videos," *IEEE Trans. Consumer Electron.*, vol. 56, no. 3, pp. 1801–1808, 2010.

[27] Z. He and M. Bystrom, "Improved conversion from DCT blocks to integer cosine transform blocks in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 851–857, 2008.

[28] J. Moon and J. Kim, "A new low-complexity integer distortion estimation method for H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 207–212, 2010.

[29] G. J. Sullivan, J. Ohm, W. Han, 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.

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

[31] J. Park, W. Nam, S. Han, and S. Lee, "2-D large inverse transform (16 × 16, 32 × 32) for HEVC (High Efficiency Video Coding)," *J. Semicond. Technol. Sci.*, vol. 2, pp. 203–211, 2012.

[32] S. Y. Park and P. K. Meher, "Flexible integer DCT architectures for HEVC," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2013, pp. 1376–1379.

[33] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-point approximate DCT for image and video compression requiring only 14 additions," *IEEE Trans. Circuits Syst. I*, to be published.

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

[35] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoustic, Signal, Speech Processing*, vol. 6, no. 10, pp. 1455–1461, 1987.

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

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

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

[39] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1D DCT algorithms with 11 multiplications," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1989, pp. 988–991.

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

[41] B. Vasudev and N. Merhav, "DCT mode conversions for field/frame coded MPEG video," in *Proc. IEEE Second Workshop Multimedia Signal Processing*, Dec. 1998, pp. 605–610.

[42] M. C. Lin, L. R. Dung, and P. K. Weng. (2006, Feb.). An ultra-low-power image compressor for capsule endoscope. *BioMedical Engineering OnLine*. [Online]. 5(1), pp. 1–8. Available: http://dx.doi.org/10.1186/1475-925X-5-14

[43] A. Edirisuriya, A. Madanayake, V. Dimitrov, R. J. Cintra, and J. Adikari. (2012). VLSI architecture for 8-point AI-based Arai DCT having low area-time complexity and power at improved accuracy. *J. Low Power Electron. Appl.* [Online]. 2(2), pp. 127–142. Available: http://www.mdpi.com/2079-9268/2/2/127/

[44] 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 *Proc. IEEE Int. Symp. Circuits Systems (ISCAS)*, 2011, pp. 1247–1250.

[45] H. Meng, Z. Wang, and G. Liu, "Performance of the Daubechies wavelet filters compared with other orthogonal transforms in random signal processing," in *Proc. 5th Int. Conf. Signal Processing Proc., WCCC-ICSP*, 2000, vol. 1, pp. 333–336.

[46] J. H. Park, K. O. Kim, and Y. K. Yang, "Image fusion using multiresolution analysis," in *Proc. IEEE Int. Geoscience Remote Sensing Symp*., Sydney, Australia, 2001, vol. 2, pp. 864–866.

[47] S. Saha. (2000, Mar.). Image compression—From DCT to wavelets: A review. *Crossroads*. [Online]. *6(3)*, pp. 12–21. Available: http://doi.acm.org/10.1145/331624.331630

[48] J. P. Andrew, P. O. Ogunbona, and F. J. Paoloni, "Comparison of "wavelet" filters and subband analysis structures for still image compression," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing, ICASSP*, 1994.

[49] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, *Wavelet Toolbox User's Guide*. Mathworks, Inc., 2011.

[50] J. Walker, *A Primer on Wavelets and their Scientific Applications*. Boca Raton, FL: CRC Press, 1999.

[51] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, "A new DCT algorithm based on encoding algebraic integers," in *Proc. 1998 IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 1998, vol. 3, pp. 1377–1380.

[52] K. Wahid, "Error-free implementation of the discrete cosine transform," Ph.D. dissertation, Univ. Calgary, 2010.

[53] P. K. Meher, "Unified DA-based parallel architecture for compting the DCT and the DST," in *Proc. 5th IEEE Int. Conf. Information, Communications, Signal Processing*, 2005, pp. 1278–1282.

[54] S. S. Nayak and P. K. Meher, "3-dimensional systolic architecture for parallel VLSI implementation of the discrete cosine transform," *IEE Proc. Circuits Devices Syst*., vol. 143, no. 5, pp. 255–258, Oct. 1996.

[55] A. Madisetti and A. N. Willson, "A 100 MHz 2-D 8 × 8 DCT-IDCT processor for HDTV applications," *IEEE Trans. Circuits Syst. Video Technol*., vol. 5, no. 2, pp. 158–165, Apr. 1995.

[56] T. Sung, Y. Shieh, and H. Hsin, "An efficient VLSI linear array for DCT/IDCT using subband decomposition algorithm," *Math. Probl. Eng*., vol. 2010, pp. 1–21, 2010.

[57] H. Huang, T. Sung, and Y. Shieh, "A novel VLSI linear array for 2-D DCT-IDCT," in *Proc. 2010 3rd Int. Congress Image Signal Processing (CISP)*, 2010, pp. 3686–3690.

[58] J. Guo, R. Ju, and J. Chen, "An efficient 2-D DCT/IDCT core design using cyclic convolution and adder-based realization," *IEEE Trans. Circuits Syst. Video Technol*., vol. 14, no. 4, pp. 416–428, Apr. 2004.

[59] Y. Chen and T. Chang, "A high performance video transform engine by using space-time scheduling stratergy," *IEEE Trans. Very Large Scale Integration (VLSI) Syst*., 2011.

[60] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, "Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT-DST-IDCT-IDST," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 6, pp. 1125–1137, June 2005.

[61] A. Tumeo, M. Monchiero, G. Palermo, F. Ferrandi, and D. Sciuto, "A pipelined fast 2D-DCT accelerator for FPGA-based SoCs," in *Proc. IEEE Computer Society Annual Symp. VLSI (ISVLSI)*, 2007.

[62] C. Sun, P. Donner, and J. Gotze, "Low-complexity multi-purpose IP core for quantized discrete cosin and integer transform," in *Proc. 2009 IEEE Int. Symp. Circuits Systems (ISCAS)*, 2009, pp. 3014–3017.

[63] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high-performance DCT architecture," *IEEE Trans. Signal Processing*, vol. 3, no. 3, pp. 955–964, Mar. 2006.

[64] C. Lin, Y. Yu, and L. Van, "Cost-effective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor," *IEEE Trans. Very Large Scale Integration (VLSI) Syst*., vol. 16, no. 8, pp. 1058–1071, Aug. 2008.

[65] C. Huang, L. Chen, and Y. Lai, "A high-speed 2-D transform architecture with unique kernel for multi-standard video applications," in *Proc. IEEE 2008 Int. Symp. Circuits Systems (ISCAS)*, 2008, pp. 21–24.

[66] J. H. Cozzens and L. A. Finkelstein, "Range and error analysis for a fast Fourier transform computed over $Z[\omega]$," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 4, pp. 582–590, July 1987.

[67] M. Fu, V. S. Dimitrov, and G. Jullien, "An efficient technique for error-free algebraic-integer encoding for high performance implementation of the DCT and IDCT," in *Proc. IEEE 2001 Int. Symp. Circuits Systems (ISCAS)*, 2001.

[68] M. Fu, G. Jullien, V. S. Dimitrov, M. Ahmadi, and W. C. Miller, "Implementation of an error-free DCT using algebraic integers," in *Proc. Micronet Annual Workshop*, 2002.

[69] M. Fu, G. Jullien, V. S. Dimitrov, M. Ahmadi, and W. C. Miller, "The application of 2D algebraic integer encoding to a DCT IP core," in *Proc. 3rd IEEE Int. Workshop System-on-Chip Real-Time Applications*, Calgary, AB, Canada, 2002, pp. 66–69.

[70] M. Fu, G. A. Jullien, V. S. Dimitrov, and M. Ahmadi, "A low-power DCT IP core based on 2D algebraic integer encoding," in *Proc. Int. Symp. Circuits Systems (ISCAS)*, May 2004, vol. 2, pp. 765–768.

[71] M. Fu, G. Jullien, and M. Ahmadi. (2004, Feb.). Algebraic integer encoding and applications in discrete cosine transform. [Online]. Gennum Presentation by Dept. ECE, University of Windsor, Canada. Available: http://www.docstoc.com/docs/74259941/Algebraic-Integer-Encoding-and-Applications-in-Discrete-Cosine

[72] V. Dimitrov, K. Wahid, and G. Jullien, "Multiplication-free 8 × 8 2D DCT architecture using algebraic integer encoding," *IEE Electron. Lett*., vol. 40, no. 20, pp. 1310–1311, 2004.

[73] V. Dimitrov and K. Wahid, "On the error-free computation of fast cosine transform," *Int. J.: Inform. Theories Appl*., vol. 12, no. 4, pp. 321–327, 2005.

[74] K. Wahid, S. B. Ko, and V. S. Dimitrov, "Area and power efficient video compressor for endoscopic capsules," in *Proc. 4th Int. Conf. Biomedical Engineering*, Kuala Lumpur, June 2008

[75] K. Wahid, "An efficient IEEE-compliant 8 × 8 Inv-DCT architecture with 24 adders," *IEEE Trans. Electron., Inform. Syst*., vol. 131, pp. 1081–1082, 2011.

[76] T. H. Khan and K. A. Wahid, "Lossless and low-power image compressor for wireless capsule endoscopy," *VLSI Design*, vol. 2011, p. 12, 2011.

[77] K. A. Wahid, M. Martuza, M. Das, and C. McCrosky, "Efficient hardware implementation of 8 × 8 integer cosine transforms for multiple video codecs," *J. Real-Time Process*., pp. 1–8, July 2011.

[78] S. Nandi, K. Rajan, and P. Biswas, "Hardware implementation of 4 × 4 DCT quantization block using multiplication and error-free algorithm," in *Proc. IEEE TENCON Region 10*, 2009, pp. 1–5.

[79] A. Edirisuriya, A. Madanayake, R. Cintra, V. Dimitrov, and N. Rajapaksha, "A single-channel architecture for algebraic integer-based 8 × 8 2-D DCT computation," *IEEE Trans. Circuits Syst. Video Technol*., vol. 23, no. 12, pp. 2083–2089, Dec. 2013.

[80] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 4th ed. London: Oxford Univ. Press, 1975.

[81] M. E. Pohst, *Computational Algebraic Number Theory*. Basel, Switzerland: Birkhäuser Verlag, 1993.

[82] H. Pollard and H. G. Diamond, *The Theory of Algebraic Numbers* (No. 9: The Carus Mathematical Monographs, ser), 2nd ed. The Mathematical Association of America, 1975.

[83] R. Baghaie and V. Dimitrov, "Systolic implementation of real-valued discrete transforms via algebraic integer quantization," *Comput. Math. Appl*., vol. 41, pp. 1403–1416, 2001.

[84] J. H. Cozzens and L. A. Finkelstein, "Computing the discrete Fourier transform using residue number systems in a ring of algebraic integer," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 5, pp. 580–588, Sept. 1985.

[85] R. E. Blahut, *Fast Algorithms for Signal Processing*. Cambridge, England: Cambridge Univ. Press, 2010.

[86] V. Dimitrov, G. A. Jullien, and W. C. Miller, "Eisenstein residue number system with applications to DSP," in *Proc. 40th Midwest Symp. Circuits Systems*, 1997.

[87] 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 *Proc. IEEE Int. Conf. Microelectronic Systems Education*, June 2007, pp. 173–174.

[88] K. Wahid, V. Dimitrov, and G. Jullien, "VLSI architectures of Daubechies wavelets for algebraic integers," *J. Circuits, Syst., Comput*., vol. 13, no. 6, pp. 1251–1270, 2004.

[89] K. A. Wahid, V. S. Dimitrov, G. A. Jullien, and W. Badawy, "An algebraic integer based encoding scheme for implementing Daubechies discrete wavelet transforms," in *Proc. Asilomar Conf. Signals, Systems and Computers*, 2002, vol. 1, pp. 967–971.

[90] K. A. Wahid, V. S. Dimitrov, G. A. Jullien, and W. Badawy, "An analysis of Daubechies discrete wavelet transform based on algebraic integer encoding scheme," in *Proc. 3rd Int. Workshop Digital and Computational Video, DCV*, 2002, pp. 27–34.

[91] G. Xing, J. Li, S. Li, and Y.-Q. Zhang, "Arbitrarily shaped video-object coding by wavelet," *IEEE Trans. Circuits Syst. Video Technol*., vol. 11, no. 10, pp. 1135–1139, 2001.

[92] Y. Wu, R. J. Veillette, D. H. Mugler, and T. T. Hartley, "Stability analysis of wavelet-based controller design," in *Proc. American Control Conf*., 2001, vol. 6, pp. 4826–4827.

[93] K. Wahid, V. Dimitrov, and G. Jullien, "Error-free arithmetic for discrete wavelet transforms using algebraic integers," in *Proc. 16th IEEE Symp. Computer Arithmetic (ARITH-16)*, IEEE Comput. Soc., 2003, p. 238.

[94] S. C. B. Lo, H. Li, and M. T. Freedman, "Optimization of wavelet decomposition for image compression and feature preservation," *IEEE Trans. Med. Imag.*, vol. 22, no. 9, pp. 1141–1151, Sept. 2003.
[95] M. Martone, "Multiresolution sequence detection in rapidly fading channels based on focused wavelet decompositions," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1388–1401, Aug. 2001.
[96] P. P. Vaidyanathan, *Multirate Systems and Filter Basnks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[97] D. B. H. Tay, "Balanced spatial and frequency localised 2-D nonseparable wavelet filters," in *Proc. IEEE Int. Symp. Circuits Systems, ISCAS*, 2001, vol. 2, pp. 489–492.
[98] M. A. Islam and K. A. Wahid, "Area-and power-efficient design of Daubechies wavelet transforms using folded AIQ mapping," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 9, pp. 716–720, Sept. 2010.
[99] F. Marino, D. Guevorkian, and J. T. Astola, "Highly efficient high-speed/low-power architectures for the 1-D discrete wavelet transform," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 12, pp. 1492–1502, Dec. 2000.
[100] K. A. Wahid, V. S. Dimitrov, G. A. Jullien, and W. Badawy, "Error-free computation of Daubechies wavelets for image compression applications," *Electron. Lett.*, vol. 39, no. 5, pp. 428–429, 2003.
[101] T. Acharya and P.-Y. Chen, "VLSI implementation of a DWT architecture," in *Proc. IEEE Int. Symp. Circuits Systems, (ISCAS)*, 1998, vol. 2, pp. 272–275.
[102] S. Gnavi, B. Penna, M. Grangetto, E. Magli, and G. Olmo, "DSP performance comparison between lifting and filter banks for image coding," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing (ICASSP) Conf.*, 2002, vol. 3.
[103] A. M. Mountassar Maamoun, M. Neggazi, and D. Berkani, "VLSI design of 2-D disceret wavelet transform for area efficient and high-speed image computing," *World Acad. Sci., Eng. Technol.*, vol. 45, pp. 538–543, 2008.
[104] I. Urriza, J. I. Artigas, J. I. Garcia, L. A. Barragan, and D. Navarro, "VLSI architecture for lossless compression of medical images using the discrete wavelet transform," in *Proc. Design, Automation and Test in Europe*, 1998, pp. 196–201.
[105] R. Baghaie and V. Dimitrov, "Computing Haar transform using algebraic integers," in *Proc. Conf Signals, Systems, Computers Record 34th Asilomar Conf.*, 2000, vol. 1, pp. 438–442.
[106] D. B. H. Tay, "Integer wavelet transform for medical image compression," in *Proc. Intelligent Information Systems Conf. 7th Australian New Zealand*, 2001, pp. 357–360.
[107] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sept. 2001.
[108] B. K. Mohanty, A. Mahajan, and P. K. Meher, "Area and power efficient architecture for high-throughput implementation of lifting 2-D DWT," *IEEE Trans. Circuits. Syst. II*, vol. 59, no. 7, pp. 434–438, July 2012.
[109] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[110] G. Dimitroulakos, M. D. Galanis, A. Milidonis, and C. E. Goutis, "A high-throughput and memory efficient 2D discrete wavelet transform hardware architecture for JPEG2000 standard," in *Proc. IEEE Int. Symp. Circuits and Systems, ISCAS*, 2005, pp. 472–475.
[111] S. Mallat, *A Wavelet Tour of Signal Processing*. Burlington, MA: Academic, 2008.
[112] D. B. H. Tay and N. G. Kingsbury, "Design of nonseparable 3-D filter banks/wavelet bases using transformations of variables," *IEEE Proc. Visual Image Signal Process.*, vol. 143, pp. 51–61, 1996.
[113] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, England: Cambridge Univ. Press, 1999.
[114] K. A. Wahid, *Low Complexity Implementation of Daubechies Wavelets for Medical Imaging Applications*. InTech, 2011.
[115] S. Madishetty, A. Madanayake, R. Cintra, V. Dimitrov, and D. Mugler. (2013). Algebraic integer architecture with minimum adder count for the 2-d daubechies 4-tap filters banks. *Multidimensional Syst. Signal Process.* [Online]. pp. 1–17. Available: http://dx.doi.org/10.1007/s11045-013-0245-4
[116] S. Athar and O. Gustafsson, "Optimization of AIQ representations for low complexity wavelet transforms," in *Proc. 20th European Conf. Circuit Theory and Design (ECCTD)*, 2011, pp. 314–317.
[117] S. Madishetty, A. Madanayake, R. Cintra, and V. Dimitrov, "Precise VLSI architecture for ai based 1-d/2-d daub-6 wavelet filter banks with low adder-count," *IEEE Trans. Circuits Syst. I*, vol. PP, no. 99, pp. 1–1, 2014.
[118] D. Quick. (2013, Sept.). Japanese broadcaster looks beyond 4k to 8k. *GizMag*. [Online]. Available: http://www.gizmag.com/ nhk-8k-shv/29076/
[119] D. D. McAdams. (2013, May). NHK and mitsubishi develop hevc encoder for 8k 'super hi-vision' tv. *TV Technology*. [Online]. Available: http://www.tvtechnology.com/article/nhk-and-mitsubishi-develop-hevc-encoder-for-k-super-hi-vision-tv/219328
[120] M. T. Heideman and C. S. Burrus. (1988). *Multiplicative Complexity, Convolution, and the DFT*, ser. Signal Processing and Digital Filtering. Springer-Verlag. [Online]. Available: http://books.google.com.br/books?id=QxWoAAAAIAAJ
[121] F. M. Bayer and R. J. Cintra, "DCT-like transform for image compression requires 14 additions only," *Electron. Lett.*, vol. 48, no. 15, pp. 919–921, 2012.
[122] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-complexity 8 $\times$ 8 transform for image compression," *Electron. Lett.*, vol. 44, no. 21, pp. 1249–1250, Sept. 2008.
[123] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A low-complexity parametric transform for image compression," in *Proc. IEEE Int. Symp. Circuits Systems (ISCAS)*, 2011.
[124] 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.
[125] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A multiplication-free transform for image compression," in *Proc. 2nd Int. Conf. Signals, Circuits and Systems*, Nov. 2008, pp. 1–4.
[126] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A fast 8 $\times$ 8 transform for image compression," in *Proc. Int. Conf. Microelectronics (ICM)*, Dec. 2009, pp. 74–77.
[127] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A novel transform for image compression," in *Proc. 53rd IEEE Int. Midwest Symp. Circuits Systems (MWSCAS)*, Aug. 2010, pp. 509–512.
[128] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Binary discrete cosine and Hartley transforms," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 4, pp. 989–1002, 2013.
[129] R. J. Cintra and F. M. Bayer, "A DCT approximation for image compression," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 579–583, Oct. 2011.
[130] W. Kuo and K. Wu. (2011, Dec.). Traffic prediction and QoS transmission of real-time live VBR videos in WLANs. *ACM Trans. Multimedia Comput., Commun. Appl.* [Online]. *7(4)*, pp. 36:1–36:21. Available: http://doi.acm.org/10.1145/2043612.2043614
[131] S. Saponara. (2012). Real-time and low-power processing of 3D direct/inverse discrete cosine transform for low-complexity video codec. *J. Real-Time Image Process.* [Online]. 7, pp. 43–53. Available: http://dx.doi.org/10.1007/s11554-010-0174-5
[132] V. Lecuire, L. Makkaoui, and J. M. Moureaux, "Fast zonal DCT for energy conservation in wireless image sensor networks," *Electron. Lett.*, vol. 48, no. 2, pp. 125–127, 2012.
[133] R. J. Cintra. (2011). An integer approximation method for discrete sinusoidal transforms. *J. Circuits, Syst., Signal Process*. [Online]. *30(6)*, pp. 1481–1501. Available: http://www.springerlink.com/content/nw5u0267254t3683/
[134] G. A. F. Seber, *A Matrix Handbook for Statisticians*. Hoboken, NJ: Wiley, 2008.
[135] N. J. Higham. (1987). Computing real square roots of a real matrix. *Linear Algebra and its Appl.* [Online]. *88/89*, pp. 405–430. Available: http://www.sciencedirect.com/science/article/pii/0024379587901182
[136] MATLAB, *version 8.1 (R2013a) Documentation*. Natick, MA: The MathWorks Inc., 2013.
[137] R. J. Cintra, F. M. Bayer, and C. J. Tablada, "Low-complexity 8-point DCT approximations based on integer functions," *Signal Process.*, vol. 99, pp. 201–214, 2014.
[138] A. V. Oppenheim, *Discrete-Time Signal Processing*. Pearson, 2006.
[139] D. S. Watkins. (2004). *Fundamentals of Matrix Computations* (ser. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts). Wiley. [Online]. Available: http://books.google.es/books?id=xi5omWiQ-3kC
[140] R. Wilson. The TV studio becomes a system. Altera Corporation. [Online]. Available: http://www.altera.com/technology/system-design/articles/2013/tv-studio-system.html.
[141] P. Frojdh, A. Norkin, and R. Sjoberg. (2013, Apr.). Next generation video compression. *Ericsson Rev.* [Online]. Available: http://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2013/er-hevc-h265.pdf
[142] Joint Collaborative Team on Video Coding (JCT-VC). (2013). HEVC references software documentation. Fraunhofer Heinrich Hertz Institute. [Online]. Available: https://hevc.hhi.fraunhofer.de/
[143] W. H. Chen, C. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sept. 1977.