Received September 18, 2019, accepted October 10, 2019, date of publication October 14, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947269

Hardware Efficient Integer Discrete Cosine Transform for Efficient Image/Video Compression

JIAJIA CHEN[1], SHUMIN LIU[2], GELEI DENG[2], AND SUSANTO RAHARDJA[3], (Fellow, IEEE)

[1]College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2]Department of Engineering Product Development, Singapore University of Technology and Design, Singapore
[3]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Susanto Rahardja (susantorahardja@ieee.org)

This work was supported by the Nanjing City Qualified Research, Nanjing University of Aeronautics and Astronautics, Nanjing, China, under Grant MCA19005 and Grant 56SYAH18043.

ABSTRACT With the introduction of high efficiency video coding (HEVC) standard which provides super compression efficiency, there has been a lot of research works on integer transform matrices that can provide good approximation to the discrete cosine transform (DCT) used in HEVC. Not only maintaining the coding performance, the hardware and power of the circuit to implement the derived integer DCT (Int-DCT) needs to be minimized. To address these multiple design considerations, a new multi-objective optimization algorithm is proposed in this paper to search for efficient Int-DCT matrix, which has the coding performance as close as possible to the transform in HEVC but implemented with reduced hardware and power. Experimental results show that the approximated Int-DCT matrix generated by the proposed algorithm can achieve almost the same coding performance as the transforms in HEVC measured in terms of BjØntegaard Delta rate. Meanwhile, the experiments demonstrate that the proposed 16-point Int-DCT can produce at least 15.5% and 26.8% lower circuit area in FPGA and ASIC respectively, compared with other state-of-the-art Int-DCT realizations which can provide similar coding performance.

INDEX TERMS DCT, image/video coding and compression, digital signal processing.

I. INTRODUCTION

Discrete cosine transform (DCT) is commonly used for image and video compression [1] such as those in published standards like Joint Photographic Expert Group (JPEG), Moving Picture Experts Group (MPEG) and International Telecommunication Union Telecommunication (ITU.T) standards. Because exact DCTs are very close to the theoretical DCT complexity, they could hardly offer dramatic computational gains and implementation cost reduction [2]. Therefore, approximate DCTs become an alternative to reduce the computational complexity. If the basis properties of the transform matrix such as orthogonality, symmetry and equal norm can be preserved, transform approximations can be applied to reduce the computational cost [3]–[6]. Integer discrete cosine transform (Int-DCT) [7] is one of the approximations whose finite precision transform coefficients can be computed with integer arithmetic. Compared to exact DCT, it has a lower computation cost and causes no drifting error [8]. As a result, it has been used in the recent coding standards like H.264/Advanced Video Coding (AVC) [9], [10], Audio and Video Streaming (AVS) [11], Video Codec 1 (VC-1) [12], and high efficiency video coding (HEVC) [13], which uses 4-point to 32-point Int-DCT.

The most straightforward way of deriving Int-DCT coefficient is scaling the DCT coefficients by a factor, followed by rounding to integer values [14]. However, the derived coefficients by this simple scaling cannot guarantee good coding performance. Therefore, a number of Int-DCTs implementations are proposed in the literature which can be used in the core transform matrices of HEVC. Variable block-size transform (VBT) [15] is one of them which uses 4-point to 32-point Int-DCT adaptively. This method selects similar blocks from the reconstructed area and uses them to derive the Karhunen-Loeve transform. The VBT can adapt to the non-stationary video signals and can improve the coding performance. Cintra et. al. has contributed by developing a sequence of approximated 4-point and 8-point DCTs [2], [3], [16]–[21]. In [2], a new class of matrices based on a parametrization of the Feig–Winograd factorization of 8-point DCT is proposed. In [16], two multiplierless algorithms are proposed to develop 2D 4-point DCT approximations for coding in digital video.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Kamrul Hasan.

In [3], the authors introduced low-complexity 3D 8-point DCT approximations which are formalized in terms of high-order tensor theory. Other 8-point approximated DCTs are proposed in [17]–[21] with different techniques to derive efficient transforms with lower number of required additions.

In addition, more transforms have been proposed [22]–[24] for 16-point and 32-point Int-DCT used in HEVC, because larger transforms such as 16-point DCT can contribute more coding gains compared with 4-point or 8-point transforms [25]. In [22], Cintra's team proposed a digital very large scale integration (VLSI) architecture for computing DCT/DST transforms without multiplications. The proposed 16×16 transform are heavily approximated to make the transform matrix consisting of 1, 0 and -1 only. This helps to minimize the hardware cost to implement this transform, but suffers from much bigger errors. In [23], the method derives orthogonal and high order Int-DCT using the lower order transforms. In [24], the proposed design can ensure a fully factorized structure and the computation is fast. Another method which can be included into HEVC standards is Joint Collaborative Team on Video Coding (JCTVC)-G579 [26] which uses scaled integer transforms and supports recursive factorization.

In recent years, a group of new Int-DCT are proposed. One of them is the recursive integer cosine transform (RICT) proposed in [8]. RICT is a method to generate high order Int-DCT using lower orders by utilizing the self-recursive property of DCT transform. Compared with JCTVC-G579 [26], basis row vectors in RICT have almost the same norm, so additional scaling is avoided. Another representative algorithm was proposed in [27] to derive scalable and orthogonal approximation of DCT. An approximate DCT of length N is derived from a pair of DCTs of length $N/2$ at the cost of additions for input preprocessing. Another more recent algorithm proposed by the authors of [27] was published in [28], where an approximated kernel for DCT of length 4 is derived. This kernel is adopted for the computation of DCT and IDCT of higher order transforms whose sizes are power-of-two numbers. Another approximated DCT for HEVC was proposed in [29]. The DCT is implemented through the Walsh–Hadamard transform followed by Givens rotations. The proposed method computes four different approximations and skip some rotations. To our best knowledge, one of the most recently proposed algorithms is the design presented in [30] which is also relevant to our proposed work. In [30], an energy- and area-efficient architecture for approximated DCT is proposed. It achieves good compression performance with reduced computation cost by truncating a couple of least significant bits (LSB), most significant bits (MSB), and some zero columns. Another design was proposed recently in [31] by Chen et. al. Compared with this paper, the main contribution in [31] is a new efficient DCT circuit implementation by using double base number system and an algorithm to minimize distinct shift counts. The design relies on existing Int-DCT coefficients for circuit implementation. There is no approximation made to the given Int-DCT coefficients.

Although these works have contributed significantly in developing low cost Int-DCT matrices with good compression efficiency, given the demanding hardware and power requirements in emerging technologies, there is a continuous need to improve the performance. Therefore, it is meaningful to develop new Int-DCT matrices for HEVC which can lead to good compression performance and at the same time achieve lower hardware cost. Given a good Int-DCT requires multiple properties such as orthogonality, basis vectors norm uniqueness and basis vectors energy compaction, a good optimization approach that can provide the flexibility to adjust the priorities of the above-mentioned measures is necessary. In this paper, we propose a new algorithm to generate power-of-two points Int-DCT, which include three main contributions:

1. We solve the Int-DCT coefficients approximation as a multi-objective optimization problem where the objectives are the critical properties of Int-DCT matrix.
2. We develop the hardware efficient solutions by optimizing the hardware cost and the coding performance measures simultaneously, as the two objectives.
3. We normalize the Int-DCT matrix properties and solve the optimization problem by using a weighted sum approach where the weights for each objectives are adjustable according to the objective priorities.

The experimental results have shown that the proposed Int-DCT can achieve almost the same compression performance as the transform in HEVC measured by Bjøntegaard Delta rate (BD-rate) [32]. In addition, the hardware cost to implement the proposed Int-DCT is significantly reduced. This paper is organized as follows. Section II explains the Int-DCT design criteria related to compression performance and implementation cost. Section III presents the proposed algorithm to derive the efficient Int-DCT with low hardware complexity. The experimental results and discussions are presented in Section IV and the paper is concluded in Section V.

II. DCT PROPERTIES AND HARDWARE COST EVALUATION

A. DCT-II TRANSFORMS

In a typical video codec, an $N \times N$ forward DCT and an $N\times N$ inverse DCT are usually required. The DCT can be categorized into four types, namely DCT-I to DCT-IV [33]. In this paper, we focus on new Int-DCT matrices used for Type II forward transform. The original 2-dimension $N \times N$ forward Type II DCT can be expressed as

$$F(u,v) = \sqrt{\frac{2}{N}}\sum_{i=0}^{N-1}\Lambda(u)\cos\left[\frac{\pi u}{2N}(2i+1)\right] \times \left[\sqrt{\frac{2}{N}}\sum_{j=0}^{N-1}\Lambda(v)\cos\left[\frac{\pi v}{2N}(2j+1)\right]S(i,j)\right] \quad (1)$$

where $\Lambda(\xi)=\begin{cases} 1/\sqrt{2} & \text{if } \xi=0 \\ 1 & \text{otherwise} \end{cases}$, $u, v \in [0, N-1]$. $F(u, v)$ is the frequency domain output and $S(i, j)$ is the spatial domain input. This can be implemented as a separable transform by applying 1-dimension N-point DCT to each row and each column separately [14]. 1-dimension N-point type-II DCT can be expressed as [15]:

$$y(n) = \mathbf{C}_N^{II} x(n) \tag{2}$$

where the input sequence $x(n) = [x(0), x(1), \ldots, x(N-1)]^T$ and the output $y(n) = [y(0), y(1), \ldots, y(N-1)]^T$. $\mathbf{C}_N^{II}$ is the DCT transform matrix whose infinite precision entries can be computed as

$$\mathbf{C}_N^{II}(m, n) = \sqrt{\frac{2}{N}}\Lambda(m)[\cos\frac{\pi m}{2N}(2n+1)] \tag{3}$$

where $m \in [0, N-1]$ and $n \in [0, N-1]$ are the row and column index of the matrix respectively.

B. TRANSFORM QUALITY EVALUATION BY CORE PROPERTIES

The DCT matrices used in HEVC are essentially finite precision approximations of infinite precision DCT matrix computed by (3). These transform matrices in the standard can help to avoid encoder-decoder mismatch and drift caused by implementations with different floating point representations [14]. Int-DCT has a few core properties which can be used to measure its compression quality. These properties include basis vectors symmetry, orthogonality, closeness to the original DCT, basis vectors norm equality, and so on. The symmetry property of the derived Int-DCT is always preserved, because the repeated coefficients provided by the symmetry are helpful to reduce the number of arithmetic operations. Some other properties such as closeness to the original DCT and transform matrix basis vectors orthogonality [14] are also critical to achieve good compression efficiency while others have impact on quantization/de-quantization process such as basis vectors norm equality.

In the proposed method, the first property evaluated is the Closeness to original DCT, which can be computed by

$$\mathbf{Close}(m, n) = \left|\alpha\mathbf{C}_N^{II}(m, n) - \mathbf{d}(m, n)\right| / \mathbf{d}(0, 0) \tag{4}$$

where $\mathbf{d}(m,n)$ is the entry at the mth row and nth column of the derived finite precision Int-DCT matrix $\mathbf{d}$ and α is a scaling factor. $\mathbf{Close}(m,n)$ is the matrix to store the closeness between each element in $\mathbf{d}(m,n)$ with their value in the original DCT matrix. For Type II DCT as in (3), we scale and truncate the entries in the first row vector, i.e. $\mathbf{C}_N^{II}(0, n)$, from its original value of $1/\sqrt{N}$ to be 2^B, where B is the wordlength we use to represent the finite precision coefficients. In this case, all the first row coefficients become power-of two-integer and in hardware realization the hardwire-free shifts can be used to multiply the first row vector with the corresponding inputs. This makes the scaling factor $\alpha = 2^B\sqrt{N}$ which is applied to other row vectors in $\mathbf{C}_N^{II}$. With α specified, we can evaluate **Close** of any given Int-DCT $\mathbf{d}$ using (4). Therefore, the average closeness, denoted as $C_A(\mathbf{d})$, for the entire Int-DCT matrix $\mathbf{d}$ can be computed by

$$C_A(\mathbf{d}) = \frac{1}{N^2}\sum_{\forall m,n}\mathbf{Close}(m, n). \tag{5}$$

Lower value of C_A indicates the derived Int-DCT matrix $\mathbf{d}$ is closer to the scaled DCT matrix $\alpha\mathbf{C}_N^{II}$.

Secondly, the basis vectors need to be orthogonal. This property makes the transform coefficients to be uncorrelated which is essential for good compression efficiency. Let $\mathbf{v}_{r,m}$ and $\mathbf{v}_{c,n}$ be the mth row vector and the nth column vector in $\mathbf{d}$ respectively. For any two different row vectors in $\mathbf{d}$, $\mathbf{v}_{r,m1} = [d(m_1, 0), d(m_1, 1),\ldots, d(m_1, N-1)]^T$ at row m_1 and $\mathbf{v}_{r,m2} = [d(m_2,0), d(m_2,1),\ldots, d(m_2, N-1)]^T$ at row m_2, the orthogonality between them is given as

$$orth\_R(m_1, m_2) = |\mathbf{v}_{r,m1}^T\mathbf{v}_{r,m2}/\mathbf{v}_{r,0}^T\mathbf{v}_{r,0}|. \tag{6}$$

Similarly, for any two different column vectors in $\mathbf{d}$, $\mathbf{v}_{c,n1} = [d(n_1,0), d(n_1,1),\ldots, d(n_1, N-1)]^T$ at column n_1 and $\mathbf{v}_{c,n2} = [d(n_2,0), d(n_2,1),\ldots, d(n_2, N-1)]^T$ at column n_2, the orthogonality between them is given as

$$orth\_C(n_1, n_2) = |\mathbf{v}_{c,n1}^T\mathbf{v}_{c,n2}/\mathbf{v}_{c,0}^T\mathbf{v}_{c,0}|. \tag{7}$$

A total of $N(N-1)/2$ different basis row or column vector pairs exist, so the average orthogonality of the Int-DCT matrix $\mathbf{d}$ is

$$O_A(\mathbf{d}) = \frac{1}{N(N-1)/2}\sum_{\forall m_1\neq m_2} orth\_R(m_1, m_2) + \frac{1}{N(N-1)/2}\sum_{\forall n_1\neq n_2} orth\_C(n_1, n_2) \tag{8}$$

Lower value of O_A indicates better orthogonality of the Int-DCT matrix $\mathbf{d}$.

Thirdly, the basis vectors should have almost equal norm to simplify the quantization/de-quantization. For any row vector $\mathbf{v}_{r,m}$, its norm is computed as $\mathbf{v}_{r,m}^T\mathbf{v}_{r,m}$. We scale the norm of any row vector by the norm of the first basis row vector which is $\mathbf{v}_{r,0}^T\mathbf{v}_{r,0}$. Let $NV\_R(\mathbf{v}_{r,m})$ be the Norm Variance of the mth row vector, where

$$NV\_R(\mathbf{v}_{r,m}) = |1 - \mathbf{v}_{r,m}^T\mathbf{v}_{r,m}/\mathbf{v}_{r,0}^T\mathbf{v}_{r,0}|. \tag{9}$$

Similarly, for any column vector $\mathbf{v}_{c,n}$, its norm is computed as $\mathbf{v}_{c,n}^T\mathbf{v}_{c,n}$. With the scaling by the norm of the first column vector which is $\mathbf{v}_{c,0}^T\mathbf{v}_{c,0}$, the Norm Variance $NV\_C(\mathbf{v}_{c,n})$ of any column vector $\mathbf{v}_{c,n}$ can be computed as

$$NV\_C(\mathbf{v}_{c,n}) = |1 - \mathbf{v}_{c,n}^T\mathbf{v}_{c,n}/\mathbf{v}_{c,0}^T\mathbf{v}_{c,0}|. \tag{10}$$

According to (9), if any $\mathbf{v}_{r,m}$ has exactly the same norm as $\mathbf{v}_{r,0}$, $NV\_R(\mathbf{v}_{r,m})$ will be 0 which means that $\mathbf{v}_{r,m}$ does not encounter any norm variance. The same applies for $NV\_C(\mathbf{v}_{c,n})$ in (10). The average Norm Variance of the entire matrix $\mathbf{d}$, which is denoted as $NV_A(\mathbf{d})$, can be computed as

$$NV_A(\mathbf{d}) = \frac{1}{2N}\left[\sum_{m=0}^{N-1} NV\_R(\mathbf{v}_{r,m}) + \sum_{n=0}^{N-1} NV\_C(\mathbf{v}_{c,n})\right]. \tag{11}$$

The lower $NV_A(\mathbf{d})$ indicates that the basis vectors of the Int-DCT matrix $\mathbf{d}$ have similar norm values. These three measures defined in (5), (8) and (11) jointly constitute to the quality of the derived Int-DCT matrix.

C. HARDWARE COST EVALUATION

Besides the above-mentioned evaluations, implementation cost of the Int-DCT matrices is another critical design consideration. Many existing Int-DCT realization such as [34] used conventional multiplier-less multiple constant multiplication (MCM) techniques which involve only adders and hardwired shifts to save hardware cost. In these designs, total full adder cost can be a hardware cost indicator. To achieve lower hardware complexity, we use the reconfigurable multiplier (RM) based method proposed in [35] to implement the multiplications. The architecture of one RM is shown in Fig. 1 (a), where mux stands for multiplexers.

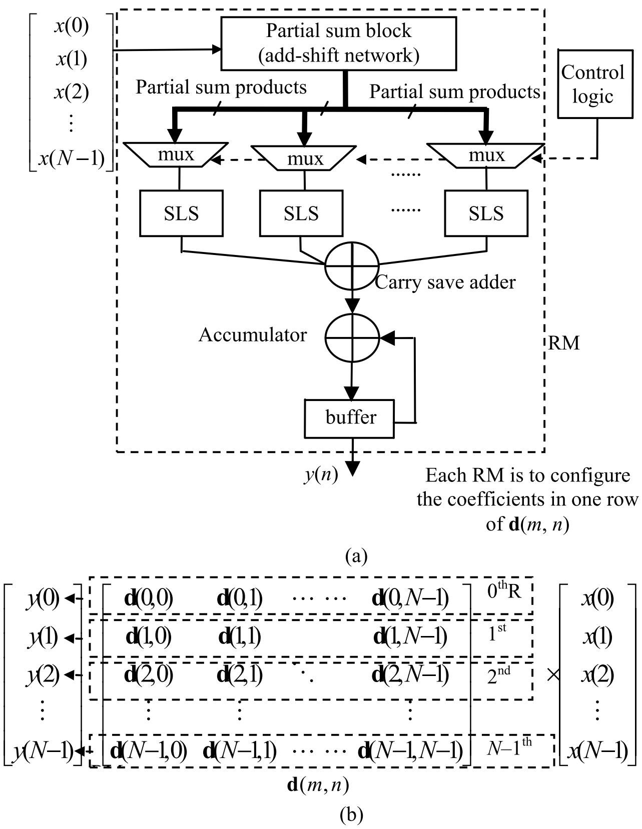

FIGURE 1. (a) RM architecture (b) Multiplications using RMs.

Each RM consists of a partial sum block with add-shift network and a sequence of multiplexers followed by shifters. To further reduce the complexity, the RM design adopted utilizes the newly proposed sporadic logarithmic shifters (SLS) in [36] as shifters. To reduce the complexity of partial sum block and the SLSs, we limit the number of different partial sums and the number of different shift amounts to be generated by the SLSs. Based on the works in [37]–[42], DBNS has been proven to be efficient to implement add-shift digital circuits. Therefore, in this design, we represent the coefficients in the same row of finite precision Int-DCT matrix $\mathbf{d}$ using double base number system (DBNS) [37] as

$$\mathbf{d}(m,n) = \sum_{t=1}^{T} 2^{\alpha_t} 3^{\beta_t} \tag{12}$$

where α_t and β_t are respectively the non-negative exponents of 2 and 3 of the t-th nonzero double base term. T is the total number of nonzero double base terms. The value of α_t is the shift to be performed by SLSs. The values of 3^{β_t} are the partial sums to be implemented inside the partial sum block. It should be noted that the DBNS representation for the same integer is not unique. This provides us the opportunity to search for the efficient DBNS representation for each coefficient in the same row of $\mathbf{d}$ such that the total number of 3^{β_t} and 2^{α_t} are minimized. This can lead to the reduced cost of the partial sum block and the SLSs. After the partial sums are determined, the partial sum block design technique proposed in [42] is adopted. The outputs of the partial sum block are the products by the input and the partial sums. The multiplexers select the correct partial sum product which will be shifted by the SLS. The carry save adder sums up all the T double base terms and the sum is the final product by the input in $x(n)$ and one Int-DCT coefficient in $\mathbf{d}(m,n)$. The buffer and accumulator at the bottom add up all the products for each row to generate one output. Ripple carry adders (RCA) are adopted in the design owing to its lower complexity compared with other adder types [43].

With the RM, the multiplications for 1-dimension DCT can be performed with a total of N RMs, as shown in Fig.1 (b). By selecting the correct inputs for multiplexers and the correct shift amounts for SLSs, each RM is configured to be one of the N coefficients in one row of $\mathbf{d}$. At one instance, the configured RM multiplies with the corresponding input in $x(n)$. For example, the 0^{th} RM is firstly configured to $\mathbf{d}(0,0)$ and multiplies with $x(0)$. In the next multiplication, the 0^{th} RM is configured to $\mathbf{d}(0,1)$ which multiplies with $x(1)$. This reconfiguration and multiplication repeats until all the products are generated.

To evaluate and compare the hardware complexity fairly, we convert the costs of multiplexers [44] and the shifters [36] to full adder count using their approximated area complexity ratios. Knowing that the complexity of a multiplexer is approximately proportional to P which is its number of input lines [35], the area ratio of one w-bit P-to-1 multiplexer to one full adder is: $w \times P \times \rho$. The value of ρ depends on the targeted device technology. Because the SLSs in RMs consist of multiplexers with different numbers of inputs, we can apply the same method to convert the shifter complexity to the equivalent full adder counts. Therefore, in our method, the total approximate full adder count of a given Int-DCT matrix $\mathbf{d}$ is formulated as

$$FA\_\,\text{total}(\mathbf{d}) = \sum_{i=1}^{N_R} w_{RCA\_i} + \rho_m \sum_{i=1}^{N_M} w_{m,i} \times P_i \tag{13}$$

where N_R and N_M are the total numbers of RCAs and multiplexers respectively in the design. w_{RCA_i} and w_{m_i} represent the wordlength of the ith RCA and the ith multiplexer respectively. The multiplexers used in programmable shifters are included in (13). With this area complexity alignment, we can always estimate and compare the implementation cost in terms of *FA_total* for any given Int-DCT matrix when it is implemented using the RM approach.

III. THE PROPOSED ALGORITHM FOR EFFICIENT INT-DCT

A. WEIGHTED SUM APPROACH

As presented in Section II, a group of design criteria to evaluate transform qualities as well as its implementation cost have been discussed. Searching for optimal solutions for these criteria simultaneously is a multi-objective non-linear optimization problem. Unfortunately, it is impossible to have one Int-DCT transform matrix which can simultaneously achieve optimum points for all the above measures. Because each objective can be more critical than others depending on different applications, an efficient optimization approach is desired which can localize the quasi-optimal solutions with flexibilities to adjust the priorities of different objectives.

All the criteria in (5), (8) and (11) are normalized and unit-less measures. Therefore, we can convert the multiple objectives problem into single objective optimization by adopting the weighted sum approach [45] which is defined as

$$J_{MO} = \sum_{i=1}^{z} \lambda_i \frac{J_i}{sf_i}, \quad \text{where} \quad \sum_{i=1}^{z} \lambda_i = 1 \tag{14}$$

and J_i is the ith objective among all the z objectives whose weight is λ_i. sf_i is the scaling factor applied to J_i. J_{MO} is the summed up single objective. In our case, we first consider $z = 3$. C_A, O_A and NV_A are all scaled objectives, so we define the summed single objective η as the quality measure of any Int-DCT matrix $\mathbf{d}$, which can be expressed as

$$\eta(\mathbf{d}) = \lambda_1 \times C_A(\mathbf{d}) + \lambda_2 \times O_A(\mathbf{d}) + \lambda_3 \times NV_A(\mathbf{d}) \tag{15}$$

where λ_1, λ_2 and λ_3 are the weighting factors for C_A, O_A and NV_A respectively. When comparing the quality of two different Int-DCT matrices $\mathbf{a}$ and $\mathbf{b}$, we can say $\eta(\mathbf{a})$ dominates $\eta(\mathbf{b})$ if and only if

$$C_A(\mathbf{a}) \leq C_A(\mathbf{b}), O_A(\mathbf{a}) \leq O_A(\mathbf{b}), \quad \text{and } NV_A(\mathbf{a}) \leq NV_A(\mathbf{b}). \tag{16}$$

One solution $\mathbf{a}$ is called efficient if and only if $\eta(\mathbf{a})$ cannot be dominated by η of any other solution. An Int-DCT matrix $\mathbf{a}$ is said to be optimal if $\eta(\mathbf{a})$ is less or equal to the η of all the remaining candidate solutions [45].

B. THE PROPOSED ALGORITHM

Because the objective is to search for an Int-DCT matrix with minimized η, higher priority for certain individual measure in (15) implies that a higher weighting factor value should be assigned to it. In such case, any reduction in this individual measure can cause η to be decreased effectively. The selection of weighting factors for λ depends on different transform priority. This flexibility for assigning different weighting factors to multiple objectives can make the design adapted to various applications which are with different priorities. For example, if fast video coding/decoding with simplified quantization is desired, NV_A should be with higher weight and hence a bigger λ_3 should be assigned. If the compression efficiency is with priority, C_A and O_A should be given higher weights. Hence, λ_1 and λ_2 should be assigned with higher values than λ_3. After defining the weights based on the specific applications, the next step is to search for the Int-DCT matrix with low η and low hardware cost.

To make the search more effective, it is important to reduce the number of inefficient solutions. This can be achieved by starting with the initial Int-DCT matrix solution denoted as $\mathbf{c}_i(m, n)$, which is from direct scaling of infinite-precision coefficients as $\alpha \cdot C_N^{II}$ followed by coefficient truncation at B-bit. This $\mathbf{c}_i$ is with the lowest C_A, but is not necessarily with good η due to the NV_A and O_A measures. When searching for the solutions to achieve lower η, we apply the constraint that only the least significant bit of the coefficient in $\mathbf{c}_i$ can be changed, i.e.

$$\mathbf{c}_i(m, n) - 1 \leq \mathbf{d}(m, n) \leq \mathbf{c}_i(m, n) + 1. \tag{17}$$

Any other candidate $\mathbf{d}$ whose coefficients are beyond the range in (17) are not assessed, because they have poorer C_A which would affect the compression efficiency. In our experiment, it is observed that the significant C_A increment by the solutions beyond the range specified by (17) cannot be offset by gains in O_A. In addition, solutions further away from $\mathbf{c}_i$ generally cannot help to improve O_A and NV_A. Therefore, if we find a local minimal solution $\mathbf{c}_{local}$ in the above range which can produce the lowest η, we treat $\mathbf{c}_{local}$ as the solution for this stage.

In addition to η which measures the transform quality, the proposed algorithm searches for solution around $\mathbf{c}_{local}$ which is with lower implementation cost measured by full adder count as computed by (13). Unit-less η and *FA_total* are two different measures, so we need to normalize them before applying the weighted sum approach. The scaling factors are chosen to be $\eta(\mathbf{c}_{local})$ and $FA\_total(\mathbf{c}_{local})$ respectively. The overall performance of one Int-DCT matrix $\mathbf{d}$ is then evaluated through another weighted sum, given as

$$p(\mathbf{d}) = \beta_1 \times [\eta(\mathbf{d})/\eta\,(\mathbf{c}_{local})] + \beta_2 \times [FA\_total(\mathbf{d})/FA\_total\,(\mathbf{c}_{local})] \tag{18}$$

where the function $FA\_total(\mathbf{d})$ computes the total FA count using (13). $p(\mathbf{d})$, defined as the p value of the given Int-DCT matrix $\mathbf{d}$, is the overall measure for compression performance and hardware optimality. β_1 and β_2, with $\beta_1 + \beta_2 = 1$, are the weighting factors for the normalized η and the normalized *FA_total* respectively. The selection of weighting factors for β depends on the priority between compression performance and implementation cost. For mobile and integrated devices with extremely limited hardware and power budget, β_2 appears to be more critical and should be assigned with higher value. For applications which require high compression efficiency, β_1 should be higher. After assigning the weighting factors with the initial values by following this rule, we perform the design and evaluate the performance. If any performance specifications are not met, the weighing factors are adjusted until the prioritized design objective is fulfilled. To limit the search space for (18), we apply the constraint that only the least significant bit of the coefficients $\mathbf{c}_{local}(m, n)$ can be varied, as

$$\mathbf{c}_{local}(m, n) - 1 \leq \mathbf{d}(m, n) \leq \mathbf{c}_{local}(m, n) + 1. \tag{19}$$

The overall optimization can therefore be summarized as below and $\mathbf{c}_f$ is the final solution.

$$\mathbf{c}_f = \underset{\mathbf{d}}{\operatorname{argmin}}\{p(\mathbf{d})\}, \text{ s.t. } (17)\&(19). \tag{20}$$

Fig. 2 shows a summary of the pseudo-code for the proposed algorithm. The output of **Int-DCT**(N, B, λ_1, λ_2, λ_3, β_1, β_2) is $\mathbf{c}_f$ which is the final Int-DCT coefficient matrix by the proposed algorithm. The function **original_DCT**(N) generates the N-point Type II infinite precision DCT coefficients using (3). **truncate**(DCT, B) scales the DCT coefficients and truncate the most significant B-bit. The resultant coefficients are converted into integer and stored in $\mathbf{c}_i$. With the given weighting factors, the function **η_compute** computes η value of every candidate matrix $\mathbf{d}$. The matrix $\mathbf{c}_{local}$ is the one with the lowest η within the search space. In the second search loop, the function **FA_compute** computes *FA_total* of the RM based implementation of every candidate matrix $\mathbf{d}$. With *FA_total*, η, weighting factors β_1 and β_2, the function **p_compute** computes the value of $p(\mathbf{d})$ using (18) for the Int-DCT matrix being evaluated. The matrix with the lowest p value is the final solution recorded as $\mathbf{c}_f$.

```
Int-DCT(N, B, λ1, λ2, λ3, β1, β2) {
  DCT = original_DCT(N);
  ci = truncate(DCT, B);
  Initialize d = ci (m, n) – 1; //initialize from lower bound
  Initialize η = ∞;
  Initialize clocal, cf;
  while ci(m, n) – 1 ≤ d(m, n) ≤ci(m, n) + 1 ∀ m & n
        η’ = η_compute(d, λ1, λ2, λ3);
           if η’ < η
             η = η’; clocal = d;
           end;
        increase(d);
  end;
  Initialize p = ∞;
  Initialize d = clocal(m, n) – 1; //initialize from lower bound
  while clocal(m, n) – 1 ≤d(m, n) ≤clocal(m, n) + 1 ∀ m & n
        FA_total = FA_compute(d);
        η’ = η_compute(d, λ1, λ2, λ3)
        p’ = p_compute(FA_total, η’, β1, β2);
           if p’ < p
             p = p’; cf = d;
           end;
        increase(d);
  end;
  return cf;
}
```

FIGURE 2. Pseudo code for the proposed algorithm.

IV. RESULTS AND DISCUSSIONS

A. DESIGN EXAMPLE OF INT-DCT WITH UNIFORM PRIORITY

In the first part, we demonstrate the design flow on the 16-point Int-DCT using the proposed algorithm. In this example, we set coefficient wordlength $B = 8$ and assume that C_A, O_A and NV_A have the uniform priority for optimization, i.e. $\lambda_1 = \lambda_2 = \lambda_3 = 0.333$. Next, a search space is created and the algorithm searches for the solution which can generate lower η than $\mathbf{c}_i$, as shown in Fig. 3. The axes in the 3D plot are C_A, O_A and NV_A respectively. The perfect point is the origin which is achievable by original infinite precision DCT coefficients in $\mathbf{C}_N^{II}$. After scaling and truncation, we obtain $\mathbf{c}_i$ and the search algorithm allocates $\mathbf{c}_{local}$. From Fig. 3, we can clearly see that $\mathbf{c}_{local}$ represented by red dot is closer to the origin compared with $\mathbf{c}_i$ represented by green dot.

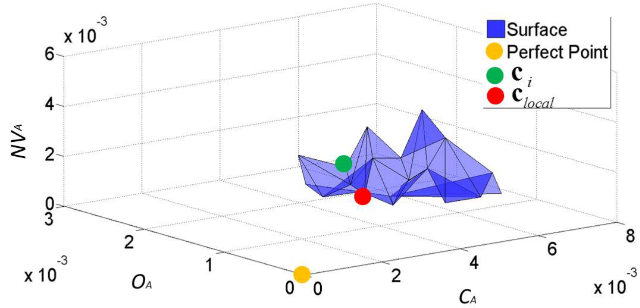

FIGURE 3. Proposed design algorithm search space to minimize η.

The next stage is to search for $\mathbf{c}_f$. To achieve lower hardware cost, we assign higher weight for β_2 compared to β_1. In this design example, we assign $\beta_2 = 0.8$, so $\beta_1 = 0.2$. A search space is created around $\mathbf{c}_{local}$ and $p(\mathbf{d})$ of different solutions are evaluated using (18). $\mathbf{c}_f$ is selected which has the lowest p value. Through the process of searching from initial solution $\mathbf{c}_i$ to the final solution $\mathbf{c}_f$, the C_A, O_A, NV_A, η, *FA_total* and p values are shown in Table 1 below.

TABLE 1. η and ***FA_total*** of the evolving Int-DCT from $\mathbf{c}_i$ to $\mathbf{c}_f$.

	$\mathbf{c}_i$	$\mathbf{c}_{local}$	$\mathbf{c}_f$
C_A	0.0035985	0.0038652	0.0048417
O_A	0.0012441	0.0011353	0.0032359
NV_A	0.0022125	0.0011597	0.0038300
η	0.0023517	0.0020534	0.0039692
FA_total	1456	1660	1146
p	-	1	0.94

From Table 1, we can see that C_A increases when $\mathbf{c}_i$ moves to $\mathbf{c}_{local}$. However, $\mathbf{c}_{local}$ achieves better η, contributed by its lower O_A and NV_A. From the results, we can see that although $\mathbf{c}_f$ has higher η compared with $\mathbf{c}_{local}$, but it reduces the *FA_total* by 31% over $\mathbf{c}_{local}$. For the scenario when we have higher β_2, such $\mathbf{c}_f$ is with the lowest p value and hence is considered as the final solution. This verifies that the proposed algorithm can generate the efficient Int-DCT solution which provides good trade-off between coding performance and implementation cost.

In addition to 16-point, we also design for the 32-point Int-DCT using the proposed algorithm and compare the performance between $\mathbf{c}_f$ with some other competing transforms proposed in the recent years. Similar as the proposed transforms, these competing methods are with limited approximations to the DCT coefficients, so the performance of their 16-point and 32-point transforms and corresponding hardware costs are comparable. Besides RICT [8] and CT [14], EDCT is the hardware efficient DCT proposed by [34]. In addition, we compare the results with the most recently published truncation scheme based DCT (TSDCT) proposed in [30]. To evaluate η, the same coefficient wordlength B is assumed. Because EDCT is a hardware efficient implementation using the Int-DCT coefficients from [14], both EDCT and CT have the same η results but different implementations. When evaluating the *FA_total*, a word length of 8 bits is assumed for the input. We have implemented all the designs into Verilog and synthesized on Xilinx Spartan VI FPGA XC6SLX45 by Xilinx ISE WebPACK with the supply power at 1.2V. For this device, our experiment on multiplexer and full adder shows that the ratio $\rho_m \approx 0.15$ is adopted to evaluate *FA_total*. To compare both Int-DCT matrix quality and the hardware cost, η, areas in #of LUT slices, delays in ns and powers in mW after place and route are presented in Table 2. The rows "Imp." present the percentage improvement by the proposed designs for each measure.

TABLE 2. η **and hardware performance comparisons on FPGA ($\lambda_1 = \lambda_2 = \lambda_3 = 0.333$) for 16-point and 32-point DCT.**

		CT [14]	EDCT [34]	RICT [8]	TSDCT [30]	Proposed
16-point	η	0.0544	0.0544	0.0042	0.0543	0.0040
	Imp.	92.6%	92.6%	4.8%	92.6%	
	area	2958	2562	2275	1535	1382
	Imp.	53.3%	46.1%	39.3%	10.0%	
	delay	11.45	15.96	12.93	13.13	7.92
	Imp.	30.8%	50.4%	38.7%	39.7%	
	power	408	133	139	136	126
	Imp.	69.1%	5.3%	9.4%	7.4%	
32-point	η	0.0023	0.0023	0.0133	0.0024	0.0021
	Imp.	17.4%	17.4%	85.7%	12.5%	
	area	12944	6066	6199	2741	2404
	Imp.	81.4%	60.4%	61.2%	12.3%	
	delay	12.43	17.25	11.23	12.5	9.18
	Imp.	26.1%	46.8%	18.3%	26.6%	
	power	599	241	220	165	161
	Imp.	73.1%	33.2%	26.8%	2.42%	

We take the average of the percentage improvements for the 16-point and the 32-point designs in Table 2 to evaluate the performances of every method. The results show that the proposed algorithm achieves the lowest η compared with other methods. The effort to keep η low by the proposed multi-objective optimization preserves the core properties of the derived Int-DCT. The root mean square errors due to this Int-DCT coefficient approximation are 0.00468 and 0.00522 for 16-point and 32-point respectively. In terms of hardware performance, the proposed algorithm designs the Int-DCT with lower areas by 67.4%, 53.2%, 50.2% and 11.1% over CT, EDCT, RICT and TSDCT respectively. The circuit delays by the proposed method are 28.5%, 48.6%, 28.5% and 33.1% shorter than CT, EDCT, RICT and TSDCT respectively. For total power consumption, the proposed designs reduced the power cost by 71.1%, 19.2%, 18.1% and 4.89% respectively from the designs by CT, EDCT, RICT and TSDCT respectively. The lower implementation cost is achieved by the proposed optimization on $p(\mathbf{d})$ which is a performance measure considering both compression efficiency and the implementation cost, as presented in (13). Through this effort, the proposed algorithm can always search around $\mathbf{c}_{local}$ and find better solutions which are with very similar η but a much lower full adder count. In addition, the adoption of the recently proposed SLSs [36] into the RMs in our designs is another reason of the lower hardware cost achieved.

TABLE 3. η **and hardware cost comparisons by MDA and the proposed transform for 16-point DCT.**

	MDA [22]	Proposed
η	0.1047	0.0040
area	927	1382
delay	4.75	7.92
power	140	126

Another 16-point DCT architecture for named as MDA (Multiplication-free Digital Architecture) was proposed in [22]. It is relevant to the proposed transforms and can be compared. However, unlike other competing methods [8], [14], [34], and [30], this transform by [22] is derived with much higher degree of approximation, so the coefficients are very different from the original DCT. Therefore, besides the scaling matrix, the transform matrix consists of 1, 0 and -1 only, which leads to the minimum hardware cost. However, this hardware minimization is at the expense of coding performance. The approximation quality η, hardware cost and power cost by MDA is listed in Table 3. It can be observed that the transform matrix with 1, 0 and -1 only can reduce the hardware cost and delay over the proposed transform by around 40%. However, the proposed transform causes slightly less power. More importantly, the heavy approximation by MDA makes η to increase significantly to be around 26 times of the one by the proposed transform. This comparison shows that MDA can achieve less hardware cost only if approximation error and coding performance can be heavily compromised. However, heavy approximation is unaffordable in most applications. This is the reason why all other competing methods [8], [14], [30], [34], proposed transforms which also achieve much lower η than MDA. In this scenario, the proposed transform is more applicable because of the much lower η. Meanwhile, the hardware cost

TABLE 4. Hardware performance comparisons on ASIC for 16-point and 32-point DCT.

		CT [14]	EDCT [34]	RICT [8]	TSDCT [30]	Proposed
16-point	area	24663	13430	27417	39109	9835
	Imp.	60.1%	26.8%	64.1%	74.9%	
	delay	7.39	6.13	6.16	6.04	6.69
	Imp.	9.5%	–9.1%	–8.6%	–10.8%	
	power	6.47	2.06	4.86	6.06	2.20
	Imp.	66.0%	–6.8%	54.7%	63.7%	
32-point	area	70970	59501	86692	78400	22196
	Imp.	68.7%	62.7%	74.4%	71.7%	
	delay	7.55	6.22	6.22	6.04	6.70
	Imp.	11.3%	–7.7%	–7.7%	–10.9%	
	power	20.59	5.45	15.7	12.04	4.91
	Imp.	76.2%	9.9%	68.7%	59.2%	

overhead by the proposed transform over MDA is limited below 50%.

All the proposed designs and the competing designs are also mapped to 45*nm* standard cell library and run by Synopsys Design-Compiler TM. Synopsys Power-Complier TM with version: J-2014.09-SP3 is used to perform the power analysis. A supply voltage of 1.0V is used. Tool optimization is set to timing constraint. The results of areas in μm^2, delays in *ns* and total power in *m*W are presented in Table 4. From the results in ASIC, it is evident that the proposed Int-DCT reduces the silicon area cost by at least 26.8% and 62.7% for 16-point and 32-point respectively, compared with other relevant designs. Due to the multipliers reconfiguration time, the proposed Int-DCT encounters slightly longer delay than some existing designs, such as EDCT and TSDCT. However, this overhead is limited within 10.9%. In terms of total power, the proposed Int-DCT can reduce the power cost over other competing methods in most of the comparisons, except the case with EDCT for 16-point. The reason is the shorter critical path delay by 16-point EDCT, and this helps to save switching activities. For 32-point, however, the proposed Int-DCT reduces power by 9.9% over EDCT. In general, although the proposed Int-DCT sometimes causes longer delay, it achieves more significant area and power reduction in ASIC over other competing methods.

To verify that the reduced hardware cost by the proposed transform is achieved without compromising the compression performance, the proposed Int-DCT matrix is implemented into the HEVC reference software HM16.14 and the coding performance is measured. The same benchmark video sequences with different resolutions are compressed and tested under the common test conditions [46]. We verify the performance using the standard BD-rates which are shown in Table 5 below. YUV color space is used to evaluate where Y represents luminance, U and V represent chrominance. For YUV, the ratio of significance for Y, U, V is 4:1:1. The BD-rate number is in terms of the percentage bitrate difference for the same peak signal to noise ratio. A positive BD-rate in Table 4 indicates coding loss compared to the

TABLE 5. Coding performance measured by BD-rate (%) compared with HM16.14 under the all-intra configuration ($\lambda_1 = \lambda_2 = \lambda_3 = 0.333$).

Resolution	Sequence	Y	U	V	YUV
4K	NebutaFestival	0.11120	–0.10750	–0.20380	–0.06430
	PeopleOnStreet	0.04714	0.01327	–0.07826	0.07896
	SteamTrain	0.07650	–0.33280	0.04530	–0.00520
	Traffic	–0.02320	–0.09296	0.04430	–0.01839
1080p	BasketballDrive	0.00400	–0.02370	–0.02420	0.00880
	BQTerrace	–0.03240	0.01670	0.00910	0.01750
	Cactus	0.03140	0.01380	0.00130	0.01150
	Kimono1	0.06630	–0.07030	–0.10380	0.03130
	ParkScene	0.02240	–0.00520	0.03870	0.02130
WVGA	BasketballDrill	0.01050	0.04050	–0.04240	0.00760
	BQMall	0.01100	–0.00520	0.04950	–0.01320
	PartyScene	–0.00020	–0.01760	–0.01730	–0.00420
	RaceHorses	0.01540	–0.03380	0.10090	0.01160
WQVGA	BasketballPass	0.01150	0.00930	–0.05920	0.00940
	BlowingBubbles	0.00540	0.11490	0.12750	0.00660
	BQSquare	–0.00400	0.02110	–0.04860	–0.00390
	RaceHorses	0.00100	–0.05310	0.08780	0.00670
720p	FourPeople	0.05310	0.11280	0.10860	0.06260
	Johnny	0.01640	–0.02070	–0.00760	0.01290
	KristenAndSara	0.02380	0.04500	0.00140	0.02490
Average		0.02236	–0.01877	0.00146	0.01012

TABLE 6. Coding performance of MDA [22] measured by BD-rate (%) compared with HM16.14 under the all-intra configuration.

Resolution	Sequence	Y	U	V	YUV
WQVGA	BasketballPass	1.5550	3.4183	2.7558	1.7397
	BlowingBubbles	0.4892	1.6071	1.7670	0.6045
	BQSquare	0.4982	0.7681	0.7386	0.5157
	RaceHorse	1.2882	3.8051	3.4257	1.6357

anchor, and a negative BD-rate relates to coding gain. The BD-rate results show that the average difference between the proposed Int-DCT and the original transforms in HEVC is less than 0.03% for different resolutions.

From Table 3, we conclude that the proposed transforms have much lower η than MDA proposed by [22]. To verify the better coding performance over [22], we implement the transforms by [22] into the same reference software and code four WQVGA video sequences. The BD-rate results are shown in Table 6, where positive BD-rate indicates coding loss and a negative BD-rate relates to coding gain. It can be seen that the approximated Int-DCT by MDA always has obvious coding loss from 0.5% up to 3.8%. On the other hand, from the BD-rates of WQVGA videos as shown in Table 5, the proposed transform can achieve some coding gains compared with the reference and the coding loss is limited at around 0.001% to 0.128% only.

From these results from Table 2 to 6, it can be shown that the proposed transforms have similar performance as the transforms in HEVC with negligible difference. Meanwhile, the proposed transforms are with the reduced hardware cost and power consumption over existing designs without compromising the coding performance.

B. DESIGNS WITH NON-UNIFORM PRIORITIES

In Section IV.A, Int-DCT solutions with the uniform optimization priority are presented. In some scenario, one particular property can have higher priority than others. For example, modern communication technologies demand higher compression efficiency for faster image and video transmission speed in a given channel bandwidth [47]. Electrocardiogram (ECG) signal processing also requires higher compression with little information loss [48]. ECG signal is decomposed by means of a linear orthogonal transformation before the transform coefficients are appropriately encoded. In these applications, we need to set higher priority for orthogonality than other properties. Based on the proposed weighted sum approach in (15), the higher λ_2 value helps to generate solution with the higher orthogonality, when other properties are slightly compromised. To verify the performance, we select $\lambda_2 = 0.8$ and $\lambda_1 = \lambda_3 = 0.1$ in this section. We re-run the proposed algorithm to generate the Int-DCT matrix $\mathbf{c}_f$. To achieve lower hardware cost, we still assign $\beta_2 = 0.8$ and $\beta_1 = 0.2$. The C_A, O_A, NV_A, η, FA_total and p values of $\mathbf{c}_i$, $\mathbf{c}_{local}$ and $\mathbf{c}_f$ by the proposed algorithm are shown in Table 7.

TABLE 7. η and *FA_total* of the evolving Int-DCT coefficients from $\mathbf{c}_i$ to $\mathbf{c}_f$ ($\lambda_1 = \lambda_3 = 0.1$ and $\lambda_2 = 0.8$).

	$\mathbf{c}_i$	$\mathbf{c}_{local}$	$\mathbf{c}_f$
C_A	0.0035985	0.0049371	0.0049371
O_A	0.0012441	0.0006551	0.0006551
NV_A	0.0022125	0.0009766	0.0009766
η	0.0023517	0.0011155	0.0011155
FA_total	1456	1748	1748
p	-	1	1

From Table 7, we can see that C_A increases when $\mathbf{c}_i$ moves to $\mathbf{c}_{local}$ to achieve much lower η. Unlike the previous results, $\mathbf{c}_f$ for this new set of weighting factor turns out to be the same as $\mathbf{c}_{local}$. The reason is $\mathbf{c}_{local}$ in this experiment can produce very low η. Any effort to change $\mathbf{c}_{local}$ for lower FA_total can cause $\eta$to increase significantly. Although we have set $\beta_2 = 0.8$, the result of the algorithm still shows that $\mathbf{c}_{local}$ is the solution which can produce the lowest p. This implies that it is worth to sacrifice η for the little reduction in FA_total and hence we should take $\mathbf{c}_{local}$ as the final solution $\mathbf{c}_f$. In spite of this, the hardware implementation cost on the same FPGA for this new Int-DCT matrix is still lower than the competing methods, as shown in Table 8. The areas are in #of LUT slices and the delays are in ns. The powers are with unit of mW. The hardware cost of the proposed solution is at least 15.50% lower than other Int-DCT architectures. Delay and power of the proposed architecture are also lower than the state-of-the-art designs by at least 3.5% and 3.7% respectively. The root mean square errors of the proposed Int-DCT approximation with non-uniform weighting factors is 0.00621.

We also implement $\mathbf{c}_f$ into the same HEVC reference software. BD-rates are shown in Table 9. By comparing with the performance by the Int-DCT matrix generated with $\lambda_1 = \lambda_2 = \lambda_3 = 0.333$ in Table 5, we can see the average difference between the proposed Int-DCT with the new weighting factors and the original transforms in HEVC becomes smaller which is less than 0.02%. This successfully verifies that the proposed algorithm is capable to generate effective Int-DCT solution with the prioritized property.

TABLE 8. η and hardware cost comparisons in FPGA ($\lambda_1 = \lambda_3 = 0.1$ and $\lambda_2 = 0.8$).

	CT [14]	EDCT [34]	RICT [8]	TSDCT [30]	Proposed
η	0.0544	0.0544	0.0042	0.059	0.0011
Imp.	98.00%	98.00%	73.81%	98.10%	
area	2958	2562	2275	1535	1297
Imp.	56.15%	49.38%	42.99%	15.50%	
delay	11.45	15.96	12.93	13.13	11.05
Imp.	3.49%	30.76%	14.54%	15.84%	
power	408	133	139	136	131
Imp.	67.89%	1.50%	5.76%	3.68%	

TABLE 9. Coding performance measured by BD-rate (%) compared with HM16.14 under the all-intra configuration ($\lambda_1 = \lambda_3 = 0.1$, $\lambda_2 = 0.8$).

Resolution	Sequence	Y	U	V	YUV
4K	NebutaFestival	–0.01878	–0.01118	–0.04593	–0.13985
	PeopleOnStreet	0.05564	0.44068	–0.06235	–0.01407
	SteamTrain	0.11108	–0.25930	0.17281	0.10685
	Traffic	–0.00508	–0.01635	–0.02041	–0.01426
1080p	BasketballDrive	0.08745	0.01471	0.05063	0.03174
	BQTerrace	–0.00453	–0.02717	–0.02702	–0.00512
	Cactus	–0.00238	–0.00714	–0.02702	–0.00525
	Kimono1	0.04935	0.02004	–0.01474	0.03731
	ParkScene	0.00396	0.00978	–0.00772	0.00469
WVGA	BasketballDrill	0.02157	0.01029	–0.05259	0.01444
	BQMall	–0.01440	–0.02654	–0.01413	–0.01757
	PartyScene	0.00622	–0.02304	–0.01332	0.00340
	RaceHorses	0.02178	0.01033	0.04772	0.01805
WQVGA	BasketballPass	–0.02240	–0.06211	–0.02699	–0.02883
	BlowingBubbles	–0.00927	0.03456	–0.02055	–0.00717
	BQSquare	0.00269	–0.08673	–0.06396	0.00093
	RaceHorses	–0.01746	0.00633	0.15921	–0.00443
720p	FourPeople	–0.00720	–0.04840	–0.02100	–0.01037
	Johnny	0.00968	0.03171	0.02146	0.01181
	KristenAndSara	0.00578	–0.02461	–0.04529	0.00103
Average		0.01368	–0.00071	–0.00056	–0.00083

C. POWER COST EVALUATION WITH VIDEO SAMPLE

In Section IV.A and B, the solutions with the uniform and non-uniform Int-DCT priorities are evaluated on FPGA and ASIC. The power cost is estimated by assuming random input samples to the DCT circuits. In this section, we evaluate the power cost (mW) when the proposed 16-point Int-DCT and other competing transforms are used to compress the video sequence RaceHorses.yuv. The compression speed is set at 500 frames per second. The experiment is carried on Xilinx Spartan6, xc6slx45 device with clock frequency at 50MHz and supply voltage at 1.2V. Both Int-DCT with uniform and non-uniform priorities by the proposed method are evaluated. From Table 2, the results by EDCT, RICT and TSDCT are more competitive than CT, so we compare our results with EDCT, RICT and TSCDT solutions in this evaluation. The results are presented in Table 10.

TABLE 10. Power costs comparison to compress sample video at maximum frequency and 50MHz for uniform and non-uniform priorities.

	EDCT [34]	RICT [8]	TSDCT [30]	Proposed λ_1=0.333 λ_2=0.333 λ_3=0.333	Proposed λ_1=0.1 λ_3 = 0.1 λ_2 = 0.8
F_{max}(MHz)	75.2	77.2	73.2	92.3	92.3
Power at F_{max} (mW)	136	166	136	121	129
Power at 50MHz (mW)	130	160	112	104	112

Both our transforms for uniform priority and non-uniform priority encounter lower power cost compared with competing methods when operating at maximum frequency and 50MHz for this video sample. On average, the power reductions contributed by the proposed transforms are 15.5%, 31.1% and 9.1% over EDCT, RICT and TSDCT respectively when compressing at maximum frequency. When running at 50MHz, the reductions by the proposed transform are 9.5%, 26.1% and 2.6% respectively over these competing methods.

V. CONCLUSION

A new algorithm to generate efficient Int-DCT is proposed in this paper. The efficient coding performance of the proposed Int-DCT is achieved by increasing the closeness to original DCT and the orthogonality of the basis vectors using a weighted sum approach. In addition, implementation cost is addressed in the proposed algorithm, so the generated Int-DCT matrices are with good trade-off between compression efficiency and hardware cost. The proposed algorithm can be applied flexibly to generate Int-DCT for different compression or hardware constraints by adjusting the weighting factors. The experimental results show that the proposed algorithm can generate Int-DCT with almost the same coding performance as the HM16.14 in HEVC. Meanwhile, the hardware cost is reduced compared with recent state-of-the-art implementations which can produce similar coding performance.

REFERENCES

[1] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

[2] C. J. Tablada, F. M. Bayer, and R. J. Cintra, "A class of DCT approximations based on the Feig–Winograd algorithm," *Signal Process.*, vol. 113, no. 8, pp. 38–51, Aug. 2015.

[3] V. A. Coutinho, R. J. Cintra, and F. M. Bayer, "Low-complexity multidimensional DCT approximations for high-order tensor data decorrelation," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2296–2310, May 2017.

[4] S. Yaldiz, A. Demir, and S. Tasiran, "Stochastic modeling and optimization for energy management in multicore systems: A video decoding case study," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1264–1277, Jul. 2008.

[5] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 2, pp. 238–251, Feb. 2015.

[6] P. A. M. Oliveira, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Madanayake, "Low-complexity image and video coding based on an approximate discrete Tchebichef transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1066–1076, May 2017.

[7] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *IEE Proc. I-Commun., Speech Vis.*, vol. 136, no. 4, pp. 276–282, Aug. 1989.

[8] C.-K. Fong, Q. Han, and W.-K. Cham, "Recursive integer cosine transform for HEVC and future video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 326–336, Feb. 2017.

[9] *Advanced Video Coding for Generic Audiovisual Services*, document ITU-T Rec. H.264, 2009.

[10] M. U. K. Khan, M. Shafique, L. Bauer, and J. Henkel, "Multicast FullHD H.264 intra video encoder architecture," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 12, pp. 2049–2053, Dec. 2015.

[11] *Advanced Coding of Audio and Video-Part 2: Video*, document GB/T 20090.2-2006 AVS, May 2006.

[12] J. Loomis and M. Wasson, *VC-1 Technical Overview*. Redmond, WA, USA: Microsoft Corporation, Oct. 2007.

[13] G. J. Sullivan, J.-R. Ohm, W.-J. 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.

[14] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, and M. Sadafale, "Core transform design in the High Efficiency Video Coding (HEVC) standard," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1029–1041, Dec. 2013.

[15] C. Lan, J. Xu, W. Zeng, G. Shi, and F. Wu, "Variable block-sized signal-dependent transform for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1920–1933, Aug. 2018.

[16] F. M. Bayer, R. J. Cintra, A. Madanayake, and U. S. Potluri, "Multiplierless approximate 4-point DCT VLSI architectures for transform block coding," *Electron. Lett.*, vol. 49, no. 24, pp. 1532–1534, Nov. 2013.

[17] 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, Jun. 2014.

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

[19] 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, Jul. 2012.

[20] 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, Reg. Papers*, vol. 61, no. 6, pp. 1727–1740, Jun. 2014.

[21] A. Edirisuriya, A. Madanayake, R. J. Cintra, V. S. 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.

[22] 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 Conv. Elect. Electron. Eng. Israel*, Eilat, Israel, Nov. 2012, pp. 1–5.

[23] C.-K. Fong and W.-K. Cham, "LLM integer cosine transform and its fast algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 844–854, Jun. 2012.

[24] E. Alshina, A. Alshin, W. Lee, J. H. Park, K. Pachauri, and P. Topiwala, *CE10: Full Factorization Core Transforms for HEVC*, document G579, JCT-VC Meeting, Nov. 2011.

[25] V. Sze, M. Budagavi, and G. J. Sullivan, *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. New York, NY, USA: Springer, 2014.

[26] R. Joshi, J. Sole, and K. Karczewicz, *CE10: Scaled Integer Transforms Supporting Recursive Factorization Structure*, document G579, JCT-VC Meeting, Nov. 2011.

[27] M. Jridi, A. Alfalou, and P. K. Meher, "A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 2, pp. 449–457, Feb. 2015.

[28] M. Jridi and P. K. Meher, "Scalable approximate DCT architectures for efficient HEVC-compliant video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1815–1825, Aug. 2017.

[29] M. Masera, M. Martina, and G. Masera, "Adaptive approximated DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2714–2725, Dec. 2017.

[30] H. Sun, Z. Cheng, A. M. Gharehbaghi, S. Kimura, and M. Fujita, "Approximate DCT design for video encoding based on novel truncation scheme," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1517–1530, Apr. 2018.

[31] J. Chen, Y. Wang, J. Zhao, and S. Rahardja, "A new area and power efficient DCT circuits using sporadic logarithmic shifters," *IEICE Electron. Express*, vol. 16, no. 14, pp. 20190317–20190323, 2019.

[32] G. Bjøntegaard, "VCEG-M33: Calculation of average PSNR differences between RD curves," ITU-T Video Coding Experts Group, Geneva, Switzerland, Tech. Rep. VCEG-M33, 2001.

[33] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, no. 5, pp. 640–643, May 1991.

[34] P. K. Meher, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo, "Efficient integer DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan. 2014.

[35] J. Chen and C. H. Chang, "High-level synthesis algorithm for the design of reconfigurable constant multiplier," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 12, pp. 1844–1856, Dec. 2009.

[36] J. Chen, C.-H. Chang, Y. Wang, J. Zhao, and S. Rahardja, "New hardware and power efficient sporadic logarithmic shifters for DSP applications," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 4, pp. 896–900, Apr. 2018.

[37] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, "Theory and applications of the double-base number system," *IEEE Trans. Comput.*, vol. 48, no. 10, pp. 1098–1106, Oct. 1999.

[38] R. Muscedere, V. Dimitrov, G. A. Jullien, and W. C. Miller, "Efficient techniques for binary-to-multidigit multidimensional logarithmic number system conversion using range-addressable look-up tables," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 257–271, Mar. 2005.

[39] V. Dimitrov, G. Jullien, and R. Muscedere, *Multiple-Base Number System: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[40] V. S. Dimitrov, K. U. Jarvinen, and J. Adikari, "Area-efficient multipliers based on multiple-radix representations," *IEEE Trans. Comput.*, vol. 60, no. 2, pp. 189–201, Feb. 2011.

[41] M. Azarmehr, M. Ahmadi, G. A. Jullien, and R. Muscedere, "High-speed and low-power reconfigurable architectures of 2-digit two-dimensional logarithmic number system-based recursive multipliers," *IET Circuits, Devices Syst.*, vol. 4, no. 5, pp. 374–381, Sep. 2010.

[42] J. Chen, C. H. Chang, F. Feng, W. Ding, and J. Ding, "Novel design algorithm for low complexity programmable FIR filters based on extended double base number system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 1, pp. 224–233, Jan. 2015.

[43] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 689–702, Oct. 1996.

[44] P. Tummeltshammer, J. C. Hoe, and M. Püschel, "Time-multiplexed multiple-constant multiplication," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1551–1563, Sep. 2007.

[45] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: A new method for Pareto front generation," *Struct. Multidisciplinary Optim.*, vol. 31, no. 2, pp. 105–116, Feb. 2006.

[46] *Common Test Conditions and Software Reference Configurations*, document G579, JCT-VC Meeting, 2012.

[47] R. Rani, "Performance analysis of different orthogonal transform for image processing application," *Int. J. Appl. Res.*, vol. 1, no. 12, pp. 844–847, Oct. 2015.

[48] C. K. Jha and M. H. Kolekar, "Electrocardiogram data compression using DCT based discrete orthogonal Stockwell transform," *Biomed. Signal Process. Control*, vol. 46, no. 9, pp. 174–181, Sep. 2018.

JIAJIA CHEN received the B.Eng. degree (Hons.) and Ph.D. degree from Nanyang Technological University, Singapore, in 2004 and 2010, respectively. From April 2012 to March 2018, he was the Faculty Member with the Singapore University of Technology and Design, Singapore. Since April 2018, he has been with the Nanjing University of Aeronautics and Astronautics, China, where he is currently a Professor. His research interests include computational transformations of low-complexity digital filters and low power VLSI designs for digital signal processing applications. Dr. Chen served as a Web Chair of Asia-Pacific Computer Systems Architecture Conference, in 2005, a Technical Program Committee member of European Signal Processing Conference, in 2014, and an Associate Editor of *EURASIP Journal on Embedded Systems* (Springer), since 2016.

SHUMIN LIU received the B.Eng. degree (Hons.) from the Pillar of Engineering Product Development, Singapore University of Technology and Design, Singapore, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include image processing, multifocus image fusion, and homography estimation for sports video analysis technologies.

GELEI DENG is currently pursuing the B.Eng. degree in electrical engineering with the Singapore University of Technology and Design. He is also a Research Assistant with the SUTD-MIT International Design Center. His main research interest includes the digital filter design and optimizations.

SUSANTO RAHARDJA (F'11) received the B.Eng. degree in electronic engineering from the National University of Singapore, in 1991, and the M.Eng. and Ph.D. degrees in electronic engineering from Nanyang Technological University, Singapore, in 1993 and 1997, respectively. He is currently a Chair Professor with the Northwestern Polytechnical University (NPU) under the Thousand Talent Plan of People's Republic of China. His research interests include in multimedia, signal processing, wireless communications, discrete transforms and signal processing algorithms, and implementation and optimization. Dr. Rahardja was the recipients of numerous awards, including the IEE Hartree Premium Award, the Tan Kah Kee Young Inventors' Open Category Gold Award, the Singapore National Technology Award, A*STAR Most Inspiring Mentor Award, Finalist of the 2010 World Technology & Summit Award, the Nokia Foundation Visiting Professor Award and the ACM Recognition of Service Award.