Scalable Variable Complexity Approximate Forward DCT

Krisda Lengwehasatit
PacketVideo Corporation
10350 Science Center Dr.
San Diego, California, 92121

Antonio Ortega
Integrated Media Systems Center
and
Signal and Image Processing Institute
Department of Electrical Engineering
University of Southern California,
Los Angeles, California 90089-2564

Corresponding Author:
Antonio Ortega
Phone: (213) 740-2320
Fax: (213) 740-4651
E-mail: ortega@sipi.usc.edu

Parts of this work have been published in [1], [2].

Abstract

The discrete cosine transform (DCT) is one of the major components in many image and video compression systems. Variable complexity algorithms have been applied successfully to achieve complexity savings in image/video decoders by reducing the computation cost of the inverse DCT. These gains can be achieved due to the highly predictable sparseness of the quantized DCT coefficients in natural video/image data. Given the increasing demand for instant video messaging and two-way video transmission over mobile communication systems running on general-purpose embedded processors, there is now an increased need for faster forward DCT, so that overall encoding (and not only decoding) complexity can be reduced. The forward DCT, unlike the inverse DCT, does not operate on sparse input data, but rather generates sparse output data. Thus, complexity reduction can also be achieved for the forward DCT, but with different methods than those used for the inverse DCT. In the literature, two major approaches have been applied to speed up the forward DCT computation, namely, *frequency selection*, in which only a subset of DCT coefficients is computed, and *accuracy selection*, in which all the DCT coefficients are computed at reduced accuracy. These two approaches can achieve significant computation savings with only minor degradation of output quality, as long as the coding parameters are such that the quantization error is larger than the error due to the approximate DCT computation. Thus, in order to be useful, these algorithms have to be combined with efficient mechanisms that can select the "right" level of approximation as a function of the characteristics of the input and the target rate, a selection that is often based on heuristic criteria. In this paper, we consider two previously proposed fast, variable complexity, forward DCT algorithms, one based on frequency selection, the other based on accuracy selection. We provide an explicit analysis of the additional distortion that each scheme introduces as a function of the quantization parameter and the variance of the input block. This analysis then allows us to improve the performance of these algorithms by making it possible to select the best approximation level for each block and a target quantization parameter. We also propose a hybrid algorithm that combines both forms of complexity reduction in order to achieve overall better performance over a broader range of operating rates. We show how our techniques lead to scalable implementations where complexity can be reduced if needed, at the cost of small reductions in video quality. Our hybrid algorithm can speed up the DCT and quantization process by close to a factor of 4 as compared to fixed-complexity forward DCT implementations, with only a slight quality degradation in PSNR.

Keywords

Forward DCT, Variable Complexity Algorithm, Scalable Complexity, Approximate DCT, SSAVT, Approximation Error Thresholding.

I. Introduction

The discrete cosine transform (DCT) has been adopted as an essential part of well-known transform block-based image/video compression standards, such as JPEG, MPEG1-2-4 and ITU's H.263. Each basis vector in the DCT domain represents a spatial frequency component of the image. Those bases have been proved to provide good energy compaction for natural images. Another reason for its popularity is also the availability of several fast algorithms [3].

The N point DCT $\bar{\mathbf{X}}$ of vector input $\bar{\mathbf{x}} = [x(0), x(1), ..., x(N-1)]^T$ is defined as $\bar{\mathbf{X}} = \mathbf{D_N} \cdot \bar{\mathbf{x}}$ where $\mathbf{D_N}$ is the transformation matrix of size NxN with elements $\mathbf{D_N}(i,j)$

$$\mathbf{D_N}(i,j) = c_i \sqrt{\frac{2}{N}} \cdot \cos \frac{(2j+1)i\pi}{2N} \tag{1}$$

where $c_i = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } i = 0 \\ 1 & \text{for } i > 0 \end{cases}$ Due to the orthogonality property of the DCT, the inverse transformation can be written as $\bar{\mathbf{x}} = \mathbf{D_N}^T \cdot \bar{\mathbf{X}}$. For 2-D signals, a separable version of the transform is used, which can be defined as $\mathbf{X} = \mathbf{D_N} \cdot \mathbf{x} \cdot \mathbf{D_N}^T$ and $\mathbf{x} = \mathbf{D_N}^T \cdot \mathbf{X} \cdot \mathbf{D_N}$ for forward and inverse DCT, respectively, where $\mathbf{X}$ and $\mathbf{x}$ are now 2-D matrices. This means that we can implement the 2-D transform using a simpler 1-D transform along each direction separately.

Many fast DCT algorithms have been proposed to reduce the number of typical arithmetic operations (e.g., multiplications and additions) such as [4], [5], [6], etc. The minimal number of multiplications required for a 1-D DCT transform was derived in [7]. Loeffler *et. al.* [8] achieves this theoretical bound for size-8 DCT (11 multiplications). It also has been shown that a fast algorithm for 2-D DCT requires fewer arithmetic operations than using 2 fast 1-D algorithms separately ([9],[10], etc.). Several other algorithms have been proposed aiming for different criteria. For example, the well-known AAN algorithm [11] computes a scaled version of the DCT with only 5 multiplications per 1-D size 8 DCT. For lossy coding, the scaling part can be combined with the quantization process.

In this paper, we focus on variable complexity algorithms (VCAs) that can adjust the forward DCT complexity as a function of the target quantization to be used. Thus, we will present algorithms that provide faster performance when quantization is coarser. The

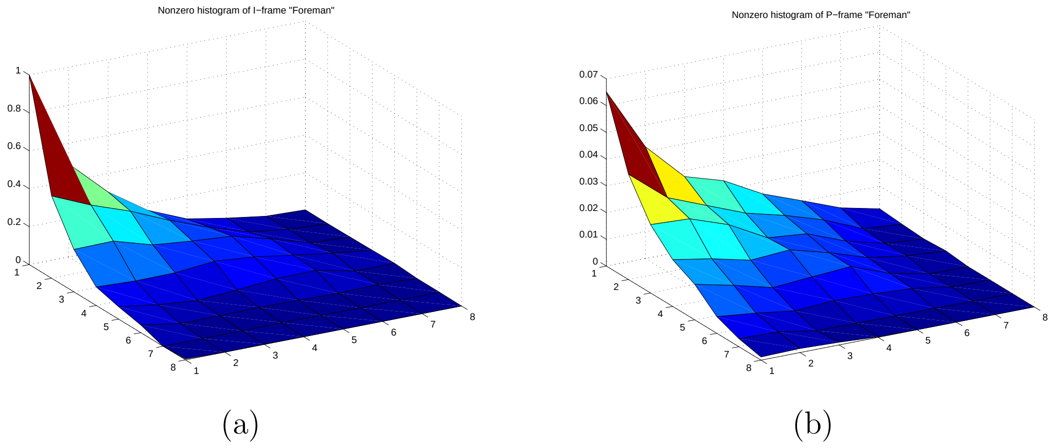


Fig. 1. Frequency of nonzero occurrence of 8x8 quantized DCT coefficients of 10 frames of “foreman” at QP=10 (a) I-frame and (b) P-frame. At higher QP, more DCT coefficients are quantized to zeros.

histograms of quantized DCT in Fig. 1 show potential complexity savings due to the sparseness of the coefficients. Computations for zero or small magnitude coefficients can be safely omitted if the locations of those coefficients are known. In the inverse DCT case it is straightforward to classify blocks (which contain transformed and quantized data) in terms of their sparseness, so that an appropriate pruned IDCT algorithm (with reduced complexity) can be used (e.g., [12], [13], [14]). The forward DCT case, however, has to address the more challenging problem of predicting the sparseness of the quantized DCT output, accurately and with minimal complexity overhead, *before* the transform and quantization are applied. As a result, before performing the DCT on a particular 8x8 block, the quantization level and the input block characteristics should be taken into account when choosing a specific reduced complexity algorithm for the DCT. The goal is to chose a reduced complexity algorithm such that *after quantization* the difference between the coefficients obtained using the exact DCT and those obtained using the approximate DCT is minimal. Clearly, the choice of algorithm (and therefore the complexity) will be input- and quantization-dependent. Two major types of VCAs for the forward DCT have been proposed in the literature, namely, those based on frequency selection and those based on accuracy selection.

A. VCA-DCT based on Frequency Selection

In these approaches, for each block only a subset of the DCT coefficients is computed. The specific subset that is chosen to be computed depends on the characteristics of the block. For example, smooth blocks can be approximated by only a few coefficients from low frequency regions. Xie and Zhu [15] propose a block-wise classification scheme in which for each input block it is determined whether it will result in an all-zero output block. The classification is done by comparing the sum of absolute value of the input block with a threshold which is a function of the quantization step. Even though the overhead due to the classification cost is minimal, this work has the limitation of being able to identify only all-zero DCT blocks. Experimental results show that, for typical images, most of the energy is in the low frequencies as can be seen in Fig. 1. Thus, filtering out the high frequency coefficients ("zonal filtering") tends to result in acceptable reconstructed images at low bit rates. For example, one could approximate the 8x8 DCT block with only DC ($X(0,0)$) and the first two AC components ($X(1,0)$,$X(0,1)$) as in [16] or the lowest 4 coefficients of a size-8 DCT as in [17], [18]. Pao and Sun [19] propose the statistical sum of absolute value testing (SSAVT) algorithm to classify each input block into one of several classes based on a first-order Markov model. For each class only a subset of coefficients is computed. This algorithm performs very well in terms of complexity savings at low bit rates. However, it is not as efficient at high bit rates because then the majority of input blocks require a computation of the full DCT and no complexity savings are achieved.

B. VCA-DCT based on accuracy selection

In these approaches, the complexity reduction is achieved by using a simplified approximation to the DCT computation; all DCT coefficients (not just a subset of coefficients) are computed but with less accuracy ("reduced accuracy"). An example of this approach can be found in distributed arithmetic techniques, where the DCT coefficients can be represented as a sum of the output of each input bit-plane [20]. Since the contribution of the last few significant bit-planes of the input is small, they can be excluded for complexity reduction without much degradation to the output. Recent work by Docef *et al* [21] proposed a multiplierless quantizer dependent approximate DCT based on an arithmetic decomposition and early termination for all-zero blocks. Another example is our previous work [1] in which we propose a multiplication-free approximate DCT algorithm. The level of approximation is made dependent on the quantization to maintain a reasonable error. These algorithms perform very well at high bit rates, where a DCT block is approximated with low complexity and small approximation error. However, at low bit rates, the complexity savings are not competitive with those achievable with frequency selection techniques, such as SSAVT, since all the DCT coefficients are computed, but the high frequency coefficients are very likely to be quantized to zero.

C. Contributions

In this paper, we start by deriving models for the error introduced by two specific VCA-DCT techniques, one based on frequency selection (SSAVT [19]) and the other on accuracy selection (Approx-Q [1]). Consider an input pixel block on which one performs the forward DCT followed by quantization. Our goal is to estimate the additional distortion in the decoded block due to using an approximate DCT instead of the exact one. Our models provide estimates of the additional distortion for each specific approximate DCT, as a function of the block variance and the quantization stepsize. This then makes it possible to select for each block the approximate DCT that best meets specific additional distortion targets. Note that, in contrast, Pao and Sun [19] selected the approximation level (i.e., the subset of DCT coefficients being computed) based on the probability that the coefficients are quantized to zero. Also, in our previous work [1], the selection of the specific approximate DCT is based on a simple QP-dependent rule, which does not take into account the approximation error. Thus, with the analysis we propose here, we can also introduce modified versions of these two algorithms (Modified-SSAVT and Approx-D) where the level of approximation is chosen to meet a target approximation error.

Further, we propose a hybrid algorithm, which we call “approximation error thresholding algorithm”, **AET**, that combines Modified-SSAVT and Approx-D in order to achieve more complexity reduction in a wider range of rate-distortion operating points. Essentially AET uses the abovementioned distortion models to decide whether to use Modified-SSAVT, Approx-D or a combination of the two algorithms. AET will then be shown to achieve improved performance at both high rates (where Approx-D would tend to be better than

Modified-SSAVT) and at low rates (where Modified-SSAVT is better.) Our experimental results show that the AET technique can achieve a speedup of at least a factor 3 (in the DCT and quantization process), with less than 0.2 dB degradation for bit rates ranging from 15 Kbps to 50 Kbps.

An important feature of the AET algorithm is that the complexity can be controlled by adjusting the level of accuracy of the transform, thus resulting in various levels of coding performance. This controllable complexity characteristic is appealing in encoders running on resource limited embedded devices. The remaining battery life and the number of applications running concurrently are time-varying factors that affect these encoding applications. By trading off the coding performance with complexity savings better overall power managements becomes possible, e.g., in low-battery situations reduced complexity DCT could be used.

Reviews of the SSAVT and Approx-Q algorithms are provided in Section II, in which the source modeling and basic concepts are introduced. The approximation error analysis is given in Section III. Based on the error analysis, the design of the proposed AET algorithm is presented in Section IV. The experimental results are shown in Section V. Finally, the conclusion is discussed in Section VI.

II. A Review of Approximate DCT Algorithms

A. Laplacian Model for Rate Distortion

In order to analyze the performance of the approximate DCT algorithms considered in this paper, we need a pixel-level model for both natural images and motion-compensated residual frames. Similar to [22], we assume that a DCT coefficient in a 2-D block is an independent random variable with Laplacian distribution, i.e., the p.d.f. of $\mathbf{X}(u,v)$ can be written as $f_{\mathbf{X}(u,v)}(x) = \frac{\lambda_{(u,v)}}{2} e^{\lambda_{(u,v)}|x|}$, where $\lambda_{(u,v)}$ is the Laplacian parameter of $\mathbf{X}(u,v)$, the DCT coefficient in position (u,v). This model, with appropriate choices for $\lambda(u,v)$, can be applied to both original images and motion-compensated residuals.

In a variable complexity algorithm, increasing complexity savings are possible as the number of zero quantized DCT coefficient increases. Given the quantization matrix, $q(u,v)$, and quantization parameter, QP, assigned to $\mathbf{X}(u,v)$, the quantizer dead-zone is in the range $[-QP \cdot q(u,v), QP \cdot q(u,v)]$. Therefore, from the Laplacian model the probability of $\mathbf{X}(u,v)$ being quantized to zero can be written as

$$p_z(u,v) = \Pr\{|\mathbf{X}(u,v)| < QP \cdot q(u,v)\} = 2(1 - e^{-\lambda_{(u,v)} QP \cdot q(u,v)}) \tag{2}$$

Furthermore, in the case of residue frames, the model parameter $\lambda_{(u,v)}$ can be obtained directly from the spatial domain. In [19], it has been observed that the correlation between pixels in residue frames can be expressed[1] as $r(m,n) = \sigma^2 \rho^{|m|} \rho^{|n|}$, where m and n are horizontal and vertical displacements, ρ is the one-dimensional correlation coefficient, and σ^2 is the pixel variance.[2] Let the correlation matrix be denoted by $\mathbf{R}$ and written as

$$\mathbf{R} = \begin{bmatrix} 1 & \rho & \rho^2 & & \rho^{N-1} \\ \rho & 1 & \rho & \cdots & \rho^{N-2} \\ \rho^2 & \rho & 1 & & \rho^{N-3} \\ & \vdots & & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & & \cdots & 1 \end{bmatrix}. \tag{3}$$

Therefore, from [23], the variance of the DCT coefficients can be derived as

$$[\sigma^2_{\mathbf{X}(u,v)}] = \sigma^2 [\mathbf{D_N R D_N^t}]_{(u,u)} [\mathbf{D_N R D_N^t}]_{(v,v)} = \sigma^2 [\mathbf{\Gamma_N}(u,v)] \tag{4}$$

where $\mathbf{D_N}$ is again the DCT matrix of size N, and the scaling factor $\mathbf{\Gamma_N}(u,v)$ is defined as a short notation for the multiplication result of the 2 brackets. Therefore, from the relationship $\lambda_{(u,v)} = \sqrt{2}/\sigma_{X(u,v)}$, we can write the probability as

$$p_z(u,v) = 2(1 - e^{-\frac{\sqrt{2}QP \cdot q(u,v)}{\sqrt{\mathbf{\Gamma_N}(u,v)}\sigma}}).$$

B. Statistical Sum of Absolute Value Thresholding

In [19], the key to complexity reduction comes from the fact that, based on the model above, if the step-size is equal to 3σ there is a 99% chance that the coefficient will be quantized to zero, and thus we can skip the computation without significantly affecting the final quality. For each coefficient, the testing would then consist of checking if

$$3\sigma_{X(u,v)} = 3\sigma\sqrt{\Gamma_N(u,v)} < QP \cdot q(u,v). \tag{5}$$

[1] For simplicity, we also apply this correlation model to pixels of INTRA frames.

[2] From our observation on five H.263 test sequences ("Miss America", "Suzie", "Mother&Daughter", "Foreman" and "Salesman"), the average ρ ranges from 0.9 to 0.97.

From (4), we can find the variance of a DCT coefficient as a scaled version of the spatial-domain variance. From the assumption of the distribution of the spatial domain signal, the variance in spatial-domain can be computed from the Sum of Absolute Value (SAV) as

$$\sigma \approx \sqrt{2} \cdot SAV/N^2 \tag{6}$$

where $SAV = \sum_{(i,j)\in Blk} |x(i,j)|$, and N^2 is the number of pixels in an NxN block. In the case of a residual frame, the SAV can be obtained as a by-product of the motion estimation in the form of the Sum of Absolute Difference (SAD) which is computed and compared in order to find the best motion vector. Therefore, the test in (5) can be rewritten as

$$SAV < (QP \cdot q(u,v) \cdot N^2)/(3\sqrt{2\Gamma_N(u,v)}) \tag{7}$$

From (4), one can find that the variances decrease from the DC to the higher frequency AC coefficients. This implies that we do not have to perform the test for every DCT coefficient. If testing proceeds from low to high frequencies, as soon as we encounter a coefficient that is deemed likely to be quantized to zero (based on our model), we know that all higher frequency (and thus lower variance) coefficients will also be within the treshold, and will be at least as likely to be quantized to zero. As a result, classification can be done by testing the SAV with a set of thresholds which corresponds to classifying the output 8x8 DCT block to *i) all-zero, ii) DC-only, iii)*[3] *low-2*x*2, iv) low-4*x*4, and v) full-DCT.* For each of the tests, $\Gamma_8(0,0), \Gamma_8(1,0), \Gamma_8(2,0)$, and $\Gamma_8(4,0)$ are used in (7), respectively. These values come from the largest $\Gamma_8(u,v)$ among the DCT coefficients outside the class of interest.

It has been shown in [19] that this method achieves significant complexity reduction due to the sparseness of the DCT coefficients for low to medium bit rate coding. At high bit rate (low QP), the threshold is smaller resulting in the more frequent occurrence of the full-DCT class. In Section III, we will provide an analysis of the distortion introduced by SSAVT.

[3] In the original paper [19], this class is not used.

C. Quantizer Dependent Approximate DCT

Now we review the Approx-Q algorithm we proposed in [1], in which rational multiplications are approximated with additions and binary shifts. In [1], five levels of approximations are used. The general structure of the proposed approximate DCT is shown in Fig. 2. This structure is modified from the structure of the fast algorithm in [5]. For exact DCT, the matrix $\mathbf{P}$ contains non-rational multiplication factors. We have proposed [1] that one can approximate the multiplications with binary shifts and additions. We can produce several algorithms with different levels of approximation by replacing the matrix $\mathbf{P}$, with one of several approximate matrices, denoted as $\mathbf{P}_j$ for $j = 1, ..., J$, where J is the number of levels of approximation. The equivalent transformation matrix using $\mathbf{P}_j$ is denoted by $\hat{\mathbf{D}}_j$. Two examples of the resulting transformation matrix are shown below for the coarsest ($\hat{\mathbf{D}}_1$) and the finest ($\hat{\mathbf{D}}_5$) approximation levels. As can be clearly seen, computing $\hat{\mathbf{D}}_1$ will be significantly faster than computing $\hat{\mathbf{D}}_5$, but will result in a worse approximation to the result produced by the exact DCT.

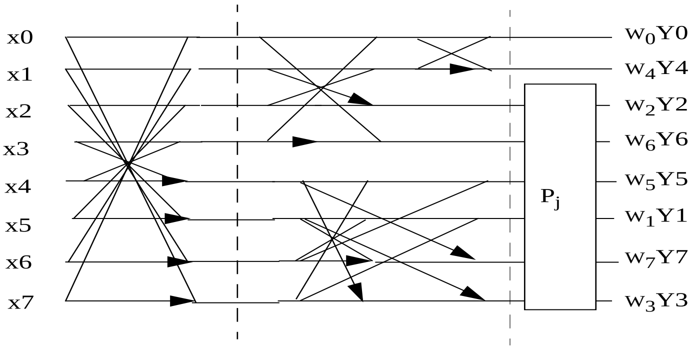


Fig. 2. Approximate DCT algorithm where the matrix $\mathbf{P}_j$, $j = 1...J$, can be one of several approximations, with J being the number of approximations, $\mathbf{P}_1$ being the coarsest approximation, and $\mathbf{P}_J$ the finest. The approximate DCT output is $\hat{X}_i = w_i Y_i$ for $i = 0...7$ where w_i is a scaling factor which can be incorporated into the quantization. The arrow lines represent multiplication by -1 before addition

$$\hat{D}_1 \;=\; \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 & -0.5 & 0.5 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0.5 & -1 & 1 & -0.5 & -0.5 & 1 & -1 & 0.5 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1.0 \\ 1.1162 \\ 1.2617 \\ 1.1162 \\ 1.0 \\ 1.1162 \\ 1.2617 \\ 1.1162 \end{bmatrix}$$

$$\hat{D}_5 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1.25 & 1.0625 & 0.6875 & 0.1875 & -0.1875 & -0.6875 & -1.0625 & -1.25 \\ 1 & 0.3750 & -0.3750 & -1 & -1 & -0.3750 & 0.3750 & 1 \\ 1.0625 & -0.1875 & -1.25 & -0.6875 & 0.6875 & 1.25 & 0.1875 & -1.0625 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0.6875 & -1.25 & 0.1875 & 1.0625 & -1.0625 & -0.1875 & 1.25 & -0.6875 \\ 0.3750 & -1 & 1 & -0.3750 & -0.3750 & 1 & -1 & 0.3750 \\ 0.1875 & -0.6875 & 1.0625 & -1.25 & 1.25 & -1.0625 & 0.6875 & -0.1875 \end{bmatrix} * \begin{bmatrix} 1.0 \\ 1.1196 \\ 1.3234 \\ 1.1196 \\ 1.0 \\ 1.1196 \\ 1.3234 \\ 1.1196 \end{bmatrix}$$

where the $*$ represents a scalar multiplication to every entry in the same row (w_i). Note that all elements in the first matrices can be implemented with only additions and binary shift operators. The scalar multiplications can be coupled with the quantization. In the Approx-Q algorithm, the level of approximation varies depending on the QP value. For a large QP, a coarse approximation is used because the quantization noise will be large and will mask out the error introduced by the approximate DCT. Conversely, finer approximation is used when QP is small. For example, if the rate control algorithm assigns QP less than 10 to a macroblock, $\hat{D}_5$ will be used as a transformation for all blocks in that macroblock whereas for QP greater than 20, $\hat{D}_1$ will be used instead. In general, five different ranges of the QP are mapped to a corresponding approximate DCT. From our preliminary observations, the Approx-Q approach is not as fast as the SSAVT approach at low bit rates because SSAVT will compute fewer coefficients according to the SAV threshold testing. However, at high bit rates, SSAVT tends to compute many full-DCT blocks, and therefore the Approx-Q approach outperforms SSAVT thanks to the fast approximation[4]. In [1], the selection of the approximation level is empirically designed. In the next section, we propose a systematic selection of the approximate DCT algorithm based on the error analysis; this error analysis can be applied to improve the algorithm selection in both SSAVT and Approx-Q.

III. Error Analysis

In this section, we will model the approximation error that each of the subsets of SSAVT and each of the matrices in Approx-Q introduce, as compared to the exact DCT algorithm, so as to have a complete knowledge of the characteristics of these algorithms in a rate, distortion, and complexity sense. This will then be used to enable real-time selection among these various algorithms. From the Laplacian model presented in Section II, we

[4] As will be seen in the experimental results in SectionV.

can compute the rate and distortion characteristics for uniform quantization given midpoint reconstruction in each quantization bin. In this paper, we assume that uniform quantization with step-size $2QP$ is used for all coefficients. The probability that the DCT coefficients are in bin $[2QPi, 2QP(i+1)]$ can be expressed as $p_i = \int_{2QPi}^{2QP(i+1)} f_{X(u,v)}(x)dx$. Therefore, the coefficient distortion $D(u,v)$ and block distortion D_{blk} can also be derived as

$$\begin{aligned} D(u,v) &= \sum_{i \neq -1,0} \int_{2QPi}^{2QP(i+1)} (x - QP(2i+1))^2 f_{X(u,v)}(x)dx + \int_{-2QP}^{2QP} x^2 f_{X(u,v)}(x)dx \\ &= \sigma^2 \Gamma_N(u,v) - \frac{2QPe^{-2\lambda QP}(3 - e^{-2\lambda QP})}{\lambda(1 - e^{-2\lambda QP})} - 3e^{-2\lambda QP} QP^2 \end{aligned} \tag{8}$$

where $\lambda = \frac{\sqrt(2)}{\sigma\sqrt{\Gamma_N(u,v)}}$.

A. SSAVT

We now analyze the distortion introduced by the SSAVT approach. For each outcome of the SAV test, a corresponding reduced output DCT is applied. We can consider the reduced output DCT as an approximation of the exact DCT. For example, the equivalent transform matrix of the low-4x4 DCT is $\begin{bmatrix} \mathbf{I}_4 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{D}_8$ where $\mathbf{I}_4$ is the 4x4 identity matrix. As in equation (7) the threshold for SAV testing can be expressed as a function of QP. Let $\{T_0, T_1, ..., T_{G-1}\}$ be a set of thresholds classifying the input into G classes. The n-th reduced output DCT (B_n, where $B_0 = \emptyset$, $B_1 = \{X_{(0,0)}\}$, $B_2 = \{X_{(i,j)} | i,j = \{0,1\}\}$, $B_3 = \{X_{(i,j)} | i,j = \{0,1,2,3\}\}$ and $B_4 = \{X_{(i,j)} | \{i,j = 0,...,7\}$) is computed if $T_n \leq SAV < T_{n+1}$ ($T_G = \infty$). From (7) and the assumption that $q(u,v) = 2$ (as in H.263), we then have

$$T_n = \frac{2QP \cdot N^2}{3\sqrt{2}\sqrt{\max\limits_{(u,v) \notin B_n} \Gamma_N(u,v)}} \tag{9}$$

for $0 \leq n \leq G = 4$. Therefore, the block distortion of a class n input can be expressed as

$$D_{ssavt}(B_n) = \sum_{(u,v) \in B_n} D(u,v) + \sum_{(u,v) \notin B_n} \sigma^2 \Gamma_N(u,v) \tag{10}$$

where $D(u,v)$ are from (8).

The first term on the right side of (10) is the sum of the distortion of coefficients that are computed while the second term corresponds to the coefficients that are not computed nor coded. Let us introduce the normalized additional distortion, which we define as the ratio between the additional distortion and the distortion due to quantization. We denote this Δ_{ssavt}, which can be written as

$$\Delta_{ssavt} = \frac{\sum_{(u,v)\notin B_n} (\sigma^2 \Gamma_N(u,v) - D(u,v))}{\sum_{\forall(u,v)} D(u,v)} \tag{11}$$

when $T_n \leq SAV < T_{n+1}$.

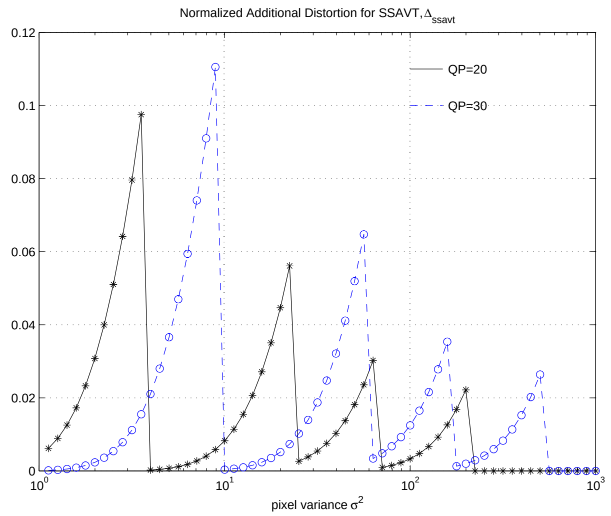


Fig. 3. Normalized additional distortion, Δ_{ssavt}, when using SSAVT at various levels of the pixel variance σ^2 assuming that the variance can be determined accurately from the SAV.

Fig. 3 shows Δ_{ssavt} as a function of σ^2 for a fixed QP. We assume that the variance of the signal is known, and therefore the distortion can be obtained directly from (10). It can be seen that the increase in distortion has a zigzag shape as a function of the pixel variance. This can be explained as follows. For each spike, the input is classified to a certain class in which a subset of the coefficients is computed. As the variance increases, the additional distortion also increases. Once the variance exceeds a threshold, the input is classified to another class which computes more DCT coefficients thus pushing the distortion down and creating the zigzag contour of Fig.3.

B. Approx-Q

First consider Fig. 4, which shows the rate-distortion (RD) performance on real data using each of the five different approximate DCT algorithms (one for each approximation to the matrix **P**) and the Approx-Q (which switches between those matrices based on the quantization values). These results are compared to the RD values obtained for the exact DCT. One can see that the performance gap between the approximate DCTs and the original DCT grows larger as the bit rate increases. However, at low bit rates all approximation levels perform equally well. Therefore, the Approx-Q algorithm in [1] shows that by adjusting the approximation accuracy as a function of QP (somewhat related to the bit rate since the rate control is done through QP assignment) one can achieve performance close to that obtained with the exact DCT. We can observe that the accuracy is increased as the QP becomes finer, so that a small degradation can be maintained over a wider range of bit rates.

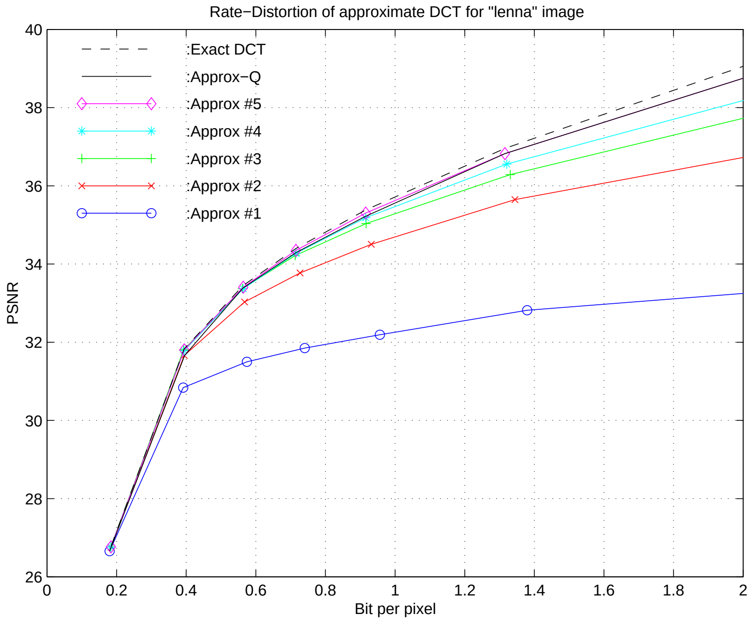


Fig. 4. Rate-Distortion curve of 512x512 lenna image JPEG coding using approximate DCT algorithms. Note that at high bit rate coarser approximate algorithm performances deviate from the exact DCT performance dramatically. Approx-Q can maintain the constant degradation level over wider range of bit rate.

However, as will be seen later, degradation is also dependent on the content. Therefore, our final goal is to select the level of approximation to ensure that the resulting additional distortion does not exceed a certain level not only for a given QP, but also for the σ^2 that characterizes each block. In order to achieve this goal, an approximation error analysis is needed. We can now use techniques similar to those used in the above SSAVT error analysis. Let us denote the transform matrix of the j-th approximate DCT by $\hat{\mathbf{D}}_j$ where $j = 1, 2, .., 5$ (number of approximation levels). Let the input spatial domain block be $\mathbf{x}$, and the DCT computed by this reduced matrix be denoted $\hat{\mathbf{X}}_j = \hat{\mathbf{D}}_\mathbf{j}\mathbf{x}\hat{\mathbf{D}}_\mathbf{j}^\mathbf{t}$. Therefore, the approximation error can be expressed as the difference between the exact and approximate output.

$$\begin{aligned} \mathbf{e_j} &= \hat{\mathbf{D}}_\mathbf{j}\mathbf{x}\hat{\mathbf{D}}_\mathbf{j}^\mathbf{t} - \mathbf{D}\mathbf{x}\mathbf{D}^\mathbf{t} \\ \bar{\mathbf{e}}'_\mathbf{j} &= ((\mathbf{D} \otimes \mathbf{D}^\mathbf{t}) - (\hat{\mathbf{D}}_\mathbf{j} \otimes \hat{\mathbf{D}}_\mathbf{j}^\mathbf{t}))\bar{\mathbf{x}}' \end{aligned} \tag{12}$$

where $\otimes$ is the Kronecker tensor product, $\bar{\mathbf{x}}'$ and $\bar{\mathbf{e}}'_\mathbf{j}$ are size N^2 vectors obtained from raster scanning (row-then-column) the input block $\mathbf{x}$ and error block $\mathbf{e}$. Let $\hat{\mathbf{E}}_\mathbf{j} = ((\mathbf{D} \otimes \mathbf{D}^\mathbf{t}) - (\hat{\mathbf{D}}_\mathbf{j} \otimes \hat{\mathbf{D}}_\mathbf{j}^\mathbf{t}))$, then the covariance matrix of the approximation error can be written as

$$\begin{aligned} E\{\bar{\mathbf{e}}'_\mathbf{j}\bar{\mathbf{e}}'^\mathbf{t}_\mathbf{j}\} &= \hat{\mathbf{E}}_\mathbf{j} E\{\bar{\mathbf{x}}'\bar{\mathbf{x}}'^\mathbf{t}\}\hat{\mathbf{E}}_\mathbf{j}^\mathbf{t} \\ &= \sigma^2 \hat{\mathbf{E}}_\mathbf{j}(\mathbf{R} \otimes \mathbf{R})\hat{\mathbf{E}}_\mathbf{j}^\mathbf{t} \end{aligned} \tag{13}$$

where $\mathbf{R}$ is the correlation matrix (3). The variance of each DCT coefficient error can then be found on the diagonal elements of (13) as

$$\sigma_e^2(u, v) = \sigma^2[\hat{\mathbf{E}}_\mathbf{j}(\mathbf{R} \otimes \mathbf{R})\hat{\mathbf{E}}_\mathbf{j}^\mathbf{t}]_{(Nu+v,Nu+v)} \tag{14}$$

It can be seen that the variance of the approximation error is simply a scaled version of σ^2. Let us rewrite (14) as $\sigma_e^2(u, v) = \sigma^2\phi_j^2(u, v)$ where the scaling factor $\phi_j^2(u, v) = [\hat{\mathbf{E}}_\mathbf{j}(\mathbf{R} \otimes \mathbf{R})\hat{\mathbf{E}}_\mathbf{j}^\mathbf{t}]_{(Nu+v,Nu+v)}$.

At this point, we assume that the error introduced in the DCT approximation and the quantization can be modeled as additive white noise, i.e., the transformed quantized DCT, $\tilde{\mathbf{X}}$, can be written as

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{n_q} + \mathbf{e_j}$$

where $\mathbf{n_q}$ represents noise from quantization. We also assume that the quantization noise is uncorrelated to the approximation error noise. Therefore, the distortion can be expressed in terms of the sum of the original quantization distortion and the distortion due to the approximation in the transform computation

$$D_{APPROX} = E\{(\mathbf{X} - \tilde{\mathbf{X}})^2\}$$

$$\begin{aligned} &= E\{(\mathbf{n_q}+\mathbf{e_j})^2\} \\ &= \sum_{(u,v)} D(u,v) + D_e(j) \end{aligned} \tag{15}$$

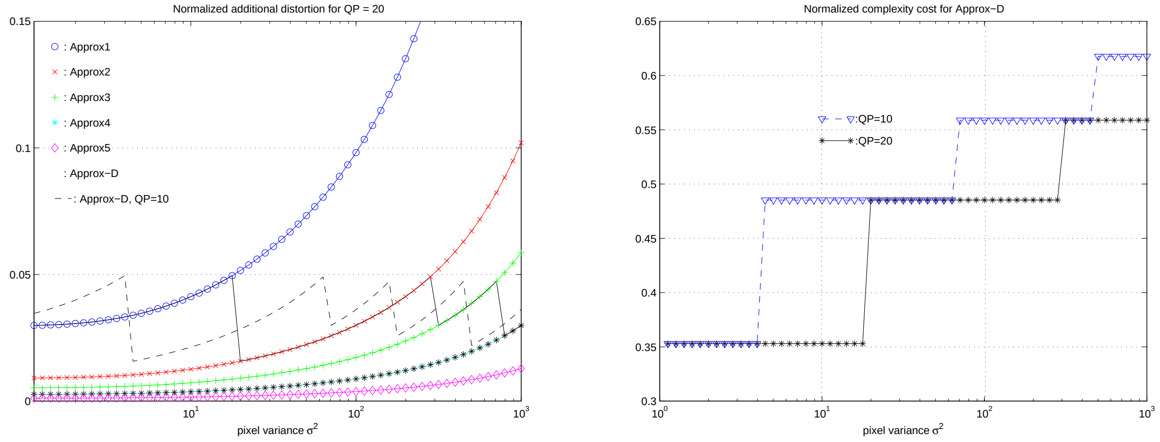
where $D(u, v)$ is from (8) and the additional distortion due to the approximation is:

$$D_e(j) = \sigma^2 \sum_{(u,v)} \phi_j^2(u,v) \tag{16}$$

is the total block approximation error. Let us denote the normalized additional distortion as

$$\Delta_{approx} = \frac{D_e(j)}{\sum_{(u,v)} D(u,v)} \tag{17}$$

In general $\phi_j^2(u, v)$ is desired to be much smaller than $\Gamma_N(u, v)$ such that the effect of DCT approximation is proportionally masked out by the quantization effect.



(a) (b)

Fig. 5. (a) Normalized additional distortion at QP = 20 using approximate DCT algorithms #1 (’o’), #2 (’x’), #3 (’+’), #4 (’$*$’), #5 (’$\diamond$’), and Approx-D (’□’) at various pixel variance σ^2. Dashed-line represents Approx-D at QP = 10. Approx-D changes level of approximation for the desired additional distortion at 0.05. (b) Corresponding normalized complexity of Approx-D at QP=20 and QP=10.

Fig. 5 (a) shows the normalized approximation error results of the 5 approximate DCT algorithms. It can be seen that not only the approximation error depends on the QP, but it also increases as the pixel variance σ^2 grows, i.e., for a fixed QP, the additional error ratio increases with pixel variance. To understand why this is the case, recall that for a given QP the distortion scales on slightly as the variance increases. This is because for each coefficient we use uniform quantization and there is practically no overload quantization error, even when the variance increases. There may be slight increases in overall distortion due to the fact that the number of coefficients being transmitted increases and the QPs are slightly higher (due to perceptual weighting) for the higher frequency ones. Contrast this, however, with the behavior of the error due to using different matrices. To simplify, assume that one particular coefficient α is approximated by $\alpha + \Delta\alpha$. Then, clearly, as the variance of the input signal increases, the variance of the error, which is proportional to $\Delta\alpha$ also increases. In short, the distortion due to the matrix approximation increases much faster with the input variance, than the distortion due to quantization. This explains the behavior seen in Fig. 5. Furthermore, for a fixed pixel variance, the additional error ratio decreases as quantization step-size increases. It can be seen that, for a given approximate algorithm, QP still plays a bigger role in the resulting error.

IV. Approximation Error Thresholding

Given the above analysis, we now have a tool to select which approximation to use given the desired level of additional distortion which is derived as a function of QP and σ^2 in previous section. As a result, the Approx-Q algorithm can be modified such that the level of approximation now depends on *both* the quantization and the block variance. This will enable us to guarantee that the additional distortion will remain below a desired threshold. The modified algorithm is then as follows for each block:

Algorithm 1 (Approx-D)

Step 1: *Let J be the number of approximation algorithms and let the level of accuracy be in ascending order with respect to j. The J-th algorithm is the exact-DCT. Set $j = 0$.*

Step 2: *Compute Δ_{approx} of the j-th algorithm where Δ_{approx} is defined in (17).*

Step 3: *If $\Delta_{approx} \leq \eta$ where η is the level of desired additional error, select the j-th algorithm. Otherwise, increment j.*

Step 4: *If $j = J$, stop. Otherwise, go back to Step 2.*

For example, when coding a frame with fixed QP for all blocks, low variance blocks (associated with low activity) require less accurate DCT approximation whereas high variance blocks must use finer approximation in order to maintain the same level of additional error throughout the entire frame. Shown in Fig. 5 (a) are the Approx-D results with

$\eta = 0.05$ or 5% of the quantization error. In Fig. 5 (b), the normalized complexity – compared to the full-DCT complexity[5]– of the Approx-D algorithm is shown. We can see that, for a given QP, the Approx-D algorithm switches from one level of approximation to another as variance increases, according to the approximation error in (16). Due to the multiplication-free property of the Approx-D algorithm, the resulting complexity saving can be significant, i.e., between 40% to 65%.

We can apply the same principle to SSAVT, so that the SAV is used to compute the additional approximation error, Δ_{ssavt}, and the subset of coefficients is chosen so as to meet a desired level of accuracy, η. The modified SSAVT algorithm performs the following operations for each block:

Algorithm 2 (Modified SSAVT)

Step 1: *Let N be the number of SSAVT algorithms in which the set of computed DCT coefficients grows with respect to n. The N-th algorithm is the full-DCT. Set $n = 0$.*

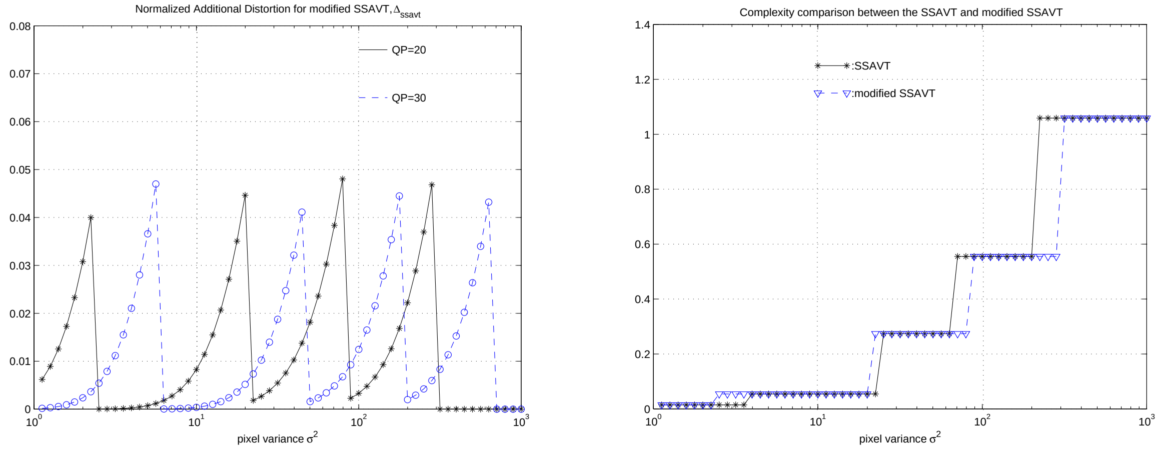
Step 2: *Compute Δ_{ssavt} of the n-th algorithm where Δ_{ssavt} is defined in (11).*

Step 3: *If $\Delta_{ssavt} \leq \eta$ where η is the level of desired additional error, select the n-th algorithm. Otherwise, increment n.*

Step 4: *If $n = N$, stops. Otherwise, go back to Step 2.*

Fig. 6 (a) shows the result of modified SSAVT. As compared to Fig. 3, it can be seen that the normalized additional distortion is kept under 0.05 by switching to a finer approximation algorithm, i.e., in this case, more DCT coefficients are computed. Eventually, as σ^2 increases, the additional distortion becomes zero after the full-DCT is used, and the complexity approaches that of the baseline algorithm as the variance increases. Fig. 6 (b) shows the normalized complexity of SSAVT and modified SSAVT. We can see that modified SSAVT switches to finer approximation earlier than SSAVT for low variance blokcs, but it switches later than the original SSAVT for high variance blocks. This is beneficial in a complexity-distortion sense since the complexity difference between the algorithms providing coarser approximation is much smaller than that between the algorithms providing finer approximation. For example, the absolute increase in number of coefficients

[5] The complexity is estimated by a weighted sum of arithmetic operations involved in the algorithms. In this paper, we use 3 for a multiplication, and 1 for an addition or a binary shift as it is considered to be a good approximation by [24].



(a) (b)

Fig. 6. (a) Additional distortion from approximation error Δ_{ssavt} for modified SSAVT at QP $= 20$ and 30 as a function of σ^2. As the approximation error grows larger than the specified threshold, $\eta = 0.05$, the algorithm switch to finer approximation, more coefficients computed. (b) Normalized complexity cost comparison between SSAVT and modified SSAVT.

computed is 3 when going from 1x1 to 2x2, but it becomes 48 when going from 4x4 to 8x8.

Comparing Fig. 5 (b) and Fig. 6 (b), it can be seen that for a given QP, in the high variance region the modified SSAVT requires higher complexity than Approx-D, since more coefficients are computed for SSAVT, whereas a fine level approximation, which provides a good compromise between accuracy and speedup, is used by Approx-D. On the other hand, in the low variance region, SSAVT is faster than Approx-D since most of the coefficients are to be quantized to zeros. Note that this comparison is performed under a desired level of approximation error constraint, $\eta = 0.05$.

This leads to our proposed approximation error thresholding (AET) algorithm, which combines the strengths of the modified SSAVT and the Approx-D algorithms. The idea behind this approach is illustrated as follows. In the low variance region where it is relatively easier to achieve speedup with acceptable accuracy, the AET employs the zonal filtering approach (SSAVT) which takes advantage of the sparseness of the coefficients. In high variance areas, the resulting class from SSAVT tends to be the same as full-DCT, the AET still achieves complexity savings coming from the Approx-D part.

Furthermore, we apply the reduced-accuracy scheme on top of the zonal-filtering approach by approximating the subset of DCT coefficients instead of computing the exact values of the full set of the DCT coefficient. In this case, the distortion can be modeled as a sum of quantization noise, zonal filtering noise, and approximation noise. Again, if we assume that these noises are uncorrelated the distortion can be expressed as

$$D_{AET}(B_n, j) = \sum_{(u,v)\in B_n} [D(u,v) + \sigma^2 \phi_j^2(u,v)] + \sum_{(u,v)\notin B_n} \sigma^2 \Gamma_N(u,v) \qquad (18)$$

for the $(n,j)-th$ approximation level, where the pair (n,j) denotes the combination class B_n in SSAVT and the j-th class in Approx-D. The first summation on the right side is the distortion plus approximation error of the approximated coefficients. The second term is for non-computed coefficients from zonal filtering. Let Δ_{aet} denote the normalized additional distortion for the AET which can be expressed as

$$\Delta_{aet} = \frac{\sum_{(u,v)\in B_n} \sigma^2 \phi_j^2(u,v) + \sum_{(u,v)\notin B_n} (\sigma^2 \Gamma_N(u,v) - D(u,v))}{\sum_{(u,v)} D(u,v)} \qquad (19)$$

In order to determine which level of approximation is to be used, Δ_{aet} is compared with the target error threshold, η.

The AET algorithm thus can be stated as follows.

Algorithm 3 (Approximation Error Thresholding (AET) Algorithm)

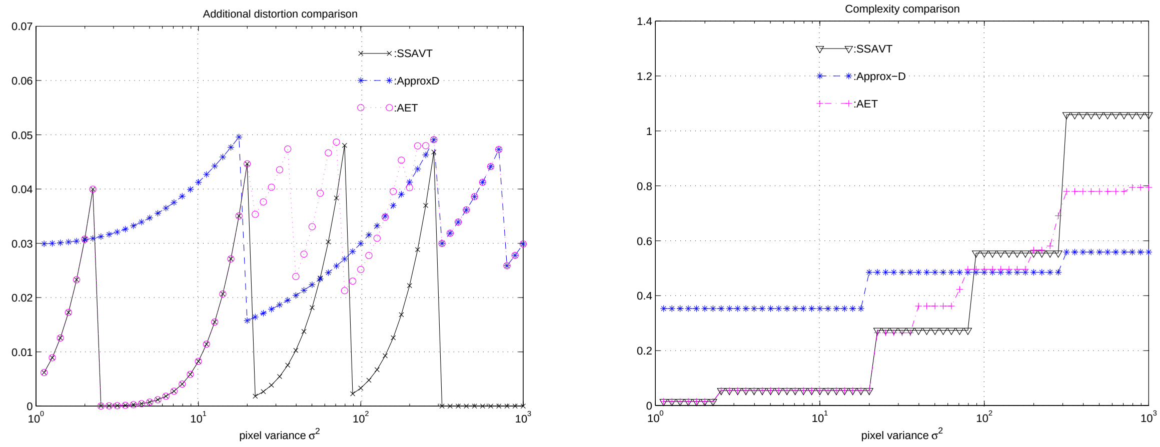
Step 1: *Let* (n,j) *represents an algorithm computing the* n*-th subset of the DCT coefficients with the* j*-th level of accuracy.* n *ranges from 0 to* N *and* j *ranges from 0 to* J*.* (N,J)*-th algorithm is the full-DCT. Set* $(n,j) = (0,0)$

Step 2: *Compute* Δ_{aet} *of the* (n,j)*-th algorithm where* Δ_{aet} *is defined in (19).*

Step 3: *If* $\Delta_{aet} \leq \eta$ *where* η *is the level of desired additional error, select the* (n,j)*-th algorithm. Otherwise, increment* j*.*

Step 4: *If* $j = J$*, set* $j = 0$*, increment* n*, and go back to Step 2. Otherwise, if* $(n,j) = (N,J)$*, stop.*

Fig. 7 (a)&(b) show the approximation error according to (19) and the normalized complexity of the AET algorithm compared to modified SSAVT and Approx-D, respectively. As expected, it can be seen that at low variance, the AET selects the SSAVT approach over Approx-D. As the variance increases, the AET goes through several transitions among the approximation levels in the same zonal filtering level before moving on to larger frequency zone. Eventually, at high variance, all the coefficients are approximated following the Approx-D approach. In the mid variance range, the AET complexity curve deviates from that of SSAVT and converges to the Approx-D complexity curve. However, as can be seen in Fig. 7 (b), there is a complexity discrepancy between the AET and the Approx-D at high bit rates due to classification overhead, i.e., there are multiple approximation levels for `2x2`, `4x4` and `8x8` DCT, but only one choice to perform all-zero and DC-only DCT. Note that the DC-only class can be computed exactly since the DC value is only a scaled sum of pixels in the 8x8 block.



(a) (b)

Fig. 7. (a) Approximation errors as a function of σ^2 of AET, modified SSAVT and Approx-D at QP = 20 for a given approximation error threshold, $\eta = 0.05$. (b) Normalized complexity cost comparison between the AET, modified SSAVT and Approx-D.

V. Experimental Results

In this section, practical implementation issues are discussed. We will demonstrate that input classification cost is low enough that the computation overhead is almost negligible compared to the complexity saving gains. According to (19), the value inside the bracket can be pre-computed for all possible (n, j) pairs. A look-up table can be used to access these pre-computed value during the classification phase. From the Laplacian model, σ can also be obtained from the SAD as a by product from the motion estimation module for INTER frames. For the denominator of (19) which refers to the distortion from quan-

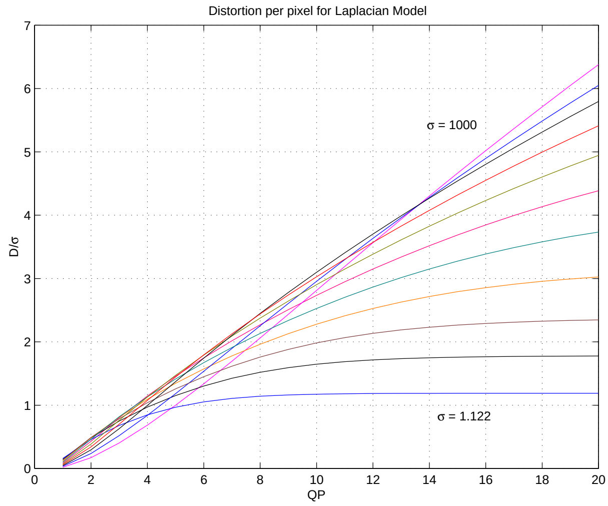


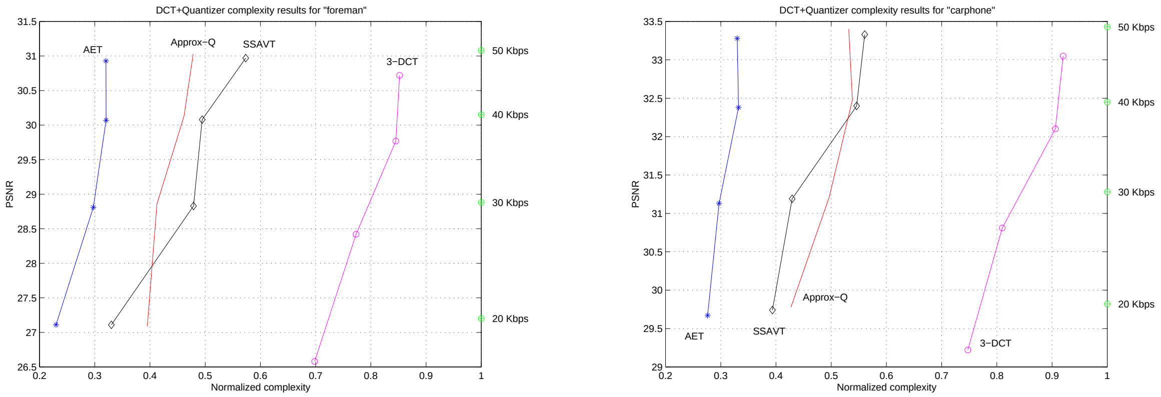
Fig. 8. Distortion from the Laplacian model for $\rho = 0.9$ where x-axis is QP and y-axis is the ratio of distortion (per pixel) and σ. Each curve represents different σ values. Note that one can approximate the distortion function by the first order polynomial model.

tization, the model of distortion of (8) is used in our experiment. However, our goal is to estimate the distortion before the actual encoding operation, therefore, to avoid heavy computation, a first-order model is used to approximate (8) as follows,

$$\frac{\hat{D}}{\sigma} = \min(k \cdot Q + c\ ,\ \sigma) \tag{20}$$

where k and c are model parameters which can be obtained from linear regression of the previous frame. Fig. 8 shows the average pixel distortion computed from (8). One can see that the ratio between the distortion and σ can be approximated by a linear function which saturates at high QP approaching the source standard deviation (i.e., distortion converges toward the source variance).

For simplicity, in our implementation we introduce a modification to the AET algorithm. Our experiments show that AET does not exhaust all approximation levels for 2x2-DCT and 4x4-DCT classes before moving on to the next frequency zone, as can be seen in Fig. 7, thus allowing us to limit the number of approximation options in these 2 classes. Taking into consideration the tradeoffs among classification overhead, complexity, and approximation error, we use only Approx#5 for the 2x2-DCT and Approx#4 for the 4x4-DCT class, respectively. The reason behind these choices is due to the fact that since only a few coefficients are computed, a finer approximation algorithm does not significantly



(a) (b)

Fig. 9. Complexity of DCT + Quant vs. PSNR at different target bit rates (20, 30, 40 and 50 Kbps) on 249 frames of (a) "foreman" and (b) "carphone" sequences where the complexity is normalized by the complexity of the baseline DCT ('$\bigoplus$'). Shown here are the results of SSAVT ('◇'), Approx-Q ('□'), AET ('*'), and Girod's 3-DCT algorithm ('o').

cost more than the coarser ones.

Experiments have been performed using the baseline H.263 TMN8 [14]. The original DCT module using Chen's algorithm [6] is replaced with the AET algorithm. The rate control is turned on to keep constant bit rate such that we can compare the complexity-distortion performance for fixed bit rate. The Visual C++ compiler with optimization option is used. The C function clock() is used to measure CPU clock cycles spent. Both the DCT *and* quantization are measured because the quantization for zero coefficients can be omitted as well. The experiment was run on a Pentium 4 733 MHz machine running WindowsNT 4.0. Note that RISC-based processors have become more commonly found in embedded devices, therefore, the results from a PC can be used as a good representative of the overall performance improvement.

In Fig. 9, it can be seen that the complexity savings by SSAVT range from 40 to 60% from high rate to low rate. As the rate gets higher, there are more nonzero coefficients to be coded due to smaller quantizer thus resulting in less complexity reduction. For Approx-Q, the quantization still has to take place since all DCT coefficients are computed, though not exactly. However, the speedup from the DCT part is very significant, as can be seen in Fig. 9, where the complexity saving ranges from 40 to 65%. It is interesting to point out that at high bit rates, Approx-Q performs a little better than SSAVT since there are many nonzero coefficients whereas SSAVT performs better in the low rate region due to the fact that a lot of zero coefficients are omitted by SSAVT. Note that these complexity numbers also include the classification overhead. Thus, it verifies our previous statement that the classification cost is negligible.

	Zero	1x1	2x2	4x4	8x8/App1	App2	App3	App4	App5
50K	1494	7935	10932	16096	12251				
					378	4158	11964	27558	4650
	1330	8107	10980	15975	0	9534	910	199	1673
40K	2362	8811	11381	15343	10217				
					3552	5892	13188	24678	804
	2093	9027	11399	15433	0	7799	479	207	1677
30K	3750	10033	11743	14350	7644				
					8364	8586	17088	13284	198
	3624	10153	11917	14278	0	5566	220	383	1379
20K	5896	10939	11285	11483	4353				
					18390	9984	13674	1830	78
	5881	11016	11242	11474	0	2574	87	595	1087

TABLE I

Frequency of occurrence of different input classes of "carphone"

Table I and II show the number of input classes classified by SSAVT, Approx-Q and AET (the first, second and third line in the same bit rate category) for "carphone" and "foreman", respectively, at different bit rates. The class Zero, 1x1, 2x2, 4x4 and 8x8 are for the reduced output classes from the SSAVT algorithm. The class App1 to App5 represent 5 level of approximations used by the Approx-Q algorithm. For AET, our experiments use only one level of approximation for each of the all-zero, 1x1 (dc-only), 2x2 and 4x4 classes, and 5 levels of approximation for the 8x8 (full-DCT) class. Therefore, Table I and II represent all possible classes of input in out experiments.

It can be seen that, in general, as the bit rate decreases, coarser approximations are

	Zero	1x1	2x2	4x4	8x8/App1	App2	App3	App4	App5
50K	1213	7858	9580	15798	13665				
					642	5034	26784	15654	0
	1438	8072	9082	15857	0	10807	1100	107	1651
40K	2446	8686	9974	15750	10664				
					3216	11346	29868	3090	0
	2472	8904	9599	16010	0	8283	501	107	1644
30K	3649	10030	10258	15321	7074				
					12426	21822	12084	0	0
	4158	9905	10177	15005	0	5231	148	241	1467
20K	6462	10685	10367	10722	3938				
					37002	4380	792	0	0
	6820	10150	10090	10661	0	1997	21	576	1265

TABLE II

Frequency of occurrence of different input classes of “Foreman”

selected as a result of higher QP contribution to the thresholds. The hybrid AET algorithm extends the range of rates at which we can achieve reductions in complexity as can be seen in Fig. 9, where the complexity savings range from 67 to 73%, e.g., 68% at 50 Kbps and 77% at 20 Kbps for “foreman” sequence and 67% at 50 Kbps and 72% at 20 Kbps for “carphone” sequence. In terms of quality degradation, AET performs only slightly worse than SSAVT and Approx-Q by staying within -0.15 dB from the exact DCT case. In terms of perceptual quality, there is almost unnoticeable difference between the AET and the exact DCT. We also show the result of Girod’s 3-DCT algorithm with $\theta = 40$ (see [16] for details) where the complexity reduction is between 10 to 30% while the PSNR degradation is up to 0.5 dB. Note that the complexity of Girod’s algorithm can be further reduced by increasing θ, however, the quality degradation is also expected to be higher.

VI. Conclusions

Based on the approximation error analysis using Laplacian model, we propose variants to the well-known two approaches, frequency selective SSAVT [19] and accuracy selective Approx-Q [1] such that the additional distortion is explicitly addressed. From these two new versions, we then propose a fast computationally scalable approximation error thresholding (AET) DCT algorithm which combines the advantage of SSAVT and Approx-Q in terms of the speedup vs. accuracy tradeoff in various bit rate ranges. Its performance shows up to 73% complexity reduction with only 0.2 dB PSNR degradation. In a video compression standard such as MPEG where the DCT contributes around 10-30% of the total encoding time, a DCT speedup at this magnitude can significantly affect the overall encoding speed.

VII. Acknowledgement

The authors would like to thank the reviewers for their constructive comments which were very helpful in improving the manuscript.

References

[1] K. Lengwehasatit and A. Ortega, “DCT computation based on variable complexity fast approximations,” in *Proc. of ICIP’98*, Chicago, IL, October 1998.

[2] K. Lengwehasatit, *Complexity-Distortion Tradeoffs in Image and Video Compression*, Ph.D. thesis, Electrical Engineering Dept., University of Southern California, 2000.

[3] K.R. Rao and P. Yip, *Discrete Cosine Transform, Algorithms, Advantages, Applications*, Academic Press, 1990.

[4] C. H. Smith W.-H. Chen and S. C. Fralick, “A fast computational algorithm for the discrete cosine transform,” *IEEE Trans. on Comm.*, vol. COM-25, no. 9, pp. 1004–1009, September 1977.

[5] A. Ligtenberg and M. Vetterli, “A discrete fourier/cosine transform chip,” *IEEE J. on Selected Areas in Comm.*, vol. SAC-4, pp. 49–61, January 1986.

[6] Z. Wang, “Fast algorithms for the discrete w transform and for the discrete fourier transform,” *IEEE Trans. on Signal Proc.*, vol. ASSP-32, no. 4, pp. 803–816, August 1984.

[7] P. Duhamel and H. H’Mida, “New 2n DCT algorithms suitable for VLSI implementation,” in *Proc. of ICASSP’87*, Dallas, April 1987, p. 1805.

[8] C. Loeffler, A. Ligtenberg, and G. Moschytz, “Practical fast 1-D DCT algorithms with 11 multiplications,” in *In Proc. of ICASSP’89*.

[9] M. Vetterli, “Tradeoffs in the computation of mono and multi-dimensional DCTs,” Tech. Rep., Ctr. fo Telecommunications Research, Columbia University, June 1988.

[10] H. R. Wu and Z. Man, “Comments on ”fast algorithms and implementation of 2-D discrete cosine transform,” *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, no. 2, pp. 128–129, April 1998.

[11] Y. Arai, T. Agui, and M. Nakajima, “A fast DCT-SQ scheme for images,” *Trans. of IEICE*, vol. 71, no. 11, pp. 1095, November 1988.

[12] E. Murata, M. Ikekawa, and I. Kuroda, “Fast 2D IDCT implementation with multimedia instructions for a software MPEG2 decoder,” in *Proc. of ICASSP'98.*

[13] MPEG Software Simulation Group, “MPEG2 video codec version 1.2,” .

[14] “University of British Columbia, Canada, TMN H.263+ encoder version 3.0,” .

[15] B. Xie and X. Zhu, “A global decision method for moving picture coding,” *IEEE Trans. on Consumer Electronics*, vol. 45, no. 1, pp. 84–90, February 1999.

[16] B. Girod and K. W. Stuhlmüller, “A content-dependent fast DCT for low bit-rate video coding,” in *Proc. of ICIP'98.*

[17] “The independent JPEG's group software JPEG, version 6,” `ftp://ftp.uu.net`.

[18] S.H. Jung, S.K. Mitra, and D. Mukherjee, “Subband DCT: Definition, analysis, and applications,” *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 6, no. 3, pp. 273–312, June 1996.

[19] I.-M. Pao and M.-T. Sun, “Modeling DCT coefficients for fast video encoding,” *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 9, no. 4, pp. 608–616, June 1999.

[20] Jonathan B. Lipsher, “Asynchronous DCT processor core,” M.S. thesis, Univ. of Cal. Davis.

[21] A. Docef, F. Kossentini, K. Nguuyen-Phi, and I. R. Ismaeil, “The quantized DCT and its application to dct-Based video coding,” *IEEE Trans. on Image Proc.*, vol. 11, no. 3, pp. 177–187, March 2002.

[22] M. J. Gormish, *Source Coding with Channel, Distortion and Complexity Constraints*, Ph.D. thesis, Stanford University, March 1994.

[23] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[24] D.A.Patterson and J.L.Hennessy, *Computer Architecture : a Quantitative Approach 2nd Ed.*, Morgan Kaufmann Publishers, 1996.

List of Figures

1 Frequency of nonzero occurrence of 8x8 quantized DCT coefficients of 10 frames of “foreman” at QP=10 (a) I-frame and (b) P-frame. At higher QP, more DCT coefficients are quantized to zeros. 4

2 Approximate DCT algorithm where the matrix $\mathbf{P}_j$, $j = 1...J$, can be one of several approximations, with J being the number of approximations, $\mathbf{P}_1$ being the coarsest approximation, and $\mathbf{P}_J$ the finest. The approximate DCT output is $\hat{X}_i = w_i Y_i$ for $i = 0...7$ where w_i is a scaling factor which can be incorporated into the quantization. The arrow lines represent multiplication by -1 before addition 10

3 Normalized additional distortion, Δ_{ssavt}, when using SSAVT at various levels of the pixel variance σ^2 assuming that the variance can be determined accurately from the SAV. 13

4 Rate-Distortion curve of 512x512 lenna image JPEG coding using approximate DCT algorithms. Note that at high bit rate coarser approximate algorithm performances deviate from the exact DCT performance dramatically. Approx-Q can maintain the constant degradation level over wider range of bit rate. 14

5 (a) Normalized additional distortion at QP = 20 using approximate DCT algorithms #1 ('o'), #2 ('x'), #3 ('+'), #4 ('$*$'), #5 ('$\diamond$'), and Approx-D ('$\Box$') at various pixel variance σ^2. Dashed-line represents Approx-D at QP = 10. Approx-D changes level of approximation for the desired additional distortion at 0.05. (b) Corresponding normalized complexity of Approx-D at QP=20 and QP=10. 16

6 (a) Additional distortion from approximation error Δ_{ssavt} for modified SSAVT at QP = 20 and 30 as a function of σ^2. As the approximation error grows larger than the specified threshold, $\eta = 0.05$, the algorithm switch to finer approximation, more coefficients computed. (b) Normalized complexity cost comparison between SSAVT and modified SSAVT. 19

7 (a) Approximation errors as a function of σ^2 of AET, modified SSAVT and Approx-D at QP = 20 for a given approximation error threshold, $\eta = 0.05$. (b) Normalized complexity cost comparison between the AET, modified SSAVT and Approx-D. 21

8 Distortion from the Laplacian model for $\rho = 0.9$ where x-axis is QP and y-axis is the ratio of distortion (per pixel) and σ. Each curve represents different σ values. Note that one can approximate the distortion function by the first order polynomial model. 22

9 Complexity of DCT + Quant vs. PSNR at different target bit rates (20, 30, 40 and 50 Kbps) on 249 frames of (a) “foreman” and (b) “carphone” sequences where the complexity is normalized by the complexity of the baseline DCT ('$\bigoplus$'). Shown here are the results of SSAVT ('$\diamond$'), Approx-Q ('$\Box$'), AET ('*'), and Girod's 3-DCT algorithm ('o'). 23

List of Tables

I Frequency of occurrence of different input classes of “carphone” 24

II Frequency of occurrence of different input classes of “Foreman” 25