Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 736460, 17 pages
doi:10.1155/2008/736460

Research Article

A Generalized Approach to Linear Transform Approximations with Applications to the Discrete Cosine Transform

Yinpeng Chen and Hari Sundaram

The Katherine K. Herberger College of the Arts and the Ira A. Fulton School of Engineering, Arts, Media and Engineering Program, Arizona State University, Tempe, AZ 85281, USA

Correspondence should be addressed to Hari Sundaram, hari.sundaram@asu.edu

Received 13 June 2007; Revised 1 February 2008; Accepted 17 March 2008

Recommended by Lisimachos P. Kondi

This paper aims to develop a generalized framework to systematically trade off computational complexity with output distortion in linear transforms such as the DCT, in an optimal manner. The problem is important in real-time systems where the computational resources available are time-dependent. Our approach is generic and applies to any linear transform and we use the DCT as a specific example. There are three key ideas: (a) a joint transform pruning and Haar basis projection-based approximation technique. The idea is to save computations by factoring the DCT transform into signal-independent and signal-dependent parts. The signal-dependent calculation is done in real-time and combined with the stored signal-independent part, saving calculations. (b) We propose the idea of the complexity-distortion framework and present an algorithm to efficiently estimate the complexity distortion function and search for optimal transform approximation using several approximation candidate sets. We also propose a measure to select the optimal approximation candidate set, and (c) an adaptive approximation framework in which the operating points on the C-D curve are embedded in the metadata. We also present a framework to perform adaptive approximation in real time for changing computational resources by using the embedded metadata. Our results validate our theoretical approach by showing that we can reduce transform computational complexity significantly while minimizing distortion.

1. INTRODUCTION

This paper presents a novel framework for developing linear transform approximations that adapt to changing computational resources. The problem is important since in real-time multimedia systems, the computational resources available to content analysis algorithms are not fixed and can also vary with time (Figure 1). A *generic* computationally scalable framework for content analysis would be very useful. The problem is made difficult by the observation that the relationship between computational resources and distortion depends on the specific content. The desired approximation framework should provide a set of approximations that significantly decreases the computational complexity while introducing small errors. Such framework would be very useful for low-power hand-held devices or wireless sensor devices since power consumption is affected by the number of CPU cycles. Hence decreasing computational complexity (CPU cycles) while minimally affecting distortion would be a useful strategy to conserve power.

1.1. Related work

There has been prior work on fast computation for exact transform. Fast, recursive DCT algorithm based on the sparse factorizations of the DCT matrix is proposed in [1–3]. Besides, 1D algorithms, two-dimensional DCT algorithms have also been investigated in [4–7]. The theoretical lower bound on the number of multiplications required for the eight point 1-D DCT has been proven to be 11 [8, 9] and the Loeffler's method [10] with 11 multiplications and 29 additions is the most efficient solution. The energy tradeoffs for DSP-based implementation of IntDCT was proposed in [11].

There has been prior work on hardware-adaptive optimal implementation of linear digital signal processing (DSP)

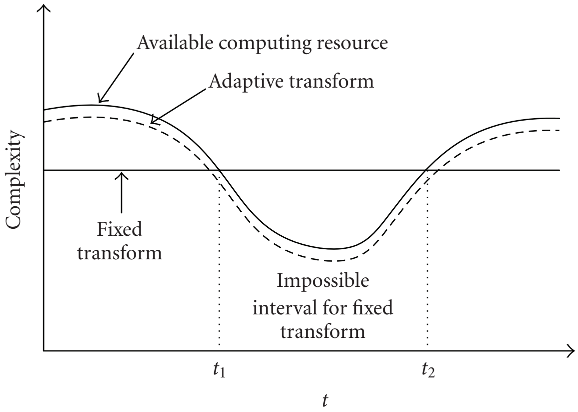

Available computing resource
Adaptive transform
Complexity
Fixed transform
Impossible interval for fixed transform
t_1
t_2
t

Figure 1: Computational complexity for fixed and adaptive transforms (e.g., video decoding algorithm that adapts to changing computational resources). During the time between t_1 and t_2, the available resources for video player are less than the computational complexity needed for video decoding and rendering. This can either result in a delay or a frame drop.

transforms. SPIRAL [12] automatically generates high-performance code that is tuned to the given platform for a specified transform. ATLAS [13, 14] is a well-known linear algebra library and generates platform-optimized BLAS routines by searching over different blocking strategies, operation schedules, and degrees of unrolling. We note that both fast DCT calculation and hardware adaptation are *exact* transform implementations. Our proposed research is complementary to these approaches and will take advantage of prior research.

The DCT approximations based on pruning techniques have been well studied. The pruning techniques save computations by removing the operations on the input coefficients that equal to zero and removing the operations on the output coefficients that have small energy. Only a subset of output coefficients that have higher energy is computed and the rest output coefficients are set to zero directly. In [15–17], several fast 1-D FFT pruning techniques are proposed. The 2-D FFT pruning method is presented in [18]. It saves more computation compared to the row-column pruning method for 2-D FFT. In [19, 20], the authors propose algorithms for pruning 1-D DCT. The 2-D DCT pruning algorithms that are more efficient than row-column pruning method are presented in [21, 22].

There has been prior work on adaptation in multimedia applications. Part 7 of the MPEG-21 standard, entitled digital item adaptation (DIA), has specified a set of description tools for adapting multimedia based on the user characteristics, terminal capabilities, network characteristics, and natural environment characteristics [23, 24]. The system-specific complexity or power optimization have already been thoroughly studied for different multimedia codecs [25–30]. The computational efficient transforms in video coding was proposed in [31, 32]. A number of complexity-scalable coders [33–38] have been proposed to perform real-time coding/decoding under different computational complexity. In more theoretical work [39], the authors look at properties of approximate transform formalisms and [40] look at relationship between Kolmogorov complexity and distortion.

However, several issues remain: (a) while there has been some success in complexity scalable decoders, there are no formal *generic* adaptation strategies to guide us for other content analysis applications, (b) given a specific transform (say DCT) approximation and distortion, there is no framework that enables us to systematically change the approximation in real-time to take advantage of additional computational resources to minimize distortion.

1.2. Our approach

In this paper, we build upon earlier results [41, 42] to develop a novel framework to systematically trade off computational complexity with output distortion, in linear transform approximation, in an optimal manner. We address three problems (shown in Figure 2) in this paper.

(i) estimate the optimal linear transform approximation for *single input* for different computational resources with minimum distortion. We address this problem by showing that a transform can be efficiently factored into two parts—a signal-dependent and a signal-independent calculation. We will use basis projection, pruning and joint pruning, and basis approximation schemes;

(ii) estimate the optimal linear transform approximation for *input set* for different overall computational resources with minimum distortion. We solve this problem by introducing the formalism of a complexity-distortion function using ideas from rate-distortion theory. We then show how approximate this function using an approximate candidate set. Finally, we will present a fast algorithm to transform each input element with an approximation operator, such that we satisfy the computational complexity requirements while minimizing distortion;

(iii) perform the real-time optimal approximation for *input set* that adapts to the available computational resources. We will show how to compute and embed metadata in the image as well as show a decoding algorithm to allow for adaptive approximation. The metadata is embedded by the encoder and the complexity adaptation is done at the decoder.

We have tested our approximation ideas on a widely used linear transform—the DCT. We have used the Haar wavelet basis projection to approximate the transforms and combine it with DCT pruning approximation. Our experimental results on the Lena image are excellent. They show that (a) the joint approximation that combines basis projection and pruning has better results (i.e., better tradeoff of computational complexity and distortion) than using basis projection or pruning alone. (b) Our fast algorithm works well for estimating conditional complexity distortion function (CCDF). The estimation result is close to the exact CCDF. The relative error is 0.039%. (c) We finally show the relationship between the metadata size and introduced distortion.

This submission is our first comprehensive submission on this subject, and includes several new theoretical and experimental results as well as detailed algorithms. In particular, there are several key innovations over prior work [41, 42].

(1) *DCT approximation:* we focus on a joint pruning-basis projection approximation strategy for the DCT in this paper—the prior work focused on FFT approximation using basis projection. This is an important difference as we exploit the unique spectral structure of the DCT for transform-based pruning in our approximation framework.

(2) *New joint pruning-projection approximation strategy:* we improve the basis projection approximation algorithms in earlier work by joint approximation that combines basis projection and pruning. This is a significant improvement, as it significantly extends the earlier theoretical framework using basis projection alone. Importantly, it reveals that incorporating the spectral characteristics of the transform can provide significant gains to approximation. In experiment results, we can clearly see that the complexity distortion curve drops down after combining basis projection and pruning approximation.

(3) *New theoretical proof and detailed algorithms:* real-time adaptive approximation. We show new theoretical proofs for operating point selection. We provide detailed algorithms for metadata embedding and decoding.

(4) *New experimental results:* we discuss how to construct approximation candidate set for each approximation technique in detail. We compare three different approximation techniques (basis projection, pruning, and joint approximation that combines basis projection and pruning) in terms of conditional complexity distortion function. The experimental results show that the joint approximation has less distortion for the same computational complexity. We show the relationship between the metadata size and sampling distortion.

This paper is organized as follows. In Section 2, we define the notations that are used in this paper. In Section 3, we define the optimal approximation for single input and propose three approximation techniques. We apply the three approximation techniques on the DCT and analyze the computational complexity of the approximations in Section 4. In Section 5, we define the optimal approximation for input set and estimate the optimal approximation by using conditional approximation algorithm. In Section 6, we define complexity distortion function and conditional complexity distortion function (CCDF) for linear transform approximation on input set. We also present a fast algorithm to estimate conditional complexity distortion function (CCDF) and propose how to find the conditional approximation based on estimated CCDF. We discuss how to encode and decode metadata for resource adaptive approximations in real time in Section 7. We show the experimental results in Section 8 and conclude the paper in Section 9.

Table 1: Notations with light background are related to single input (e.g., image block). Notations with dark background are related to input set (e.g., entire image).

Notation	Explanation
x	Single input (e.g., image block)
T	Linear transform operator (e.g., DCT)
$\tilde{T}$	Approximate transform operator for a single input
Tx	Result of exact transform for a single input x
$\tilde{T}x$	Result of approximation transform for a single input x
$C(T)$	Computational complexity of the linear transform T for single input (number of operations)
$C(\tilde{T})$	Computational complexity of the approximate transform $\tilde{T}$ for a single input (number of operations)
$\mathbf{X}$	A set of inputs ($\mathbf{X} = \{x_i\},\ i = 1,\dots,N$), where x_i is an element of the input set $\mathbf{X}$ (e.g., image)
N	Number of elements in input set $\mathbf{X}$. $\lvert\mathbf{X}\rvert = N$
$\mathbf{T}$	Linear transform set operator (e.g., DCT) $\mathbf{T} = \{T_i \mid T_i = T,\ i = 1,\dots,N\}$. Each element T_i is the linear transform operator for the corresponding input element x_i. All elements are identical (exact transform T)
$\tilde{\mathbf{T}}$	Approximate transform set for an input set ($\tilde{\mathbf{T}} = \{\tilde{T}_i \mid i = 1,\dots,N\}$). Each element $\tilde{T}_i$ is the approximation operator for the corresponding input element x_i
$\mathbf{TX}$	Result of exact transform for input set $\mathbf{X}$ ($\mathbf{TX} = \{Tx_i\}$)
$\tilde{\mathbf{T}}\mathbf{X}$	Result of approximation transform for input set $\mathbf{X}$ ($\tilde{\mathbf{T}}\mathbf{X} = \{\tilde{T}_i x_i\}$)
$C(\mathbf{T})$	Computational complexity of the linear transform set $\mathbf{T}$ for input set (number of operations)
$C(\tilde{\mathbf{T}})$	Computational complexity of the approximate transform set $\tilde{\mathbf{T}}$ or input set (number of operations)

2. PRELIMINARIES

In this section, we define the notations that are used in the rest of this paper. Table 1 shows a list of notations and their meanings. We separate notations into two categories:

(1) notations related to approximate transform for *single input* (e.g., DCT approximation for an image block);

(2) notations related to approximate transform for *input set* (e.g., DCT approximation for entire image).

The computational complexity of the exact transform set $\mathbf{T}$ and the computation complexity of the approximate transform set $\tilde{\mathbf{T}}$ for any input set $\mathbf{X}$—(i.e., $C(\mathbf{T})$ and $C(\tilde{\mathbf{T}})$) are defined as the average number of operations per input element to compute $\mathbf{T}\mathbf{X}$ and $\widetilde{\mathbf{T}}\mathbf{X}$ for any input set $\mathbf{X}$:

$$C(\mathbf{T}) \triangleq \frac{1}{N}\sum_{i=1}^{N} C(T_i) = C(T), \qquad C(\widetilde{\mathbf{T}}) \triangleq \frac{1}{N}\sum_{i=1}^{N} C(\widetilde{T}_i), \tag{1}$$

where N is the number of elements in input set $\mathbf{X}$ (i.e., $|\mathbf{X}| = N$), since all elements in exact transform set $\mathbf{T}$ are identical (i.e., exact DCT operator T), the average operation number of exact transform set $\mathbf{T}$ equals the operation number of the DCT operator T (i.e., $C(\mathbf{T}) = C(T)$). We use the definition involving the average in (1), as it allows us to analyze the input independent of the input resolution.

Note that in this paper, when we refer to complexity, it is computational complexity of the transform. We will assume that a single real addition, subtraction, or multiplication uses equivalent computing costs and they are all considered to cost one operation. This is also true for some of the DSP chips. The case when the costs are different is easily handled by using appropriate weights in the calculations.

3. TRANSFORM APPROXIMATION FOR SINGLE INPUT

In this section, we will discuss the transform approximation for single input. First, we define the optimal transform approximation for single input x and then discuss our approximation approach.

3.1. Problem statement

The optimal approximate transform $\widetilde{T}_x^*(C)$ for the single input x for desired exact transform T for available computational resource C is defined as follows:

$$\widetilde{T}_x^*(C) \triangleq \mathop{\arg\min}_{\widetilde{T}:C(\widetilde{T})\leq C} d(Tx, \widetilde{T}x), \tag{2}$$

where $d(\cdot)$ is the standard Euclidean metric. The equation indicates that the optimal approximate transform $\widetilde{T}_x^*(C)$ minimizes output distortion while satisfying computational complexity constraints C. In the rest of this section, without loss of generality, we will assume that x is an $M \times 1$ dimensional vector and that the exact transform T and approximate transform $\widetilde{T}$ are both $M \times M$ matrices. The matrix B_k is an $M \times k$ matrix with only k orthogonal column vectors.

3.2. Our approach

We now propose three techniques for linear transform approximation for single input: (a) basis projection approximation, (b) pruning, and (c) joint approximation that combines basis projection and pruning.

3.2.1. Basis projection approximation

The main idea in our basis projection approximation algorithm for the single input involves dimensionality reduction. The approximate transform based on basis projection approximation can be represented as follows:

$$\widetilde{T}x = TB_kB_k^Tx. \tag{3}$$

This decomposition allows us to compute $\widetilde{T}x$ into two steps: (a) project x onto B_k: (i.e., B_k^Tx), then (b) project the result onto TB_k. The significant advantage is that TB_k is *independent of the input*, and can be precomputed and stored offline. We only need compute B_k^Tx and combine with the stored TB_k matrix during real-time computation (Figure 3).

3.2.2. Pruning

The key idea of a pruning algorithm [19, 20] is that we remove the calculations in the exact transform that are only related to the output coefficients with small energy (close to zero).

The pruning operator P is an $M \times M$ diagonal matrix defined as follows:

$$P = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$$
$$\lambda_i = \begin{cases} 1, & \text{if the } i\text{th coefficient of } Tx \text{ is computed,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

If the ith coefficient of transform result Tx is computed, $P(i,i)$ equals 1, otherwise $P(i,i)$ equals 0. The approximation operator $\widetilde{T}$ is the product of T and P.

3.2.3. Joint approximation—combination of basis projection and pruning

The combination (Figure 4) of basis projection and pruning can further reduce the computational complexity for approximating the input. The joint approximation can be represented as follows:

$$\widetilde{T}x = PTB_kB_k^Tx. \tag{5}$$

Compared to basis projection approximation (3), joint approximation saves more calculations in the second projection (PTB_k). This is because that pruning operator P is a diagonal matrix with diagonal coefficients equal to 1 or 0. Hence PTB_k has more zero coefficients than TB_k thus saving calculations.

In Section 4, we will discuss how to apply these three approximation techniques on a DCT for single input (8 × 8 image block).

4. DCT APPROXIMATION FOR IMAGE BLOCK

In this section, we show how the three approximation techniques (discussed in Section 3) can be applied on the 2D DCT for an 8 × 8 image block. We will specifically show the effect of using Haar wavelet basis projection, pruning, and joint approximation using basis projection and pruning.

The DCT for 8 × 8 image block can be represented as a 64 × 64 real matrix. The exact 2D DCT has a *fixed*

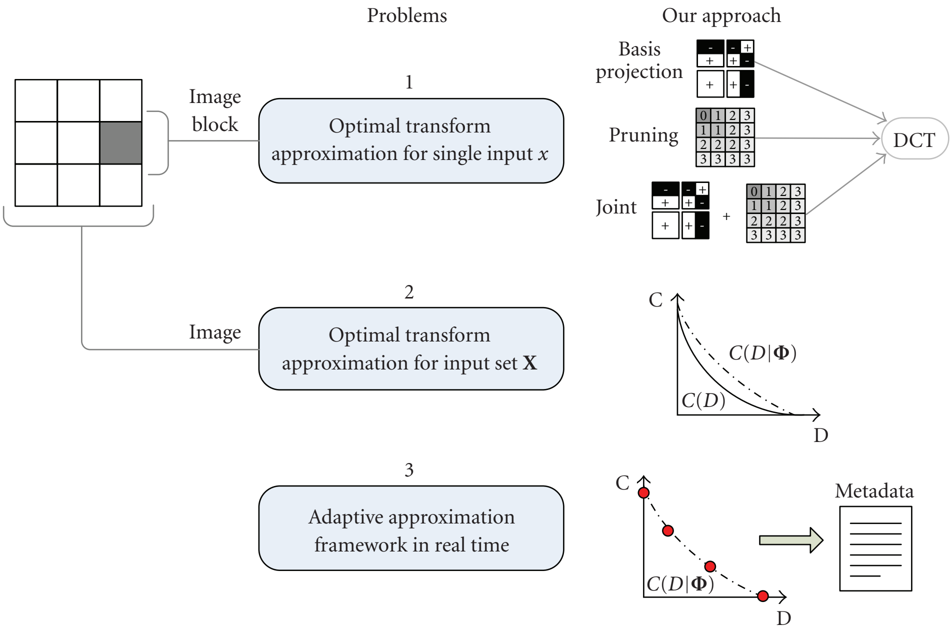

Problems
Our approach
Image block
1
Optimal transform approximation for single input x
Basis projection
Pruning
Joint
DCT
2
Image
Optimal transform approximation for input set $\mathbf{X}$
C
$C(D|\mathbf{\Phi})$
$C(D)$
D
3
Adaptive approximation framework in real time
C
$C(D|\mathbf{\Phi})$
D
Metadata

Figure 2: Three problems addressed in this paper: (1) estimation of optimal approximation for single input, (2) estimation of optimal transform approximation for input set, and (3) real-time adaptive approximation framework through selecting operating points on the conditional complexity distortion function.

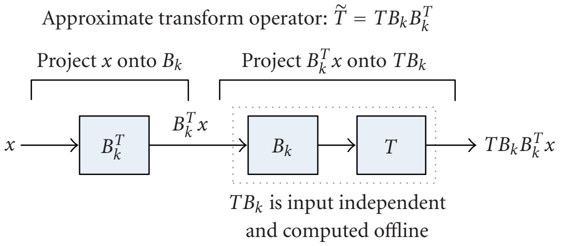

Approximate transform operator: $\widetilde{T} = TB_kB_k^T$
Project x onto B_k
Project $B_k^T x$ onto TB_k
x
B_k^T
$B_k^T x$
B_k
T
$TB_kB_k^T x$
TB_k is input independent and computed offline

Figure 3: Basis projection approximation for single input.

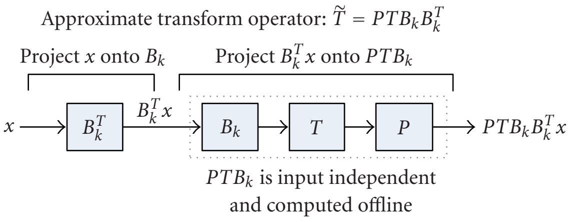

Approximate transform operator: $\widetilde{T} = PTB_kB_k^T$
Project x onto B_k
Project $B_k^T x$ onto PTB_k
x
B_k^T
$B_k^T x$
B_k
T
P
$PTB_kB_k^T x$
PTB_k is input independent and computed offline

Figure 4: Diagram of joint approximation (combination of basis projection approximation and pruning).

computational complexity. The algorithm proposed in [43] makes possible the calculation of an eight point 1D DCT using just 29 additions and 5 multiplications. Thus just total 544 operations (464 additions and 80 multiplications) are needed for the 2D DCT calculation of one 8 × 8 image block. In this section, when we refer to the DCT, it is the scaled DCT [43].

4.1. DCT approximation using Haar wavelet basis projection

In this section, we present DCT approximation on 8 × 8 image block using Haar wavelet basis projection. The 2D nonstandard Haar wavelet basis decomposition [44] for an 8 × 8 image block (i.e., x) can be represented as follows:

$$x'_J = c^0_{0,0}\phi\phi^0_{0,0} + \sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1}\sum_{l=0}^{2^j-1}\left(d^j_{k,l}\phi\psi^j_{k,l} + e^j_{k,l}\psi\phi^j_{k,l} + f^j_{k,l}\psi\psi^j_{k,l}\right), \tag{6}$$

where x'_J is the approximation of image block x using Haar wavelet basis at the Jth resolution, $c^0_{0,0}$ and $\phi\phi^0_{0,0}$ are the scaling coefficient and scaling function, respectively, $d^j_{k,l}$ and $\phi\psi^j_{k,l}$ are the (k,l)th horizontal wavelet coefficient and function at the $(j+1)$th resolution, $e^j_{k,l}$ and $\psi\phi^j_{k,l}$ are the (k,l)th vertical wavelet coefficient and function at the $(j+1)$th resolution, $f^j_{k,l}$ and $\psi\psi^j_{k,l}$ are the (k,l)th diagonal wavelet coefficient and function at the $(j+1)$th resolution.

The 2D Haar wavelet basis can be easily represented using basis matrix B_k. B_k is a $64 \times k$ matrix, each column is a vector representation of basis. k equals 1, 4, and 16 at resolution $J = 0, 1$ and 2, respectively. The higher-resolution basis set includes the basis at the lower resolution. Since Haar wavelet basis are orthogonal, the columns of B_k are orthogonal. We do not consider resolution $J = 3$ because when $J = 3$ the Haar wavelet basis is complete for 8 × 8 image block and the basis projection approximation is equivalent to the exact DCT.

Table 2 shows the computational complexity of DCT approximation using Haar wavelet basis projection. We can

Table 2: Computational complexity (number of operations) of DCT approximation using Haar wavelet basis projection.

Operation	Resolution			
	$J = 0$	$J = 1$	$J = 2$	Exact DCT
Projection onto B_k	63	68	88	
Projection onto TB_k	0	18	184	
Total	63	86	272	544

0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	2	3	4	5	6	7
3	3	3	3	4	5	6	7
4	4	4	4	4	5	6	7
5	5	5	5	5	5	6	7
6	6	6	6	6	6	6	7
7	7	7	7	7	7	7	7

(a)

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14

(b)

Figure 5: Resolution indicator matrices for DCT pruning on an 8 × 8 image block.

see that as the resolution J increases, complexity of projection of input x onto Haar wavelet basis B_k increases slowly while the complexity of projection of $B_k^T x$ onto TB_k increases fast. This is because we can save computations in computing $B_k^T x$ by reusing intermediate results.

4.2. DCT pruning

We now present a 2D DCT pruning approximation framework using rectangle and triangle pruning. Figure 5(a) shows the rectangle pattern of DCT coefficients in DCT approximation using rectangle pruning and Figure 5(b) shows the triangle pattern of DCT coefficients in triangle pruning.

We classify the DCT coefficients into several pruning resolutions based on frequency value for both the rectangle pruning and the triangle pruning. Each coefficient is associated with a resolution indicator. The resolution indicator matrices of the rectangle pruning and the triangle pruning are shown in Figures 5(a) and 5(b), respectively. There are 8 resolutions (J: 0–7) for the rectangle pruning and 15 resolutions (J: 0–14) for the triangle pruning. At resolution J, only the coefficients with resolution number less than or equal to J are computed and remaining coefficients are set to zero. At the lowest resolution ($J = 0$), only the top left coefficient (lowest frequency) is computed and at the highest resolution all coefficients are computed, which is equivalent to the exact DCT. We can define the rectangle pruning operator and triangle pruning operator for the DCT pruning. Both pruning operators can be represented as 64 × 64 diagonal matrices. Figure 5 illustrates the matrix representation of I_R and I_T in the DCT pruning.

In this paper, we use 1D DCT pruning techniques and apply it on row and column separately. In the future, we will use 2D DCT pruning which can be easily incorporated in our framework.

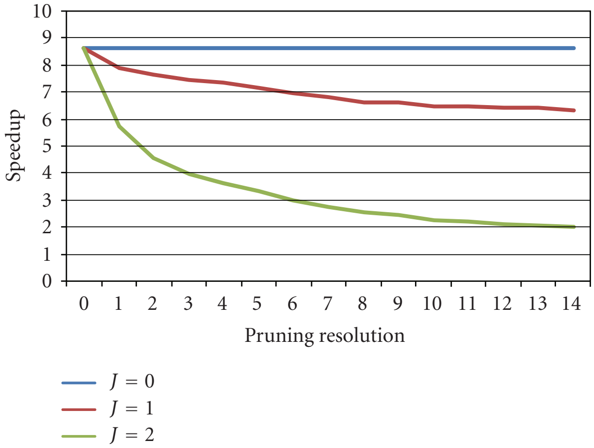


Figure 6: The figure shows the speedup of the approximate 2D DCT under joint Haar projection (three resolutions, $J = 0, 1, 2$) with triangular pruning when compared to the baseline, the exact 2D DCT (544 operations). The x axis shows the pruning resolution, the y-axis shows the speedup.

4.3. Joint DCT approximation

We compute the joint DCT approximation through combining Haar wavelet basis projection and the DCT pruning. The combination yields significant savings when compared to the baseline exact 2D DCT (544 operations). Figure 6 shows a plot of the speedup achieved when using the joint DCT approximation combining *triangle* pruning and Haar wavelet basis projection at three different Haar resolutions, when compared to the baseline, exact DCT transform. The speedup is just the ratio of the number of operations needed for the exact 2D DCT (544 operations) to the number of operations needed for the approximation.

Increasing the pruning resolution implies that more coefficients in the triangular pruning matrix (Figure 5(b)) are nonzero. This is why the speedup decreases with increasing pruning resolution. Similarly, when the Haar wavelet resolution increases, speedup decreases as the number of basis elements increases. The graph for the rectangular pruning case has been omitted for the sake of brevity and is similar to Figure 6.

In this section, we applied the three approximation techniques (*basis projection*, *pruning*, and *joint approximation* Section 3) on the 2D DCT for an 8 × 8 image block and analyzed the computational complexity.

5. TRANSFORM APPROXIMATION FOR INPUT SET

In this section, we define the technical problem of linear transform approximation for *input set* and present our approach. Let us explain the problem of approximation for input set by an example. Let us assume that we need to compute the DCT approximation for all 8 × 8 image blocks of a given image. Each image block is a *single input* and the entire image is the *input set*. The problem is to select proper approximation operator for each image block such that the overall transform computational complexity satisfies the resource complexity constraint and the overall distortion is minimized. We will first define the optimal approximation for input set and then propose our approach.

In this paper, we define the computational complexity constraint C and the computational complexity and distortion of approximation for input set in the sense of *average* per input element. We use the definition involving the average, as it allows us to analyze the input independent of the input resolution. We acknowledge that the complexity constraint, computational complexity and distortion can also be defined in terms of summation over all input elements in the input set. In the following of the paper, when we refer to the computational complexity constraint C, the computational complexity and distortion of approximation for input set, they are all in the sense of average per input element.

5.1. Optimal approximation for input set $\mathbf{X} = \{x_i\}$

We now define the optimal approximation for an input set:

> *the optimal approximation operators $\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C)$ for input set $\mathbf{X} = \{x_i\}$, for a linear transform T, for a average computational complexity constraint C is defined as a set of approximation operators (i.e., $\widetilde{\mathbf{T}}^*_{\mathbf{X}} = \{\widetilde{T}_i\}$) such that the average computation complexity per input element satisfies the average complexity constraint and the average distortion is minimized.*

Formally, the definition can be represented as follows:

$$\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C) \triangleq \underset{\widetilde{\mathbf{T}}:\widetilde{\mathbf{T}}=\{\widetilde{T}_i\},\, C(\widetilde{\mathbf{T}})\leq C}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} d\left(Tx_i, \widetilde{T}_i x_i\right), \tag{7}$$

where $C(\widetilde{\mathbf{T}})$ is the computational complexity of approximation set $\widetilde{\mathbf{T}}$ (1), x_i is the ith element (e.g., image block) of the input set $\mathbf{X}$ (e.g., entire image), $\widetilde{T}_i$ is the ith element in approximation set $\widetilde{\mathbf{T}}$ that indicates the approximation operator for the input element x_i and N is the cardinality of the input set $\mathbf{X}$ ($|\mathbf{X}| = N$). Note that, $d(\cdot)$ represents the standard Euclidean metric. Note that, $\widetilde{T}_i$ is an approximation operator for a *single input*. The equation indicates that the optimal approximation $\widetilde{\mathbf{T}}^*_{\mathbf{X}}$ has minimum average output distortion while satisfying computational complexity constraint C. The optimal approximation set $\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C)$ is related to the computational complexity constraint C and the input set $\mathbf{X}$. Furthermore, in (7), the optimization is over *all* possible approximations to the operator T. Formally, this is equivalent to the halting problem and hence not computable. Note, however, with additional constraints (e.g., reduced approximation space to a finite approximation set), we can determine a conditional approximation. We will discuss the conditional approximation in Section 6.

5.2. Our approach

We now propose our approach to estimate the optimal approximation for input set $\mathbf{X}$. The key idea is that we reduce the dimension of approximation operator space by constraining the approximate operator $\widetilde{T}_i$ for every input element x_i to be in a *finite* approximation candidate set $\mathbf{\Phi}$ (i.e., $\widetilde{T}_i \in \mathbf{\Phi}$). Let us explain how to construct the finite approximation candidate set $\mathbf{\Phi}$ by an example. Let us assume we compute the DCT approximation for all image blocks using the Haar wavelet basis projection (Section 4.1). Hence we have four options of DCT approximation for every image block, that is, DCT approximation using Haar wavelet basis projection at resolution $J = 0, 1, 2$ (denoted as $\widetilde{T}^0_H$, $\widetilde{T}^1_H$, $\widetilde{T}^2_H$) and exact DCT operator T. Therefore, we can use these four operators to construct $\mathbf{\Phi} = \{\widetilde{T}^0_H, \widetilde{T}^1_H, \widetilde{T}^2_H, T\}$.

We now define the conditional approximation for input set using finite approximation candidate set $\mathbf{\Phi}$:

> *the conditional approximation set $\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C \mid \mathbf{\Phi})$ for an input set $\mathbf{X} = \{x_i\}$, for a linear transform T, for an average complexity constraint C for a given approximation candidate set $\mathbf{\Phi}$ defined as a set approximation operators such that*:
>
> (1) *each element (i.e., $\widetilde{T}_i$) belongs to $\mathbf{\Phi}$*;
>
> (2) *the average computation complexity satisfies the average complexity constraint C;*
>
> (3) *the average distortion is minimized.*

Mathematically, the conditional approximation for input set is defined as follows:

$$\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C \mid \mathbf{\Phi}) \triangleq \underset{\widetilde{\mathbf{T}}:\widetilde{T}_i\in\mathbf{\Phi},\, C(\widetilde{\mathbf{T}})\leq C}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} d\left(Tx_i, \widetilde{T}_i x_i\right). \tag{8}$$

The equation indicates that every approximation operator element in $\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C \mid \mathbf{\Phi})$ (i.e., $\widetilde{T}_i$) belongs to the approximation candidate set $\mathbf{\Phi}$ and the conditional approximation $\widetilde{\mathbf{T}}^*_{\mathbf{X}}(C \mid \mathbf{\Phi})$ has minimum average output distortion while satisfying average computational complexity constraint C.

In Section 6, we will address the conditional approximation for input set $\mathbf{X}$ in detail by introducing the formalism of the conditional complexity distortion function.

6. THE COMPLEXITY DISTORTION FUNCTION

In this section, we will propose a complexity distortion framework to address the approximation for an input set $\mathbf{X}$ for any complexity constraint C (8). We solve this problem in three steps. First, we present a theoretical definition for the *complexity distortion function*. Second, we show how the complexity distortion function can be approximated by specifying an approximation candidate set $\mathbf{\Phi}$. Finally, we show an algorithm to select the optimal approximation candidate set $\mathbf{\Phi}^*$ from multiple approximation candidate sets.

6.1. Definition

We now discuss the complexity distortion function for linear transform approximation given an input set $\mathbf{X}$. The problem can be stated as follows: given an input set $\mathbf{X} = \{x_i\}$ and a distortion measure D, what is the minimum distortion achievable at a specific computational complexity constraint? Or, equivalently, what is the minimum computational complexity required to achieve a particular distortion?

We use the well-established definitions from rate distortion theory [45] to define the relationship between the computational complexity and distortion. The computational complexity of transform approximation set $C(\widetilde{\mathbf{T}})$ is defined in (1). We now define the distortion due to the transform approximation as follows.

Definition 1. The *distortion* $D_{\mathbf{X}}(\widetilde{\mathbf{T}})$ due to a transform approximation set $\widetilde{\mathbf{T}} = \{\widetilde{T}_i\}$ for a transform T on an input set $\mathbf{X} = \{x_i\}$ is defined as follows:

$$D_{\mathbf{X}}(\widetilde{\mathbf{T}}) = \frac{1}{N}\sum_{i=1}^{N} D_{x_i}(\widetilde{T}_i) = \frac{1}{N}\sum_{i=1}^{N} d(Tx_i, \widetilde{T}_i x_i), \tag{9}$$

where $\mathbf{X}$ is a set of inputs ($\mathbf{X} = \{x_i\}$, $i = 1,\ldots,N$), x_i is the ith element of the input set $\mathbf{X}$, N is the cardinality of the input set, $\widetilde{\mathbf{T}}$ is an approximation set ($\widetilde{\mathbf{T}} = \{\widetilde{T}_i\}$, $i = 1,\ldots,N$), each element $\widetilde{T}_i$ is the approximation operator for the corresponding input x_i, $D_{xi}(\widetilde{T}_i)$ is the distortion due to the approximation $\widetilde{T}_i$ for the ith element of the input set (i.e., x_i), and $d(\cdot)$ is the distortion measure. In this paper, $d(\cdot)$ is the standard Euclidean norm.

Definition 2. The *complexity distortion region* is the closure of the set of achievable complexity distortion pairs (C, D). This definition is similar to the definition of the rate distortion region in rate distortion theory [45].

Definition 3. The *complexity distortion function* $C_{\mathbf{X}}^{T}(D)$ for an input set $\mathbf{X}$, for the approximation of linear transform T, is defined as the infimum of all complexities C such that (C, D) is in the achievable complexity distortion region for a given distortion D.

$$C_{\mathbf{X}}^{T}(D) = \operatorname*{Inf}_{\widetilde{\mathbf{T}}: D_{\mathbf{X}}(\widetilde{\mathbf{T}}) \le D} C(\widetilde{\mathbf{T}}), \tag{10}$$

where $C_{\mathbf{X}}^{T}(D)$ is the complexity distortion function of approximation of linear transform T for an input set $\mathbf{X}$, $C(\widetilde{\mathbf{T}})$ and $D_{\mathbf{X}}(\widetilde{\mathbf{T}})$ are the computational complexity (1) and distortion (9) of transform approximation $\widetilde{\mathbf{T}}$ for the input set $\mathbf{X}$, respectively. In the case of DCT of image, each image ($\mathbf{X}$) has a complexity distortion function $C_{\mathbf{X}}^{T}(D)$ for a particular transform approximation T (DCT). It is straightforward to show that the complexity distortion function is nonincreasing and convex. These properties are used in estimating the complexity distortion function.

6.2. Conditional complexity distortion function (CCDF)

In this section, we discuss how we can estimate the complexity distortion function, given a set of approximation operators. The conditional complexity distortion allows us to estimate the C-D curve in practice. This is because the complexity distortion function (10) is a theoretical lower bound, obtained via a search over all possible approximations of T. In practice, we need to define a set of approximation operators on T so that we determine the complexity distortion function conditioned on that approximation strategy.

Assume that we have a *finite* approximation candidate set $\mathbf{\Phi}$. Then similar to the definitions in Section 6.1, it is straightforward to define a conditional complexity distortion region and a conditional complexity distortion function.

Specifically, the *conditional complexity distortion function* (CCDF) $C_{\mathbf{X}}^{T}(D \mid \mathbf{\Phi})$ for an input set $\mathbf{X}$, for the approximation of linear transform T, is defined as the infimum of all complexities C such that (C, D) is in the conditional complexity distortion region achieved by using approximation candidate set $\mathbf{\Phi}$ for a given distortion:

$$C_{\mathbf{X}}^{T}(D \mid \mathbf{\Phi}) = \operatorname*{Inf}_{\widetilde{\mathbf{T}}: \widetilde{T}_i \in \mathbf{\Phi},\, D_X(\widetilde{\mathbf{T}}) \le D} C(\widetilde{\mathbf{T}}), \tag{11}$$

where $C(\widetilde{\mathbf{T}})$ and $D_{\mathbf{X}}(\widetilde{\mathbf{T}})$ are the computational complexity (1) and distortion (9) of transform approximation $\widetilde{\mathbf{T}}$ for the input set $\mathbf{X}$, respectively.

Estimation of the complexity distortion function is a challenging computational problem. Let Q denote the cardinality of the given approximation candidate set $\mathbf{\Phi}$ and let N the cardinality of the input set $\mathbf{X}$. Then the number of possible achievable C-D pairs is N^Q. Therefore, computational cost of searching the lower bound of achievable complexity distortion region is exponential in Q. In order to reduce the computational cost, we developed a fast stepwise algorithm that is linear in Q to estimate CCDF.

We now outline a fast stepwise algorithm to estimate the conditional complexity distortion function (details can be found in the appendix). Let us assume that the approximation candidate set $\mathbf{\Phi}$ has Q elements $\mathbf{\Phi} = \{\Phi_j,\ j = 1,\ldots,Q,\ C(\Phi_1) \ge C(\Phi_2) \ge \cdots \ge C(\Phi_Q)\}$. We start assigning all input elements x_i with the highest computational complexity approximation in the approximation candidate set $\mathbf{\Phi}$ (i.e., $\widetilde{T}_i = \Phi_1$, $i = 1,\ldots,N$). At each step, we try to find one input element such that by changing its approximation to the lower complexity approximation in $\mathbf{\Phi}$ (e.g., $\Phi_1 \rightarrow \Phi_2$ or $\Phi_2 \rightarrow \Phi_3$), we are able to minimize the slope of distortion increment with respect to complexity decrement. We repeat this procedure until all input elements use the lowest computational complexity approximation in $\mathbf{\Phi}$ (i.e., $\widetilde{T}_i = \Phi_Q$, $i = 1,\ldots,N$).

Intuitively, we are looking for that location in the image for which reducing the complexity of the approximation has minimum effect on distortion. This strategy is equivalent to traversing the D-C curve, starting from the highest complexity, lowest distortion value to the lowest complexity, highest distortion point. Our fast algorithm only generates $NQ - N + 1$ complexity-distortion (C-D) pairs, where

N and Q are the cardinality of the input set $\mathbf{X}$ and the approximation candidate set $\mathbf{\Phi}$, respectively (i.e., $N = |\mathbf{X}|$, $Q = |\mathbf{\Phi}|$).

6.3. Optimal approximation set selection

We now show how we can determine the optimal approximation candidate set $\mathbf{\Phi}^*$ from multiple approximation candidate sets (e.g., $\mathbf{\Phi_1}, \mathbf{\Phi_2}, \ldots, \mathbf{\Phi}_W$). This is useful since for every linear transform, there exist many options to construct the approximation candidate set $\mathbf{\Phi}$.

We use the average distortion of conditional distortion complexity function (CDCF) (Section 6.2) to evaluate the approximation candidate set $\mathbf{\Phi}$. Then the optimal approximation candidate set $\mathbf{\Phi}^*_{\mathbf{X}}(T)$ for an input set $\mathbf{X}$ for the linear transform T is defined as the approximation candidate set with minimum average distortion:

$$\mathbf{\Phi}^*_{\mathbf{X}}(T) = \underset{\mathbf{\Phi}\in\mathbf{\Psi}}{\arg\min}\left[\delta^T_{\mathbf{X}}(\mathbf{\Phi})\right], \tag{12}$$

where $\delta^T_{\mathbf{X}}(\mathbf{\Phi})$ is the average distortion of CDCF for input set $\mathbf{X}$, for the linear transform T and for a given approximation candidate set $\mathbf{\Phi}$ and $\mathbf{\Psi}$ is a set that includes multiple approximation candidate sets (i.e., $\mathbf{\Psi} = \{\mathbf{\Phi}_i,\ i = 1, \ldots, W\}$).

7. REAL-TIME RESOURCE ADAPTIVE APPROXIMATION

In this section, we present a real-time adaptive framework for linear transform approximation on input set $\mathbf{X}$ using conditional complexity distortion function (CCDF). The main idea is that we *sample* the CCDF using several operating points and *store* operating points as part of the input metadata at the encoder.

Hence we can use the operating points embedded by the encoder as part of the metadata to perform *adaptive* approximation at the decoder. We select the proper operating point in the metadata that satisfies the complexity constraint and use its corresponding conditional approximation to perform the approximation at the decoder.

We will discuss this method in detail over the next few sections. First, we present an algorithm to determine the optimal operating points. Second, we show the structure of metadata. Finally, we show how to decode the metadata for adaptive approximation in real time.

7.1. Operating point selection

We now present an iterative algorithm to determine the optimal operating points on the distortion complexity function $D^T_{\mathbf{X}}(C)$ (Section 6.1). For the sake of simplicity, we use $D(C)$ to represent distortion complexity function $D^T_{\mathbf{X}}(C)$. It is straightforward to extend the algorithm to the conditional complexity distortion function.

Assume that we wish to sample the $D(C)$ function using K points. We can denote the K operating points on $D(C)$ as a set $\mathbf{\Omega}_K = \{(C_k, D_k),\ k = 1, \ldots, K,\ C_1 \le \cdots \le C_K,\ D_1 \ge \cdots \ge D_K\}$. When the available complexity C is in the interval $[C_k, C_{k+1})$, the operating point (C_k, D_k) is used because it introduces minimum distortion amongst all operating points while satisfying the complexity constraint ($C_k \le C$). The result distortion is $D_k - D(C)$. We call this distortion as *sampling distortion* because it is introduced by sampling the distortion complexity function using the operating points. The overall sampling distortion $d_s(\mathbf{\Omega}_K)$ due to K operating points $\mathbf{\Omega}_K$ on the D-C function is computed as follows:

$$d_s(\mathbf{\Omega}_K) = \sum_{k=0}^{K}\int_{C_k}^{C_{k+1}} p(C)\left[D_k - D(C)\right]dC, \tag{13}$$

where $\mathbf{\Omega}_K$ contains the K operating points on $D(C)$ ($\mathbf{\Omega}_K = \{(C_k, D_k),\ k = 1, \ldots, K\}$), (C_0, D_0) and (C_{K+1}, D_{K+1}) are two extreme points, ($C_0 \le C_1 \le \cdots \le C_K \le C_{K+1}$, $D_0 \ge D_1 \ge \cdots \ge D_K \ge D_{K+1}$), $p(C)$ is the pdf of the complexity constraint. We define the set $\mathbf{\Omega}^*_K$ with minimum sampling distortion to be the one with the optimal K operating points on $D(C)$. Formally, it is defined as follows:

$$\mathbf{\Omega}^*_K = \underset{\mathbf{\Omega}_K}{\arg\min}\, d_s(\mathbf{\Omega}_K), \tag{14}$$

where $d_s(\mathbf{\Omega}_K)$ is the sampling distortion (13). In each of the small figures in Figure 7, the area of dark region is proportional to the sampling distortion when $p(C)$ is a uniform distribution.

We now discuss our algorithm to iteratively determine the K operating points that minimize sampling distortion. The intuition behind the algorithm rests on two ideas: (a) operating points that are globally optimal are also locally optimal (the proof is straightforward) (b) given two operating points on the D-C curve, it we can determine an operating point between the two that minimizes sampling distortion. This latter idea is repeatedly used in our algorithm.

We first show how to compute the optimal operating point given two extrema. Let us assume that we wish to determine the operating point $\mathbf{\Omega}_1 = (C_1, D_1)$, that lies between (C_0, D_0) and (C_2, D_2). That is, ($C_0 \le C_1 \le C_2$, $D_0 \ge D_1 \ge D_2$). The problem is to find the optimal (C_1, D_1) to minimize the sampling distortion. We proceed by splitting the sampling distortion as follows:

$$\begin{aligned} d_s(\mathbf{\Omega}_1) &= \int_{C_0}^{C_1} p(C)\left[D_0 - D(C)\right]dC + \int_{C_1}^{C_2} p(C)\left[D_1 - D(C)\right]dC \\ &= \underbrace{-\left(D_0 - D_1\right)\left[F(C_2) - F(C_1)\right]}_{1} \\ &\quad \underbrace{+\,D_0\left[F(C_2) - F(C_0)\right] - \int_{C_0}^{C_2} p(C)D(C)\,dC,}_{2} \end{aligned} \tag{15}$$

where F is cumulative distribution function for $p(C)$. Since the second part of (15) is only related to the extreme points (C_0, D_0) and (C_2, D_2) which are fixed, it is a constant. Thus minimizing the sampling distortion is equivalent to minimizing the first part of (15). Therefore, the optimal operating point can be obtained as follows:

$$\begin{aligned}\boldsymbol{\Omega}_1^* &= (C_1^*, D_1^*),\\ C_1^* &= \underset{C_0 \le c \le C_2}{\arg\max}\left([D_0 - D(c)]\cdot[F(C_2) - F(c)]\right),\\ D_1^* &= D(C_1^*).\end{aligned} \tag{16}$$

Once, we can determine an optimal operating point between two extrema, the iterative algorithm is shown in Algorithm 1 (Figure 7 illustrates the iteration procedure).

7.2. *Encoding metadata*

In this section, we discuss the metadata that needs to be embedded at the encoder, to allow the decoder to approximate the transform T in an adaptive manner, in response to changing computational constraints. We need to know three things in order to adaptively approximate the transform at the decoder side. They include (a) the optimal approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$ (12), (b) the operating points (C, D) along the conditional complexity distortion function (CCDF) for $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$, and (c) the approximation operator $\widetilde{T}_i$ for every input element x_i.

Let us assume that we have W approximation candidate sets $\boldsymbol{\Phi}_{\mathbf{1}}, \ldots, \boldsymbol{\Phi}_{\mathbf{W}}$. For the sake of simplicity, let us assume without loss of generality that these W sets have the same cardinality Q. First, estimate the conditional complexity distortion function (CCDF) for all approximation candidate sets $(\boldsymbol{\Phi}_{\mathbf{1}}, \ldots, \boldsymbol{\Phi}_{\mathbf{W}})$ and select the approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$ with the minimum average distortion (12). Then given the optimal approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$, select K optimal operating points along the conditional distortion complexity function (CCDF). Each optimal operating point is associated with an approximation index list $\mathbf{L_k}$.

The metadata contains the following information.

(1) Approximation candidate set indicator—the index of the optimal candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$.

(2) Complexity distortion pairs for $(K + 2)$ operating points (K operating points on the C-D curve and two extreme points).

(3) K approximation index lists $\mathbf{L_k}$. Each operating point (C_k, D_k) $(k = 1, \ldots, K)$ is associated with an approximation index list $\mathbf{L_k}$. The cardinality of each approximation index list $\mathbf{L_k}$ is the same as the number of elements in the input set $\mathbf{X}$ $(|\mathbf{L_k}| = |\mathbf{X}| = N)$. The element of list $\mathbf{L_k}(i)$ indicates the approximation operator $\widetilde{T}_i$ for the corresponding input element x_i. For example, if we use the jth operator in the approximation candidate set $\boldsymbol{\Phi}$ (i.e., Φ_j), for the ith input element x_i (i.e., $\widetilde{T}_i = \Phi_j$) then $\mathbf{L_k}(i) = j$.

The inclusion of the metadata has a size penalty. The approximation candidate set indicator needs $\log_2 W$ bits. The $K + 2$ *operating points* need $32(K + 2)$ bits if we use 16 bit precision to represent complexity and distortion values. And finally, the K *approximation index lists* need $KN(\log_2 Q)$ bits, where Q and N are the cardinality of optimal approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$ and the cardinality of input set $\mathbf{X}$, respectively. Hence the overall metadata size S is

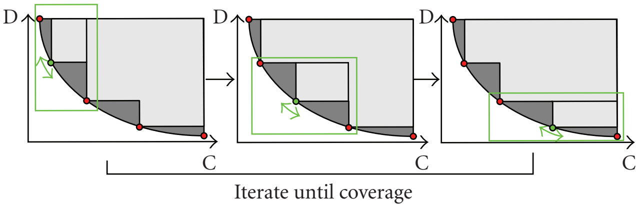


Figure 7: Iteration for optimal multiple selection.

$$S = K(N(\log_2 Q) + 32) + (\log_2 W + 64). \tag{17}$$

If the approximation candidate sets $(\boldsymbol{\Phi}_{\mathbf{1}}, \ldots, \boldsymbol{\Phi}_{\mathbf{W}})$ and the input set $\mathbf{X}$ are given, Q, N, and W are fixed. Then the metadata size is a linear function of the number of operating points K on the distortion complexity function $D(C)$. The selection of K can be influenced by application-dependent constraint on metadata size.

7.3. *Real-time decoding*

We now show how the decoder can use the metadata embedded at the encoder for real-time adaptive approximation. Let us assume the input set $\mathbf{X}$ and computational complexity constraint C are given. The decoding includes four steps.

(1) The approximation candidate set indicator is used to select the optimal approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$ (12).

(2) Then we select the operating point (C_k, D_k) such that $C_{k+1} > C \ge C_k$ from the operating points saved in the metadata.

(3) We determine the approximation index list $\mathbf{L_k}$ corresponding to the selected operating point (C_k, D_k) and assign the approximation selection for each input. For example, if $\mathbf{L_k}(i) = j$, we select the jth approximation in the approximation candidate set $\boldsymbol{\Phi}_{\mathbf{X}}^*(T)$ for the ith input element x_i.

(4) Finally, we perform approximation for every input element using its assigned approximation operator $\widetilde{T}_i$.

The complexity of this approximation is guaranteed to be less than the complexity constraint C.

In this section, we addressed the problem of real-time adaptive approximation. First, we presented an algorithm to select K operating points (C_k, D_k) along the conditional distortion complexity function (CDCF). Second, we encode the operating points (C_k, D_k) and associated approximation index lists $\mathbf{L_k}$ into metadata as part of input. Finally, we used the embedded metadata to perform real-time approximation at the decoder.

```
Input: distortion complexity function $D(C)$, number of operating points $K$, two extreme points $(C_0, D_0)$ and $(C_{K+1}, D_{K+1})$.
Output: $K$ operating points $\mathbf{\Omega}_K = \{(C_k, D_k),\ k = 1, \ldots, K\}$.
    1. Initialization—randomly select $K$ points on the distortion complexity function $D(C)$, sort them in ascending order of complexity value, and compute the sampling distortion $d$.
    2. for $j = 1 : K$.
           Update $(C_j, D_j)$ with the optimal single selection (16) of the subcurve of $D(C)$ from $(C_{j-1}, D_{j-1})$ to $(C_{j+1}, D_{j+1})$.
       end
    3. Update sampling distortion $d$.
    4. If the sampling distortion no longer decreases, stop, otherwise go to step 2.
```

Algorithm 1: Iterative algorithm.

8. EXPERIMENTAL RESULTS

In this section, we present our experimental results for

(i) estimate the conditional complexity distortion function. CCDF (Section 6.2).

(ii) compare three different approximation techniques: basis projection, pruning, and joint approximation (Section 3).

(iii) select optimal operating points (Section 7.1).

We have used a well-known image—Lena at resolution 256 × 256 and 64 × 64 to test our framework. The Lena image at resolution 64 × 64 as the input set $\mathbf{X}$ is used for estimation of conditional complexity distortion (CCDF). We select the resolution 64 × 64 rather than 256 × 256 here because of the high computational complexity of searching *exact* CCDF (Section 6.2). We use Lena image at resolution 256 × 256 as the input set $\mathbf{X}$ to compare three approximation techniques and to test operating point selection. In this section, let us denote the exact DCT as T^{DCT}.

8.1. Estimation of conditional complexity distortion function (CCDF)

We now present our experimental results for estimating the conditional complexity distortion function (CCDF). We select DCT as the linear transform T^{DCT} and construct DCT approximation candidate set $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$ by four DCT approximation operators: $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}} = \{\widetilde{T}_H^{\text{DCT}}(0), \widetilde{T}_H^{\text{DCT}}(1), \widetilde{T}_H^{\text{DCT}}(2), T^{\text{DCT}}\}$, where $\widetilde{T}_H^{\text{DCT}}(0)$, $\widetilde{T}_H^{\text{DCT}}(1)$, and $\widetilde{T}_H^{\text{DCT}}(2)$ are DCT approximation for 8 × 8 image block using Haar wavelet basis projection approximation at resolution $J = 0, 1, 2$, respectively, and T^{DCT} is the exact DCT operator for 8 × 8 image block. The cardinality of $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$ is four ($|\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}| = 4$). We select Lena image with resolution 64 × 64 as the input set $\mathbf{X}$ that contains 64 8 × 8 image blocks ($|\mathbf{X}| = 64$). We select the resolution 64 × 64 rather than 256 × 256 here because the computational complexity of searching *exact* CCDF (Section 6.2) increases exponentially with the number of image blocks. The number of achievable C-D pairs for Lena 256 × 256 and Lena 64 × 64 are 1024^4 and 64^4, respectively. Therefore, computing the *exact* CCDF for Lena 256 × 256 is very expensive.

Table 3: Approximation techniques and their approximation candidate set notations.

Approximation techniques	Approximation candidate set notations
Haar wavelet basis projection	$\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$
Rectangle pruning	$\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$
Triangle pruning	$\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}$
Joint approximation (rectangle pruning + Haar wavelet basis projection)	$\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$
Joint approximation (triangle pruning + Haar wavelet basis projection)	$\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$

We use the relative difference between the average distortions of exact CCDF and estimated CCDF to evaluate our fast estimation algorithm. The relative difference is defined as follows:

$$\tau = \frac{\delta' - \delta}{\delta}, \tag{18}$$

where δ and δ' represent the average distortion of exact CCDF and our estimate, respectively ($\delta' > \delta$). The relative difference in estimating the conditional complexity distortion function for DCT approximation for Lena 64 × 64 using approximation candidate set $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$ is $\tau = 1.30\%$. This shows that our fast CCDF estimation algorithm works well.

8.2. Comparing different approximation techniques

In this section, we compare three approximation techniques:

(i) basis projection approximation (Section 3.2.1);

(ii) pruning (Section 3.2.2);

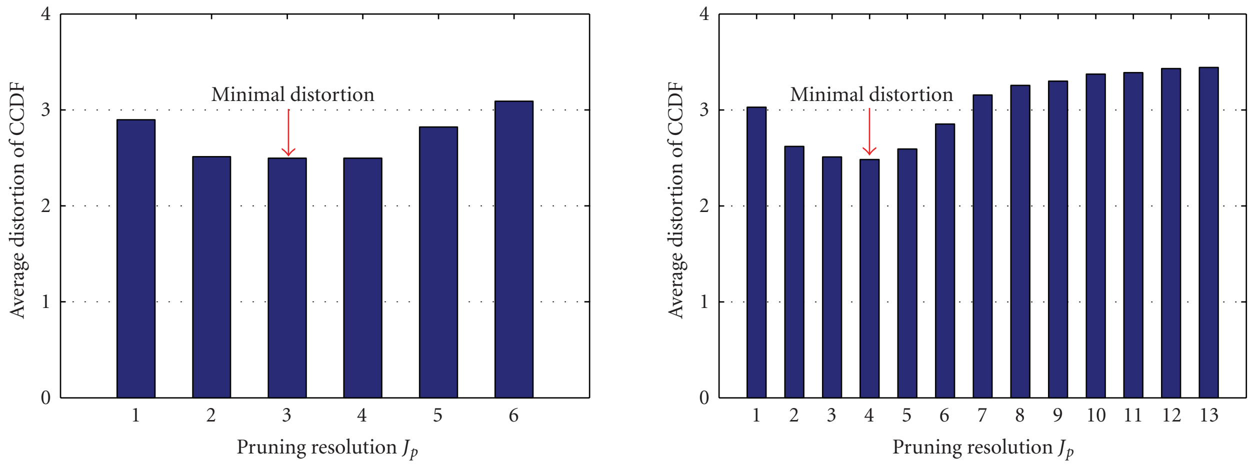
(iii) joint approximation that combines basis projection approximation and pruning (Section 3.2.3).

We use Lena image at resolution 256 × 256 as the input set $\mathbf{X}$ and we apply these three approximation techniques on the DCT. We use the fast estimation of conditional complexity distortion function (CCDF) and use the average distortion as the evaluation metric.

8.2.1. Constructing approximation candidate set

We now show how to construct approximation candidate set $\mathbf{\Phi}$ (Section 5.2) for each approximation technique. We construct five approximation candidate sets. Table 3 shows the notations of these five approximation candidate sets and their corresponding approximation techniques.

For the sake of consistency, every approximation candidate set has four approximation operators (e.g., $|\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}| = |\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}| = |\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}| = |\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}| = |\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}| = 4$) and every approximation candidate set includes the exact transform



(a) Rectangle pruning + Haar wavelet basis projection

(b) Triangle pruning + Haar wavelet basis projection

FIGURE 8: Average distortion of CCDF for Lena image 256 × 256 for joint DCT approximation using combination of DCT pruning and Haar wavelet basis projection at different pruning resolutions J_p. At each pruning resolution J_p, the approximation candidate set $\mathbf{\Phi}$ includes combination of DCT pruning at pruning resolution J_p and Haar wavelet basis projection at basis resolution $J_b = 0, 1, 2$ and exact DCT.

TABLE 4: Approximation candidate set construction for all DCT approximation techniques for Lena image 256 × 256. Each approximation candidate set has four elements (Φ_1–Φ_4). The five approximation candidate sets share the same Φ_1 (only computing the lowest coefficient) and the same Φ_4 (exact DCT).T^{DCT} is the exact DCT for 8 × 8 image block, respectively.$\widetilde{T}_H^{\text{DCT}}(j)$ is DCT approximation using Haar wavelet basis projection at basis resolution$J_b = j$.$\widetilde{T}_R^{\text{DCT}}(j)$ and$\widetilde{T}_T^{\text{DCT}}(j)$ are DCT pruning using rectangle pruning and triangle pruning at pruning resolution$J_p = j$, respectively.$\widetilde{T}_{R+H}^{\text{DCT}}(j,k)$ is joint DCT approximation combining rectangle pruning and Haar wavelet basis projection at pruning resolution$J_p = j$ and basis resolution$J_b = k$.$\widetilde{T}_{T+H}^{\text{DCT}}(j,k)$ is joint DCT approximation combining triangle pruning and Haar wavelet basis projection at pruning resolution$J_p = j$ and basis resolution$J_b = k$.

Transform	Candidate set	Approximation operator elements			
		Φ_1	Φ_2	Φ_3	Φ_4
DCT	$\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$	$\widetilde{T}_H^{\text{DCT}}(0)$	$\widetilde{T}_H^{\text{DCT}}(1)$	$\widetilde{T}_H^{\text{DCT}}(2)$	T^{DCT}
	$\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$	$\widetilde{T}_R^{\text{DCT}}(0)$	$\widetilde{T}_R^{\text{DCT}}(1)$	$\widetilde{T}_R^{\text{DCT}}(2)$	
	$\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}$	$\widetilde{T}_T^{\text{DCT}}(0)$	$\widetilde{T}_T^{\text{DCT}}(1)$	$\widetilde{T}_T^{\text{DCT}}(2)$	
	$\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$	$\widetilde{T}_{R+H}^{\text{DCT}}(3,0)$	$\widetilde{T}_{R+H}^{\text{DCT}}(3,1)$	$\widetilde{T}_{R+H}^{\text{DCT}}(3,2)$	
	$\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$	$\widetilde{T}_{T+H}^{\text{DCT}}(4,0)$	$\widetilde{T}_{T+H}^{\text{DCT}}(4,1)$	$\widetilde{T}_{T+H}^{\text{DCT}}(4,2)$	

(DCT) and the lowest complexity approximation (only compute the lowest frequency coefficient in DCT). Note that the lowest complexity approximations for all DCT approximation techniques presented in this paper are equivalent. They all compute the lowest frequency coefficient and require 63 operations. For the sake of completeness, we note that the $C = 0$ (constant output) is the lowest complexity case, but is ignored here as this is unlikely to be of practical interest. We describe the construction of five approximation candidate set for DCT in details as follows.

Haar wavelet basis projection ($\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$)—the DCT approximation candidate set using Haar wavelet basis projection (Section 4.1)—$\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$ includes DCT approximation using Haar wavelet basis projection at basis resolution $J_b = 0, 1, 2$ and exact DCT. $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}}$ can be represented as $\mathbf{\Phi}_{\mathbf{H}}^{\mathbf{DCT}} = \{\widetilde{T}_H^{\text{DCT}}(0), \widetilde{T}_H^{\text{DCT}}(1), \widetilde{T}_H^{\text{DCT}}(2), T^{\text{DCT}}\}$, where $\widetilde{T}_H^{\text{DCT}}(0)$, $\widetilde{T}_H^{\text{DCT}}(1)$, and $\widetilde{T}_H^{\text{DCT}}(2)$ are DCT approximation for 8 × 8 image block using Haar wavelet basis projection approximation at resolution $J = 0, 1, 2$, respectively, and T^{DCT} is the exact DCT operator.

Pruning ($\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$ and $\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}$)—we now present the construction of DCT approximation candidate set for rectangle pruning (Section 4.2)—$\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$. In the similar manner, we can construct $\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}$. There are eight pruning resolutions in DCT rectangle pruning operator. The minimum pruning resolution $J_p = 0$ (computing the lowest frequency coefficient) and maximum pruning resolution $J_p = 7$ (exact DCT) are included in the $\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$. We need to select another two resolutions (J_p^a, J_p^b) from $J_p = 1, 2, 3$. Let us denote the DCT approximation using rectangle pruning at resolution J_p as $\widetilde{T}_R^{\text{DCT}}(J_p)$. We choose the resolution pair (J_p^a, J_p^b) such that the approximation candidate set $\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}} = \{\widetilde{T}_R^{\text{DCT}}(0), \widetilde{T}_R^{\text{DCT}}(J_p^a), \widetilde{T}_R^{\text{DCT}}(J_p^b), T^{\text{DCT}}\}$ has the minimum average distortion of conditional complexity distortion

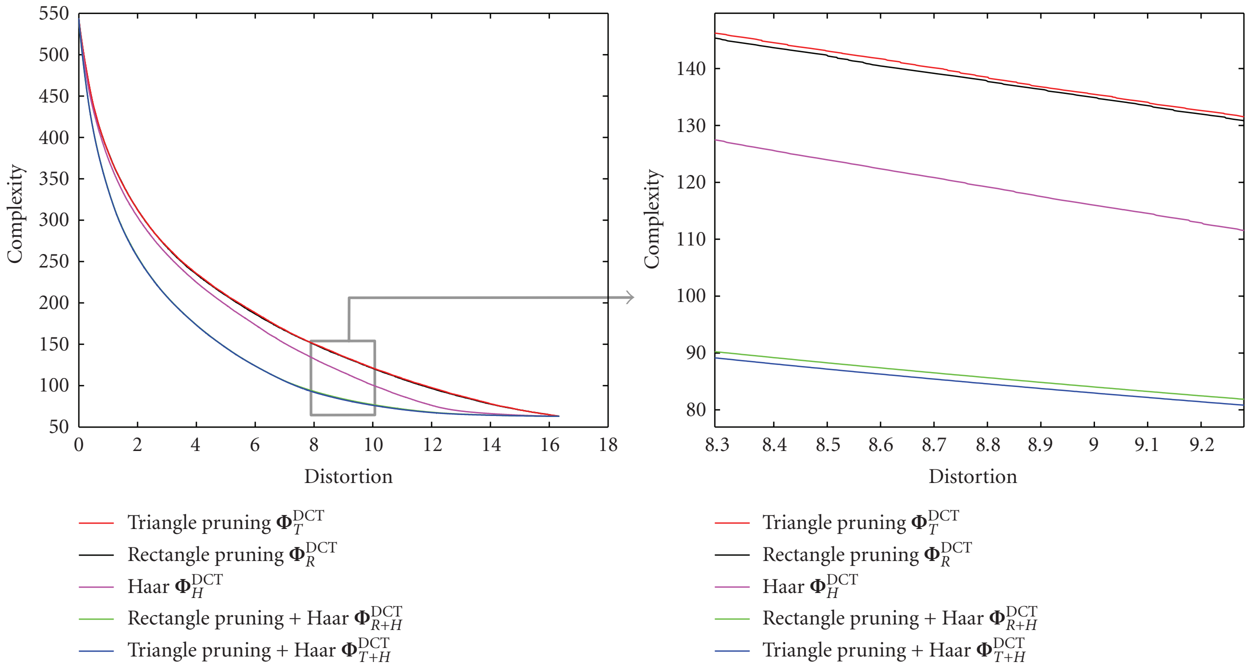


Figure 9: Conditional complexity distortion function (CCDF) for Lena image 256 × 256 for five DCT approximation candidate sets.

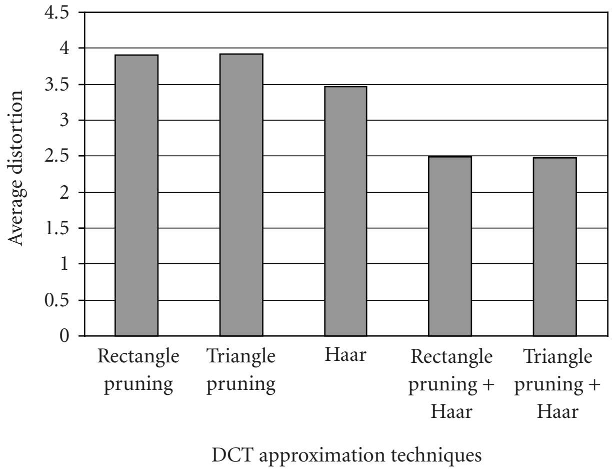


Figure 10: Average distortion of conditional complexity distortion function (CCDF) for Lena image 256 × 256 for five DCT approximation candidate sets.

function (CCDF) $\delta_{\mathbf{X}}^{\mathrm{DCT}}(\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}})$ over $3C_2$ possible resolution pairs (J_p^a, J_p^b). The $\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$ for the input set $\mathbf{X}$ is computed as follows:

$$\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}(\mathbf{X}) = \underset{\substack{\Phi=\{\widetilde{T}_R^{\mathrm{DCT}}(0),\, \widetilde{T}_R^{\mathrm{DCT}}(J_p^a),\, \widetilde{T}_R^{\mathrm{DCT}}(J_p^b),\, T^{\mathrm{DCT}}\} \\ 1\le J_p^a<J_p^b\le 3}}{\arg\min} \left[\delta_{\mathbf{X}}^{\mathrm{DCT}}(\mathbf{\Phi})\right]. \tag{19}$$

For Lena image 256 × 256, it is straightforward to show that the selection $(J_p^a = 1, J_p^b = 2)$ has the minimum average distortion. Hence $\mathbf{\Phi}_{\mathbf{R}}^{\mathbf{DCT}}$ includes the DCT rectangle pruning at pruning resolution $J_p = 0, 1, 2$ and exact DCT (equivalent to $J_p = 7$). Similarly, we can show that $\mathbf{\Phi}_{\mathbf{T}}^{\mathbf{DCT}}$ includes DCT triangle pruning at pruning resolution $J_p = 0, 1, 2$ and exact DCT.

Joint approximation that combines basis projection and pruning ($\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$ and $\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$)—we now discuss the construction of approximation candidate set for joint DCT approximation that combines Haar wavelet basis projection and DCT rectangle pruning (Section 4.3)—$\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$. In the similar manner, we can create $\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$. The key idea is that we select an optimal pruning resolution J_p^* and use the joint approximation that combines DCT pruning at resolution J_p^* and Haar basis projection at basis resolution $J_b = 0, 1, 2$ and exact DCT to construct $\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$. The optimal is in the sense of minimum average distortion of conditional complexity distortion function (CCDF) $\delta_{\mathbf{X}}^{\mathrm{DCT}}(\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}})$. Let us denote the joint DCT approximation using rectangle pruning at resolution J_p and Haar wavelet basis projection at basis resolution J_b as $\widetilde{T}_{R+H}^{\mathrm{DCT}}(J_p, J_b)$. The $\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$ for the input set $\mathbf{X}$ is computed as follows:

$$\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}(\mathbf{X}) = \underset{\substack{\Phi=\{\widetilde{T}_{R+H}^{\mathrm{DCT}}(J_p,0),\, \widetilde{T}_{R+H}^{\mathrm{DCT}}(J_p,1),\, \widetilde{T}_{R+H}^{\mathrm{DCT}}(J_p,2),\, T^{\mathrm{DCT}}\} \\ 1\le J_p\le 3}}{\arg\min} \left[\delta_{\mathbf{X}}^{\mathrm{DCT}}(\mathbf{\Phi})\right]. \tag{20}$$

For Lena image 256 × 256, the pruning resolution with minimum average distortion is $J_p = 3$ (shown in Figure 8(a)). Hence $\mathbf{\Phi}_{\mathbf{R+H}}^{\mathbf{DCT}}$ includes the combination of DCT pruning at pruning resolution $J_p = 3$ and Haar wavelet basis projection at basis resolution $J_b = 0, 1, 2$ and exact DCT. In the similar manner, we can construct $\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$. The optimal pruning resolutions J_p to construct $\mathbf{\Phi}_{\mathbf{T+H}}^{\mathbf{DCT}}$ is 4 (shown in Figure 8(b)).

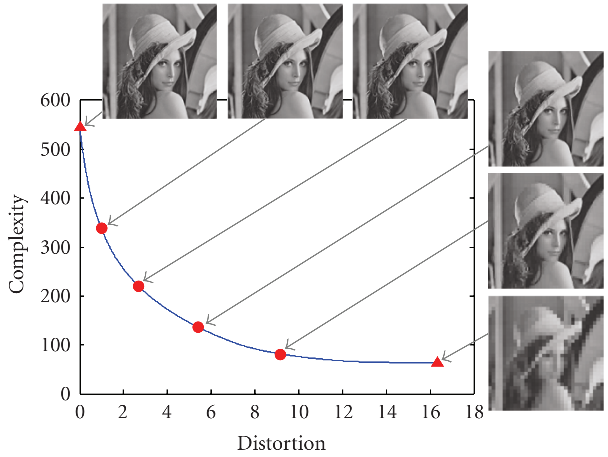


Figure 11: Optimal operating point selection along the conditional complexity distortion function (CCDF) using DCT approximation candidate set $\mathbf{\Phi}_{\mathrm{T+H}}^{\mathrm{DCT}}$ (joint DCT approximation that combines triangle pruning and Haar wavelet basis projection) for Lena image 256 × 256. Each operating point is associated with a recovered image that is obtained by using exact inverse DCT on the DCT approximation result.

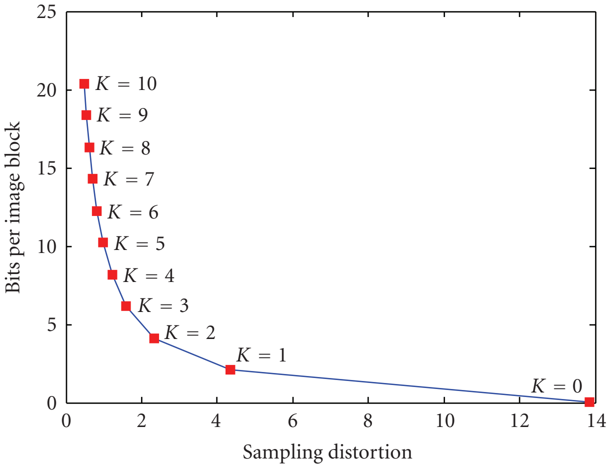


Figure 12: Tradeoff between average saving load of metadata (bits per image block) and sampling distortion. K: number of operating points on the C-D curve.

Table 4 shows the approximation operator elements for all approximation candidate sets for DCT approximation for Lena image 256 × 256. Note that each operator element is used for approximation of 8 × 8 image blocks. Note also that the first elements Φ_1 of all approximation candidate sets are equivalent. This is just the computation of the lowest frequency coefficient of DCT result and it requires 63 operations.

8.2.2. *Results*

We now discuss the experimental results of DCT approximation using five approximation candidate sets (Table 4). Figure 9 shows the CCDFs based on five DCT approximation candidate sets (Table 4). Figure 10 plots the average distortions of CCDF for these five DCT approximation candidate sets.

We have two observations:

(1) *Haar > Pruning*—the approximation *only* using Haar wavelet basis projection is *better* than the approximation *only* using pruning (rectangle or triangle pruning).

(2) *Joint > Haar*—the joint approximation is *better* than both the approximation *only* using Haar wavelet basis approximation and the approximation *only* using pruning.

The term "better" is in the sense of lower conditional complexity distortion function (CCDF) that results in lower average distortions of CCDF. We use ">" to represent "is better than."

The result *Haar > Pruning* holds true because the Haar wavelet basis projection can approximate the low-frequency coefficients of DCT with small distortion while costing less computations than the pruning operator. For the same distortion D, the Haar wavelet basis projection needs less computational resources C than pruning. Let us explain this in terms of complexity and distortion in detail. The approximation candidate set for Haar wavelet basis projection ($\mathbf{\Phi}_{\mathrm{H}}^{\mathrm{DCT}}$) only have two approximation operators (Φ_2 and Φ_3 Table 4) that are different from the approximation candidate set for pruning ($\mathbf{\Phi}_{\mathrm{R}}^{\mathrm{DCT}}/\mathbf{\Phi}_{\mathrm{T}}^{\mathrm{DCT}}$). Hence we only compare the Haar wavelet basis at basis resolution basis $J_b = 1, 2$ (Φ_2 and Φ_3 in $\mathbf{\Phi}_{\mathrm{H}}^{\mathrm{DCT}}$) to the pruning at pruning resolution $J_p = 1, 2$ (Φ_2 and Φ_3 in $\mathbf{\Phi}_{\mathrm{R}}^{\mathrm{DCT}}/\mathbf{\Phi}_{\mathrm{T}}^{\mathrm{DCT}}$).

(i) *Complexity*—the computational complexity of DCT approximation only using Haar wavelet basis projection at basis resolution $J_b = 2$ (272 operations Table 2) is close to the complexity of DCT pruning at pruning resolution $J_p = 2$ (286 operations for rectangle pruning and 262 operations for triangle pruning). However, the DCT approximation using Haar wavelet basis projection at basis resolution $J_b = 1$ (86 operations) requires significantly less computation than DCT pruning at pruning resolution $J_p = 1$ (210 operations for rectangle pruning and 196 operations for triangle pruning).

(ii) *Distortion*—the DCT approximation using Haar wavelet basis projection at basis resolution $J_b = 1, 2$ does not introduce more distortion compared to DCT pruning at pruning resolution $J_p = 1, 2$. This is because of two reasons: (a) the DCT coefficients that are *not dropped* in the DCT pruning at pruning resolution $J_p = 1, 2$ (Figure 5) are approximated with small distortion by the DCT approximation using Haar wavelet basis projection at basis resolution $J_b = 1, 2$, respectively, (b) some of higher-frequency coefficients that are *dropped* in the DCT pruning at pruning resolution $J_p = 1, 2$ are still approximated with small distortion by the DCT approximations using Haar wavelet basis projection at basis resolution $J_b = 1, 2$.

Input: input set $\mathbf{X} = \{x_i,\ i = 1, \ldots, N\}$ and approximation candidate set $\mathbf{\Phi} = \{\Phi_j,\ j = 1, \ldots, Q,\ C(\Phi_1) \geq C(\Phi_2) \geq \cdots \geq C(\Phi_Q)\}$.
Output: estimation of CCDF-(C_k, D_k) and the corresponding approximation index list $\mathbf{L_k}$.

1. Initialization—using approximation Φ_1 for all input elements and obtain the first C-D pair (C_1, D_1) and the first approximation index list $\mathbf{L_1}(\mathbf{L_1}(i) = 1,\ i = 1, \ldots, N)$, $k = 1$.
2. Compute the ratio between distortion increment and complexity decrement for the input elements that do *not* use the lowest complexity approximation Φ_Q as follows:
$$r_i = \frac{D_{x_i}(\Phi_{\mathbf{L_k}(i)+1}) - D_{x_i}(\Phi_{\mathbf{L_k}(i)})}{C(\Phi_{\mathbf{L_k}(i)}) - C(\Phi_{\mathbf{L_k}(i)+1})} \quad \text{if } \mathbf{L_k}(i) < Q,$$
where $C(\cdot)$ is the complexity operator for single input (Table 1), $D_{x_i}(\cdot)$ is the distortion operator for the single input x_i (9), $\mathbf{L_k}$ is the approximation index list associated with the C-D pair (C_k, D_k), each element of $\mathbf{L_k}$—(i.e., $\mathbf{L_k}(i)$) indicates the selection of approximation for the corresponding input element x_i, Φ_j is the jth element of approximation candidate set $\mathbf{\Phi}$.
3. Find the input element x_i with the minimum ratio r_i, change its approximation operator $\widetilde{T}_i$ to the lower complexity approximation operator in $\mathbf{\Phi}$ (e.g., $\widetilde{T}_i \leftarrow \Phi_{j+1}$ if the current $\widetilde{T}_i$ is Φ_j), obtain a new C-D pair (C_{k+1}, D_{k+1}) and new approximation index list $\mathbf{L_{k+1}} = \{\mathbf{L_k}(1), \ldots, \mathbf{L_k}(i-1), \mathbf{L_k}(i) + 1, \mathbf{L_k}(i+1), \ldots, \mathbf{L_k}(N)\}$, $k = k + 1$.
4. If all input elements use the lowest complexity approximation in $\mathbf{\Phi}$—(i.e., Φ_Q), stop, otherwise go to step 2.

Algorithm 2: Fast stepwise algorithm.

The result *Joint* > *Haar* holds true because in the joint case by combining the basis projection with the pruning operator, we save computation in approximating high-frequency coefficients. Since these high-frequency coefficients typically have small energy, directly setting these high coefficients zero only introduces a small distortion but saves significant number of computations.

8.3. Optimal operating point selection

We now present our experimental results for optimal operating point selection (Section 7.1). Figure 11 shows the optimal operating point selection ($K = 4$) results on the estimated CCDF for Lena image 256 × 256 using approximation candidate set $\mathbf{\Phi}^{\mathbf{DCT}}_{\mathbf{T+H}}$ (Table 3). For each operating point, we also show the corresponding recovered image by using exact inverse DCT (IDCT). The triangles in the figure are the two extreme points and the dots are the 4 optimal operating points.

Figure 12 shows the tradeoff between the metadata size (17) and sampling distortion (13). K operating points are selected on the CCDF for Lena image 256 × 256 based on the approximation candidate set $\mathbf{\Phi}^{\mathbf{DCT}}_{\mathbf{T+H}}$ (DCT approximation using the combination of Haar wavelet basis projection and triangle pruning). In Figure 12, we use bits per image block to represent the metadata size. We can see that as the number of operating points K on the C-D curve increases, the sampling distortion decreases and the metadata size (bits per image block) increases. We also find that when $K > 4$, the sampling distortion decrease very slowly, but the metadata size increase significantly. If we select $K = 4$, the bits per image block is about 8 bits which means the metadata size is about 1/64th of the gray image.

9. CONCLUSION

In this paper, we have attempted to create a systematic framework for linear transform approximation. There were three key ideas: (a) we presented the basis projection approximation technique and combined it with pruning approximation techniques, (b) we proposed an algorithm to estimate the complexity distortion function and search for optimal transform approximation using several approximation candidate sets. We also proposed a measure to select the optimal approximation candidate set, and (c) we presented an adaptive approximation framework in which the operating points on the C-D curve are embedded in the metadata. Our approach is generic, and applies to any linear transform.

First, we developed an efficient Haar wavelet basis projection framework to approximate a widely used multimedia transform—the DCT. We showed how approximations to the input signal as well as the transform output can efficiently trade off computational complexity for signal distortion. *Second*, we presented a theoretical definition of the complexity distortion function, using ideas from rate-distortion theory and proposed conditional complexity distortion function (CCDF) to estimate C-D function. We also presented a fast CCDF estimation algorithm and showed how to estimate the optimal transform approximation. *Finally,* we showed how to compute the optimal operating points on the CCDF curve and embed their information into the image metadata. We also presented a framework to perform adaptive approximation in real time for changing computational resources by using this metadata.

Our experimental results on the Lena image are excellent. They showed (a) that combination of the input approximation (basis projection) and output transform approximation (pruning) have the best results. (b) Our CCDF estimation algorithm is close to the exact CCDF. The relative error is 0.039%. (c) We additionally showed the relationship between the metadata size and the introduced distortion.

In our paper, we did not consider the issue of transform coefficient quantization, an important real world issue. The quantization of the coefficients will have the effect of shifting right the C-D curves. However, this needs further elaboration as the exact effect in general will be content dependent.

This result can be extended in many directions. We plan to combine different basis projection techniques (e.g., Haar and polynomials), for more efficient basis approximation.

We plan to incorporate metadata size as a constraint of our approximation algorithm. We are hopeful that the results in this paper along with the metadata size constraint can allow us to develop a joint complexity distortion rate (C-D-R) optimization framework, thus trading off complexity versus distortion versus rate.

APPENDIX

ESTIMATION OF THE CONDITIONAL COMPLEXITY DISTORTION FUNCTION

We now describe the algorithm for fast estimation of the conditional complexity distortion function in detail. Let us assume that the approximating candidate set $\mathbf{\Phi}$ has Q elements. For each C-D pair (C_k, D_k), that is part of the C-D curve, we generate an *approximation index list* $\mathbf{L_k}$. The cardinality of each approximation index list $\mathbf{L_k}$ is the same as the number of elements in the input set $\mathbf{X}$ ($|\mathbf{L_k}| = |\mathbf{X}| = N$). The purpose of the list is to indicate the specific approximation to be performed at the specific location in the image, that is, each element of list $\mathbf{L_k}(i)$ indicates the approximation operator $\widetilde{T}_i$ for the corresponding input element x_i. For example, if we use the jth operator in the approximation candidate set $\mathbf{\Phi}$ (i.e., Φ_j), for the ith input element x_i (i.e., $\widetilde{T}_i = \Phi_j$), then $\mathbf{L_k}(i) = j$. The approximation operator for the input element x_i can be also represent as $\widetilde{T}_i = \Phi_{\mathbf{L_k}(i)}$, the detailed algorithm is shown in Algorithm 2.

Our fast stepwise algorithm saves significant computations. Our algorithm only generates $NQ - N + 1$ C-D pairs (linear in Q) while searching the exact conditional complexity distortion function requires computing N^Q C-D pairs.

ACKNOWLEDGMENT

This research work is supported by the NSF Grant no. 03-08268: "Development of Quality-Adaptive Media-Flow Architectures to Support Sensor Data Management."

REFERENCES

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

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

[3] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1243–1245, 1984.

[4] S. C. Chan and K. L. Ho, "A new two-dimensional fast cosine transform algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 481–485, 1991.

[5] P. Duhamel and C. Guillemot, "Polynomial transform computation of the 2D DCT," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '90)*, vol. 3, pp. 1515–1518, Albuquerque, NM, USA, April 1990.

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

[7] J. Makhoul, "A fast cosine transform in one and two dimensions," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 27–34, 1980.

[8] P. Duhamel and H. H'Mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '87)*, pp. 1805–1808, Dallas, Tex, USA, April 1987.

[9] E. Feig and S. Winograd, "On the multiplicative complexity of discrete cosine transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1387–1391, 1992.

[10] C. Loeffer, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '89)*, vol. 2, pp. 988–991, Glasgow, UK, May 1989.

[11] A. Molino, F. Vacca, and T. Nguyen, "Energy tradeoffs for DSP-based implementation of IntDCT," in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 2166–2170, Pacific Grove, Calif, USA, November 2003.

[12] M. Puschel, J. M. F. Moura, J. R. Johnson, et al., "SPIRAL: code generation for DSP transforms," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 232–275, 2005.

[13] J. Demmel, J. Dongarra, V. Eijkhout, et al., "Self-adapting linear algebra algorithms and software," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 293–312, 2005.

[14] R. C. Whaley, A. Petitet, and J. J. Dongarra, "Automated empirical optimizations of software and the ATLAS project," *Parallel Computing*, vol. 27, no. 1-2, pp. 3–35, 2001.

[15] Z. Hu and H. Wan, "A novel generic fast Fourier transform pruning technique and complexity analysis," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 274–282, 2005.

[16] J. D. Markel, "FFT pruning," *IEEE Transactions on Audio Electroacoustics*, vol. 19, no. 4, pp. 305–311, 1971.

[17] H. V. Sorensen and C. S. Burrus, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1184–1200, 1993.

[18] K. S. Knudsen and L. T. Bruton, "Recursive pruning of the 2D DFT with 3D signal processing applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1340–1356, 1993.

[19] Y.-M. Huang, J.-L. Wu, and C.-L. Chang, "A generalized output pruning algorithm for matrix-vector multiplication and its application to compute pruning discrete cosine transform," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 561–563, 2000.

[20] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Transactions on Signal Processing*, vol. 42, no. 7, pp. 1833–1837, 1994.

[21] C. A. Christopoulos and A. N. Skodras, "Pruning the two-dimensional fast cosine transform," in *Proceedings of the 7th European Signal Processing Conference (EUSIPCO '94)*, pp. 596–599, Scotland, UK, September 1994.

[22] A. Silva and A. Navarro, "Fast 8 × 8 DCT pruning algorithm," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, pp. 317–320, Genova, Italy, September 2005.

[23] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: the 21st century multimedia framework," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 53–62, 2003.

[24] A. Vetro and C. Timmerer, "Digital item adaptation: overview of standardization and research activities," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 418–426, 2005.

[25] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 704–716, 2003.

[26] G. Landge, M. van der Schaar, and V. Akella, "Complexity metric driven energy optimization framework for implementing MPEG-21 scalable video decoders," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '05)*, vol. 2, pp. 1141–1144, Philadelphia, Pa, USA, March 2005.

[27] M. Mattavelli and S. Brunetton, "Implementing real-time video decoding on multimedia processors by complexity prediction techniques," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 760–767, 1998.

[28] M. Ravasi, M. Mattavelli, P. Schumacher, and R. Turney, "High-level algorithmic complexity analysis for the implementation of a motion-JPEG2000 encoder," in *Proceedings of the 13th International Workshop on Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation (PATMOS '03)*, pp. 440–450, Torino, Italy, September 2003.

[29] S. Saponara, K. Denolf, G. Lafruit, C. Blanch, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 2, pp. 220–235, 2004.

[30] J. Valentim, P. Nunes, and F. Pereira, "Evaluating MPEG-4 video decoding complexity for an alternative video complexity verifier model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1034–1044, 2002.

[31] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, 2003.

[32] S. Srinivasan and S. Regunathan, "Computationally efficient transforms for video coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 2, pp. 325–328, Genova, Italy, September 2005.

[33] F. Argenti, F. Del Taglia, and E. Del Re, "Audio decoding with frequency and compleixty scalability," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 149, no. 3, pp. 152–158, 2002.

[34] Y. Chen, Z. Zhong, T.-H. Lan, S. Peng, and K. van Zon, "Regulated complexity scalable MPEG-2 video decoding for media processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 8, pp. 678–687, 2002.

[35] I. R. Ismaeil, A. Docef, F. Kossentini, and R. K. Ward, "A computation-distortion optimized framework for efficient DCT-based video coding," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 298–310, 2001.

[36] M. Mattavelli, S. Brunetton, and D. Mlynek, "Computational graceful degradation for video sequence decoding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '97)*, vol. 1, pp. 330–333, Santa Barbara, Calif, USA, October 1997.

[37] W. Pan and A. Ortega, "Complexity-scalable transform coding using variable complexity algorithms," in *Proceedings of the Data Compression Conference (DDC '00)*, pp. 263–272, Snowbird, Utah, USA, March 2000.

[38] M. van der Schaar and P. H. N. D. de With, "Near-lossless complexity-scalable embedded compression algorithm for cost reduction in DTV receivers," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 923–933, 2000.

[39] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M. Winograd, and J. T. Ludwig, "Approximate signal processing," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 15, no. 1-2, pp. 177–200, 1997.

[40] D. M. Sow and A. Eleftheriadis, "Complexity distortion theory," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 604–608, 2003.

[41] Y. Chen and H. Sundaram, "Basis projection for linear transform approximation in real-time applications," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 2, pp. 637–640, Toulouse, France, May 2006.

[42] Y. Chen and H. Sundaram, "A framework for linear transform approximation using orthogonal basis projection," *Journal of Multimedia*, vol. 2, no. 3, pp. 26–35, 2007.

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

[44] E. J. Stollnitz, T. D. Derose, and D. H. Salestin, "Wavelets for computer graphics: a primer. 1," *IEEE Computer Graphics and Applications*, vol. 15, no. 3, pp. 76–84, 1995.

[45] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.