

Source Model for Transform Video Coder and Its Application—Part I: Fundamental Theory

Hsueh-Ming Hang, *Senior Member, IEEE*, and Jiann-Jone Chen

Abstract—A source model describing the relationship between bits, distortion, and quantization step sizes of a large class of block-transform video coders is proposed. This model is initially derived from the rate-distortion theory and then modified to match the practical coders and real image data. The realistic constraints such as quantizer dead-zone and threshold coefficient selection are included in our formulation. The most attractive feature of this model is its simplicity in its final form. It enables us to predict the bits needed to encode a picture at a given distortion or to predict the quantization step size at a given bit rate. There are two aspects of our contribution: one, we extend the existing results of rate-distortion theory to the practical video coders, and two, the nonideal factors in real signals and systems are identified, and their mathematical expressions are derived from empirical data. One application of this model, as shown in the second part of this paper, is the buffer/quantizer control on a CCITT $P \times 64$ k coder with the advantage that the picture quality is nearly constant over the entire picture sequence.

Index Terms—Image coding, rate distortion theory, source coding.

I. INTRODUCTION

TRANSFORM coding is a very popular technique in image compression. It is one of the key components in the international video communication standards [1]–[3]. Often, the communication channel poses constraints on the bit rate it can accept, the video coder bit rate control or output buffer control becomes one of the critical problems in designing a video compression system. In order to predict the effect (output bit rate) due to the adjustment of coding parameters, it is very desirable to be able to construct a source model that can estimate the bits produced by a video coder for a chosen set of coding parameters.

There are two approaches in constructing such a source model: 1) the analytic approach that constructs a mathematical description of a coder by analyzing the structure and behavior of every component in the coder and 2) the empirical approach that derives the input/output relationship of a coder based on the observed data. Although the analytic model of a simple quantizer already existed for a long time [4]–[6], the complete analysis of a standard transform coder has not, to

our knowledge, been fully explored. An attempt is made by Hang *et al.* [7], but it does not contain rigorous theoretical justification. On the other hand, several studies on source modeling [8], [9] based on the empirical data have been reported. Although the empirical approach is rather useful in practice, it does not provide us with insights on the principles of video coder operations and the amount of information contained in an image sequence. In addition, because it is derived from the training data, one may worry about its robustness—its performance on the unknown data because these data may be rather different from the training data statistically.

Our approach in this paper belongs to the analytic approach category. We decompose both the coding system and image signal into components of known mathematical models and then combine them together to form a complete description. The theoretical foundation of this approach is the rate-distortion theory. A few elements in our model already exist in the literature. Our contribution is two fold. One, we combine and extend the existing results to the standard-type video coders, and two, the nonideal factors in real signals and systems are incorporated as adjustable parameters to compensate their bias effects on the ideal model. The goal is to build a general source coding model that can be used to predict the coder behavior by taking simple and basic measures of signals such as variances.

This paper is organized as follows. We first briefly review the rate distortion theory of Gaussian signals and uniform quantizers that are relevant to our coder source model. In Sections III and IV, we derive the source model by putting the known elements together with our own extensions. The parameters in our model, which are originally derived from theory, are adjusted to match real pictures and the standard coding algorithms in practical applications. The impact of the nonideal factors in picture and practical coders on our source model are discussed in Sections V-A and V-B. The numerical values of model parameters computed based on the compressed image data are described in Section V-C. Section VI summarizes the results presented in this paper.

II. RATE-DISTORTION MODEL OF QUANTIZER

In this section, the one-dimensional (1-D) discrete signal properties relevant to this paper are briefly reviewed, and then these properties are extended to the two-dimensional (2-D) signals. Extensions are often straightforward under the assumptions we made on the 2-D signals.

Manuscript received May 27, 1995; revised June 1, 1996. This paper was recommended by Associate Editor W. Li. This work was supported in part by the National Science Council of R.O.C., Taiwan, under Grant NSC82-0404-E009-176.

The authors are with the Department of Electric Engineering, National Chiao Tung University, Hsinchu, Taiwan, 300, R.O.C.

Publisher Item Identifier S 1051-8215(97)01140-3.

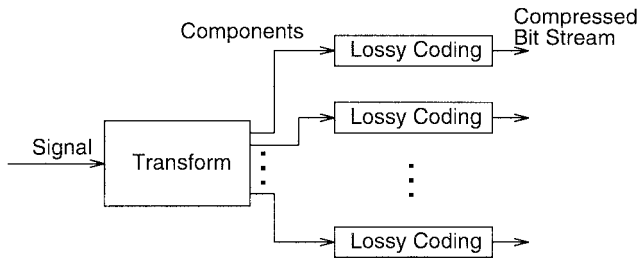


Fig. 1. Signal decomposition using transform.

A. Stationary Gaussian Process

The following results known in information theory are the bases of our future discussions. First, if a (composite) signal can be decomposed into two or more independent components, then its (total) rate-distortion function can be derived directly from the rate distortion functions of the individual components. In theory, there is no loss of compression efficiency in decomposing a composite signal into simpler components and then compressing each component independently [6].

Second, the rate distortion functions of a few *independent identically distributed* (i.i.d.) sources are known such as the i.i.d. Gaussian sequence. Based on the proposition stated in the above, the rate-distortion functions of signals with memory can be derived by decomposing these signals into independent components with known rate-distortion functions. An example is the stationary Gaussian process. We can apply Fourier transform to it and transformed frequency components are i.i.d. Gaussian sequences. This procedure also suggests a way for compression. That is, an *optimal* procedure for compressing stationary random processes is to transform a composite signal into independent components and then apply the ordinary data compression techniques to each component separately [10]. A pictorial illustration of this concept is shown in Fig. 1.

The well-known rate distortion function of a discrete stationary Gaussian process $\{x(n)\}$ under the mean square distortion criterion is given as [6], [10]

$$R_\theta = \frac{1}{4\pi} \int_{A_1} \log_2 \frac{\Phi(\omega)}{\theta} d\omega \quad (1)$$

and

$$D(R_\theta) = \frac{1}{2\pi} \int_{A_1} \theta d\omega + \frac{1}{2\pi} \int_{B_1} \Phi(\omega) d\omega \quad (2)$$

where $\theta \geq 0$ and $\Phi(\omega)$ is the power spectrum density function of $\{x(n)\}$, and

$$\begin{cases} \text{Region } A_1: & \{\omega \in (-\pi, \pi] \text{ and } \Phi(\omega) \geq \theta\} \\ \text{Region } B_1: & (-\pi, \pi] - A_1. \end{cases}$$

An interpretation of the above formula is that R_θ is the minimum bit necessary to achieve an average distortion D by an ideal coder of possibly unbounded complexity and time delay. Similarly, $D(R)$ is the minimum average distortion that

can possibly be achieved at bit rate R . In (1) and (2), θ is a dummy parameter whose value is decided by a selected D or R value. The above formulas suggest that to achieve the optimum coding performance, the frequency components of power less than θ should be discarded and an θ amount of distortion should be imposed on every one of the retained frequency components.

In reality, we cannot use infinite length transforms to decompose a signal sequence. A typical approach is to partition a signal sequence into nonoverlapped blocks and perform block transformation on each data block separately. There are two problems associated with this approach. One is the correlation among the neighboring blocks, which can be significant for the low frequency components. A part of this correlation can be reduced by applying another layer of correlation reduction techniques on the low frequency components of nearby blocks. The other problem is the power spectrum used in (1) and (2). Practically, the signal power spectrum has to be estimated from data samples. A simple and popular spectrum estimation method is the *periodogram* that computes the spectrum based on the weighted average of the Fourier transforms of nonoverlapped data blocks [11]. This method is consistent with the finite-size transform we use in data compression if we view the block transform components as the discrete approximation of the ideal continuous power spectrum. Assuming a uniform sampling grid in the frequency domain, (1) and (2) can thus be approximated by the following discrete versions:

$$R_\theta = \frac{1}{2L} \sum_{\omega_i \in A_1} \log_2 \frac{\Phi(\omega_i)}{\theta} \quad (3)$$

and

$$D(R_\theta) = \frac{1}{L} \sum_{\omega_i \in A_1} \theta + \frac{1}{L} \sum_{\omega_i \in B_1} \Phi(\omega_i) \quad (4)$$

where L is the number of samples in a data block, and $\omega_i = i \cdot 2\pi/L$, $i = 0, 1, \dots, L-1$. This is exactly the system represented by Fig. 1 with L components.

One may note that (3) can be rewritten as

$$e^{L \cdot \alpha \cdot R_\theta} = \prod_{\omega_i \in A_1} \frac{\Phi(\omega_i)}{\theta} \quad (5)$$

where $\alpha = 1.386$. An interesting property of the above equation is that the exponential function of bit rate is proportional to the *product* of the variances of the signal components rather than the sum of variances.

A case of interest is that at low distortion when $A_1 = (-\pi, \pi]$ (or B_1 is empty), (4) becomes $D = \theta$. And thus, (5) becomes

$$e^{L \cdot \alpha \cdot R(D)} = \prod_{i=0}^{L-1} \frac{\Phi(\omega_i)}{D} \quad (6)$$

or

$$D = E \cdot e^{-\alpha \cdot R(D)} \quad (7)$$

where

$$E = \left[\prod_{i=0}^{L-1} \Phi(\omega_i) \right]^{1/L} \quad (8)$$

Essentially, we are approximating a joint Gaussian source by multiple i.i.d. Gaussian sources. If the approximation errors can be viewed as white noise, because the power of high frequency components is much lower than that of low frequency components for typical images, the SNR values of high frequency components are smaller. According to (7) and (8), the total distortion is proportional to the product of component power. Hence, a 50% error in one component would translate to 50% total error. Therefore, we should weight the higher frequency components less in the process of selecting quantization parameters in coding. In addition, the unequal frequency weighting also matches the uneven human visual sensitivity.

There are two frequency weighting approaches. One is using the uneven frequency-dependent weights in assigning bits to each frequency component in (3). The other is using uneven weights in computing the total distortion in (4). A special case of the former, frequency-weighted bit allocation, is discussed in Section II-C. A study of the latter, the frequency-weighted distortion case, has been described in [12]. Both approaches, in fact, would lead to similar yet not identical results.

An interesting point of (7) is that the bits and distortion of a (composite) signal are decided by a single parameter E , which is the product of all the components' variances. Thus, we may call E the *entropy variance* of a signal. It represents the *complexity* of the signal. The signal entropy is proportional to $\ln E$. In theory, two signals of the same ordinary variance require different numbers of bits in coding if their *entropy variances* are significantly different.

Assuming the 2-D signals we are dealing with are separable in the horizontal and the vertical directions, then all the above properties can readily be extended to the 2-D signals without significant modifications. The Karhunen–Loève (K–L) transform is recognized as the optimal transform for decorrelating signals and packing the signal energy to the fewest number of transform coefficients. However, K–L transform is data dependent and is computationally intensive because it is derived from the signal autocovariance function. In practical applications, the separable discrete cosine transform (DCT) is often adequate for most of the natural pictures and thus is widely used [12].

B. Quantization

The lossy compression element in Fig. 1 is often, in practice, implemented by a quantizer and an entropy coder such as in JPEG and MPEG. Although the vector quantizer offers potentially better compression efficiency, the scalar quantizer is often used in real systems not only due to its simplicity but also due to its adaptability to the local pictorial data as will be discussed in Section IV.

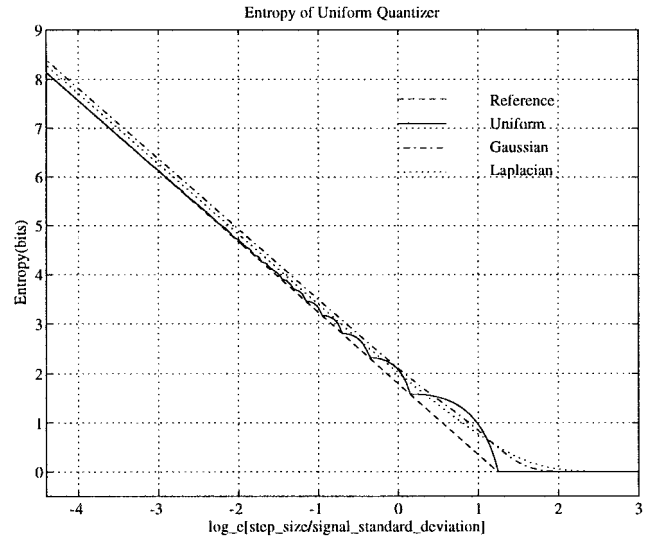


Fig. 2. Entropy of uniform quantizer for sources of different probability density functions (PDF's).

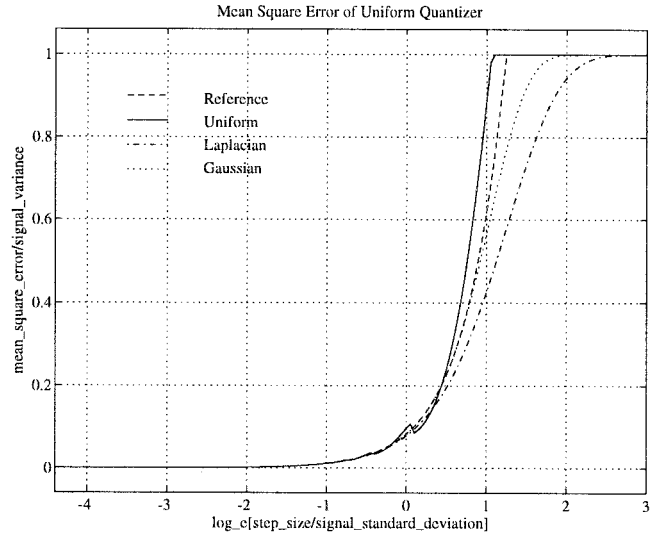


Fig. 3. Mean square error of uniform quantizer for sources of different PDF's.

A uniform midtread quantizer (in which zero is a reconstruction level) is often used in a practical coding system. Typically, the reconstruction levels are the centers of the decision regions, i.e., $\{k\Delta; k = \dots, -1, 0, 1, \dots\}$, where Δ is the quantization step size. The behavior of such a quantizer has been analyzed for inputs with known probability distributions. Except for the uniform distribution, closed-form formulas of entropy, $H_Q(\cdot)$, and distortion, $d_Q(\cdot)$, for arbitrary probability distributions are generally unavailable. The formulation of $H_Q(\cdot)$ and $d_Q(\cdot)$ for uniform, Gaussian, and Laplacian distributions are described in the Appendix. The numerical values of these functions are plotted in Figs. 2 and 3. The “Reference” curves in Figs. 2 and 3 are computed using (9) and (10) with $\epsilon^2 = 1$ (see below).

It is known that, at high bit rates (small distortion), the bits (b) versus distortion (D) relation of an entropy-coded uniform quantizer for a zero-mean i.i.d. source $X(\cdot)$ can be

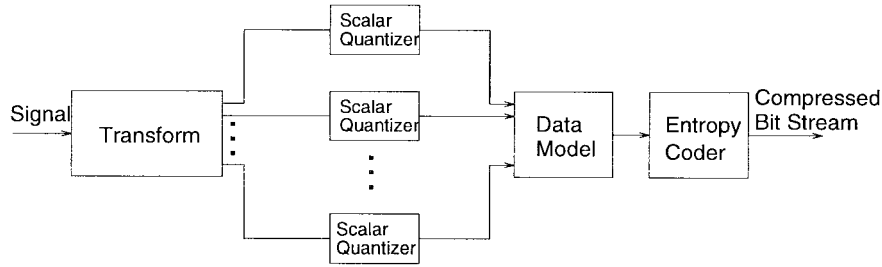


Fig. 4. Practical image transform coder.

approximated by the following formulas [4], [12]:

$$b(D) = \frac{1}{\alpha} \log_e \left(\epsilon^2 \cdot \frac{\sigma_X^2}{D} \right) = \frac{1}{\alpha \log_2 e} \log_2 \left(\epsilon^2 \cdot \frac{\sigma_X^2}{D} \right) \quad (9)$$

and

$$D(b) = \frac{\Delta^2}{\beta}. \quad (10)$$

Thus

$$\frac{\Delta^2}{\beta} = \epsilon^2 \cdot e^{-\alpha b} \cdot \sigma_X^2 \quad (11)$$

where β is 12 and α is 1.386 ($= 2/\log_2 e$) for uniform, Gaussian, and Laplacian distributions, ϵ^2 is source dependent and is about one for uniform distribution, 1.4 for Gaussian, and 1.2 for Laplacian, and σ_X^2 is the signal variance. As shown by Figs. 2 and 3, the above approximations are fairly accurate when the quantization step size is smaller than the signal standard deviation. Combining (9) and (10) we obtain

$$b(\Delta) = \frac{1}{\alpha} \log_e \left(\epsilon^2 \cdot \beta \cdot \frac{\sigma_X^2}{\Delta^2} \right). \quad (12)$$

This gives us a more direct relation between b and Δ .

In image coding, bits are typically spent on a small percentage of dominant transform coefficients of which the allocated bit rates are often higher than a couple of bits per coefficient. Although there is no simple and accurate formula for the low variance coefficients, these coefficients do not affect the overall model very much since their contribution in bits is relatively small. Therefore, the above formulas, although are accurate for medium to high bit rates, they are still practically useful for all the bit rates of interest.

In a real system, the ideal entropy coder is typically replaced by a variable-length coder (VLC), a simplified version of Huffman code [13]. Assuming this VLC is nearly as efficient as the ideal entropy coder, the bits produced by this VLC, \bar{b} , may be approximated by $s_{VLC} \cdot b$, where b is the ideal entropy bits of the quantizer outputs, and s_{VLC} is a scaling factor (for adjustment) greater than one. Under this assumption, (12) may still be used for a practical scalar quantizer with a modified value of α (to be discussed in Section V).

III. PRACTICAL TRANSFORM CODER

A practical image transform coder, such as the DCT coders used in [1]–[3], can be represented by the general block diagram in Fig. 4. It follows roughly the aforementioned

principles, namely, a transform (DCT) used to decompose the original pictures into nearly independent components, a uniform scalar quantizer used to reduce the output levels, and a VLC used to further compress the output bit stream. As compared to that described in Sections II-A and II-B, the individual entropy coders are replaced by a data model followed by an entropy coder. The objective of transformation and data model together is to produce (nearly) statistically independent data sequences so that they can be coded separately with high efficiency.

The data model used in [1]–[3] simply rearranges the transform coefficients in a zigzag scan order. That is, the 2-D array of a block of DCT coefficients are assembled to form a zigzag scanned 1-D linear array or vector $X = [X_0 \cdots X_{L-1}]$, in which the lower frequency's DCT coefficients are usually associated with smaller indices. When the coefficients are ordered in this fashion, the variances of coefficients are approximately monotonically decreasing [2]. If the transformed source vector $X = [X_0 \cdots X_{L-1}]$ is stationary and every frequency component is independent, then the average entropy per block, $H(X_0, \cdots, X_{L-1})$, equals the sum of the entropies of all the components.

According to [12] and [14], the conditions discussed in the above, to a great extent, satisfied; that is, the frequency components (transform coefficients) are nearly i.i.d. sources, the distortion of quantized coefficients are relatively small, and the VLC performance is close to that of the ideal entropy coder. Then, the behavior of such a transform coder can be derived from (9) or (11) by combining all the components together. Assuming the probability distribution of the L frequency components is either uniform, Gaussian, or Laplacian, and b_i is the average bits of the i th entropy-coded, quantized coefficient, the total average bits of such a source is

$$\bar{b}(\bar{D}) = \frac{1}{L} \sum_{i=0}^{L-1} b_i(D_i) \quad (13)$$

$$= \frac{1}{L\alpha} \log_e \left[\prod_{i=0}^{L-1} \left(\epsilon_i^2 \cdot \frac{\sigma_i^2}{D_i} \right) \right] \quad (14)$$

where D_i , σ_i , and ϵ_i^2 are the distortion, the variance, and the ϵ^2 parameter associated with the i th component. Since $D_i = \Delta_i^2/\beta_i$ (10)

$$\bar{D} = \frac{1}{L} \sum_{i=0}^{L-1} D_i = \frac{1}{L} \sum_{i=0}^{L-1} \frac{\Delta_i^2}{\beta_i} \quad (15)$$

where Δ_i is the quantization step size of the i th component, and β_i is the β parameter associated with that component.

Due to the frequency-dependent visual sensitivity of human perception [12], bits assigned to a frequency component should be adjusted according to their perceptual threshold. Thus, the quantization step size of each transform coefficient, Δ_i , ($i = 0, \dots, L-1$) can be different in JPEG [2] and MPEG [3]. In addition, the quantization step sizes in MPEG are made of two components: q_s , a quantization scaling factor for the entire picture block, and $\{W_i, i = 0, \dots, L-1\}$, a weighting matrix whose elements are used as multiplicative factors to produce the true step sizes in quantization. In other words, $\Delta_i = q_s \cdot W_i$. On the other hand, the H.261 specification assumes that all the step sizes in a block are identical [1], that is, $W_i = 1$ for all i . The frequency-dependent step size design implies a frequency-weighted bit allocation scheme discussed in Section II-A in the sense that some frequency components get assigned fewer bits because their actual step sizes used in quantization are larger; that is [from (12)]

$$b_i(q_s) = \frac{1}{\alpha} \log_e \left(\frac{\beta_i \cdot \epsilon_i^2 \cdot \sigma_i^2}{q_s^2 \cdot W_i^2} \right). \quad (16)$$

Therefore, (14) and (15) become

$$\begin{aligned} \bar{b}(\bar{D}) &= \frac{1}{L\alpha} \log_e \left[\left(\frac{1}{q_s^2} \right)^L \cdot \prod_{i=0}^{L-1} \left(\frac{\beta_i \cdot \epsilon_i^2 \cdot \sigma_i^2}{W_i^2} \right) \right] \\ &= \frac{1}{\alpha} \log_e \left[\frac{1}{q_s^2} F \right] \end{aligned} \quad (17)$$

with

$$F = \left[\prod_{i=0}^{L-1} \left(\frac{\beta_i \cdot \epsilon_i^2 \cdot \sigma_i^2}{W_i^2} \right) \right]^{1/L}$$

and

$$\bar{D} = \frac{q_s^2}{L} \sum_{i=0}^{L-1} \frac{W_i^2}{\beta_i}. \quad (18)$$

They completely describe the bits and distortion behavior of an MPEG or JPEG transform coder under the ergodic Gaussian signal assumption. A special case that $\beta_i = \beta$ and $W_i = 1$ for all i has been reported in [7].

In reality, a frequency component may have an *effective variance* ($= \epsilon_i^2 \cdot \sigma_i^2$) less than the weighted distortion, $W_i^2 \cdot q_s^2 / \beta_i$. Also, as the theory predicts, the value of β_i is close to 12 when σ_i is much higher than $W_i \cdot q_s$, but its value may be different if σ_i is close to or smaller than $W_i \cdot q_s$. We then need to go back to (3) and (4), and modify (17) and (18) to the following:

$$q_s^{[2(LA_2/L)]} = F \cdot e^{-\alpha \bar{b}} \quad (19)$$

$$\bar{D} = \frac{q_s^2}{L} \sum_{i \in A_2} \frac{W_i^2}{\beta_i} + \frac{1}{L} \sum_{i \in B_2} \sigma_i^2 \quad (20)$$

where regions of A_2 and B_2 are

$$\begin{cases} A_2: & \left\{ i \in \{0, 1, \dots, L-1\} \text{ and } (\epsilon_i^2 \cdot \sigma_i^2) \geq \frac{W_i^2 \cdot q_s^2}{\beta_i} \right\} \\ B_2: & \{0, 1, \dots, L-1\} - A_2 \end{cases}$$

L_{A_2} is the size of set A_2 , and

$$F = \left[\prod_{i \in A_2} \left(\frac{\beta_i \cdot \epsilon_i^2 \cdot \sigma_i^2}{W_i^2} \right) \right]^{1/L}.$$

Using optimization techniques, Netravali and Haskell derived similar but not identical results under the frequency-weighted distortion criterion [12].

IV. THRESHOLD TRANSFORM CODER

In digital image coding, theory and practice do not agree completely due to several nonideal factors. First, the assumptions undertaken by theory (in the previous sections) such as ergodicity and stationary of image sources do not hold exactly on real data. Second and more importantly, the human visual system is highly nonlinear and cannot be approximated by a simple distortion measure such as the mean square error (MSE). For example, a very small percentage of a coded picture bearing visible artifacts does not affect the overall MSE very much; however, a human viewer can easily pick up the distorted areas, and thus the entire picture quality is rated low. Therefore, in image coding we have to deal with not only the statistical behavior of the entire picture (objective criterion) but also the fidelity of individual samples embedded in their texture neighborhood (subjective criterion).

Several intuitive schemes have been proposed to solve the above problem. Instead of selecting a fixed number of transform coefficients according to their average variances as suggested by the theory, we select the coded coefficients by their magnitudes, the so-called *threshold transform coding*. The direct implementation of threshold coding requires transmitting the *threshold mask*—locations of the chosen coefficients—which often needs a significant number of bits. A popular approach is rearranging the transform coefficients in a fixed scanning order, transmitting them sequentially until the last above-threshold coefficient is hit, and then appending an end-of-block code to conclude this data block. The exact analysis of such a coder is involved and may not be worth the efforts since the real image data do not match exactly our stationary model assumption. The following simplification seems to be sufficient for image coding purpose.

Our focus is on the bit-distortion model, i.e., the average bits and distortion associated with a typical block. Assuming that T_i is the threshold value used in picking up the i th transform coefficient; that is, the i th coefficient is set to zero before quantizing if its magnitude is less than T_i . A popular variant of this scheme is the *dead zone* quantizer—the decision region of the zero reconstruction level is larger than the decision region used for the other reconstruction levels. In both cases and the combined situation, the analysis of such quantizers is similar to that of a simple uniform quantizer; however, the decision levels and the reconstruction levels have to be modified in the analysis equations. Details of the above modifications are described in the Appendix.

For various *threshold* (T) values, the entropy versus distortion curves of a uniform quantizer with a dead zone set to $n \cdot \Delta$, $n = 1, 2, 3$, are plotted in Figs. 5 and 6. The

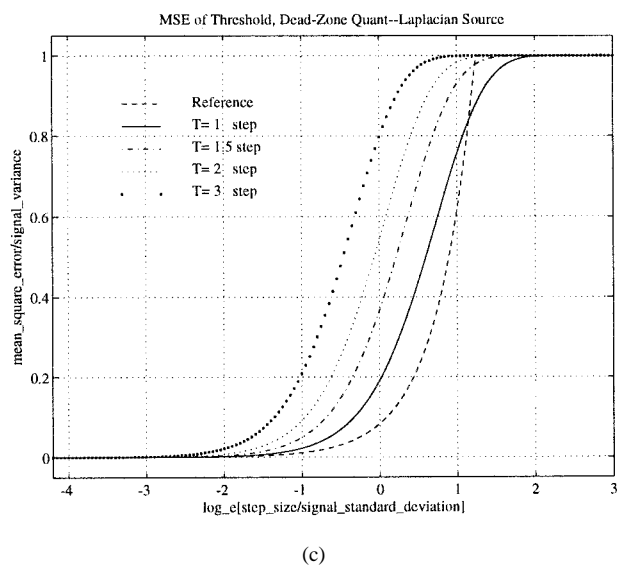
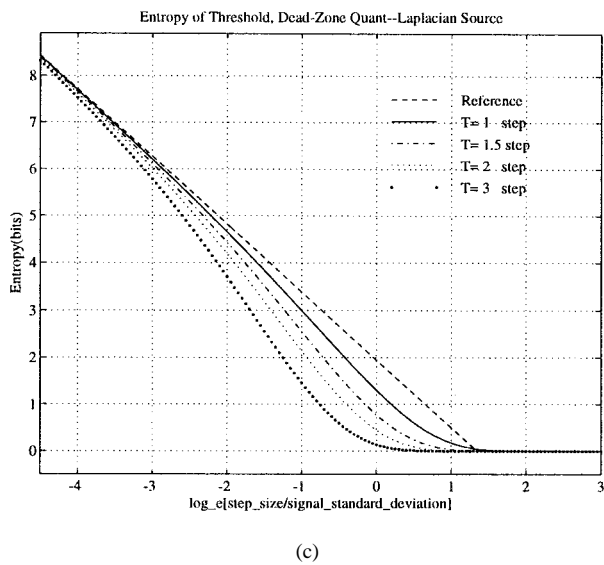
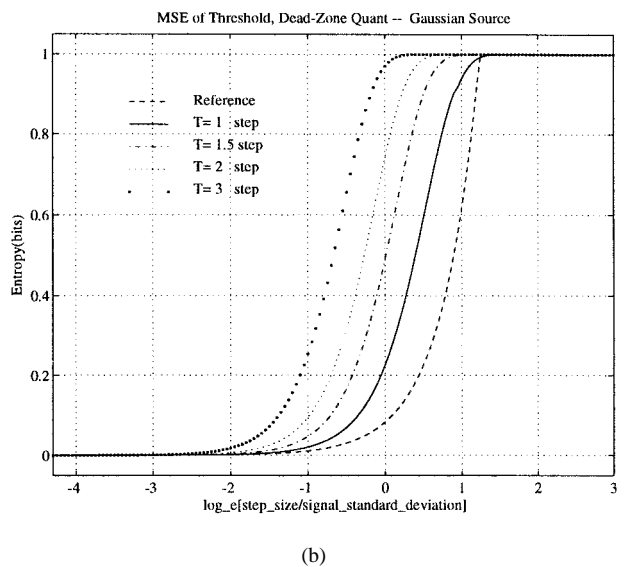
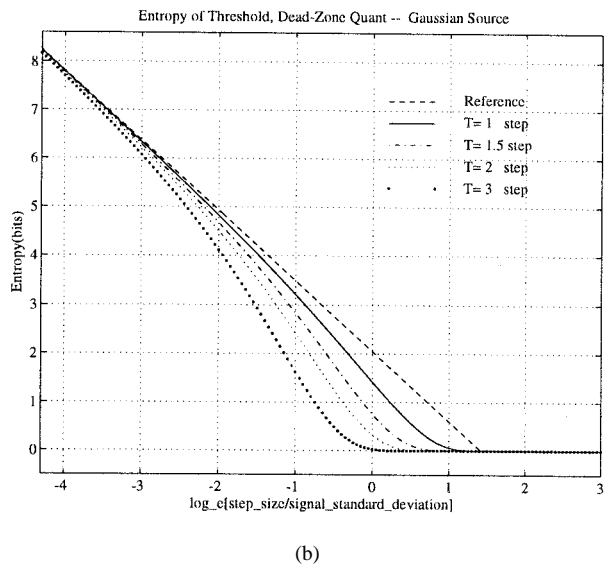
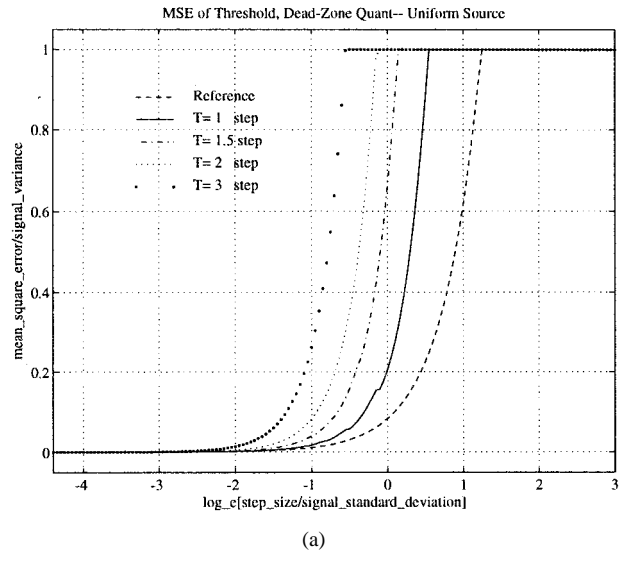
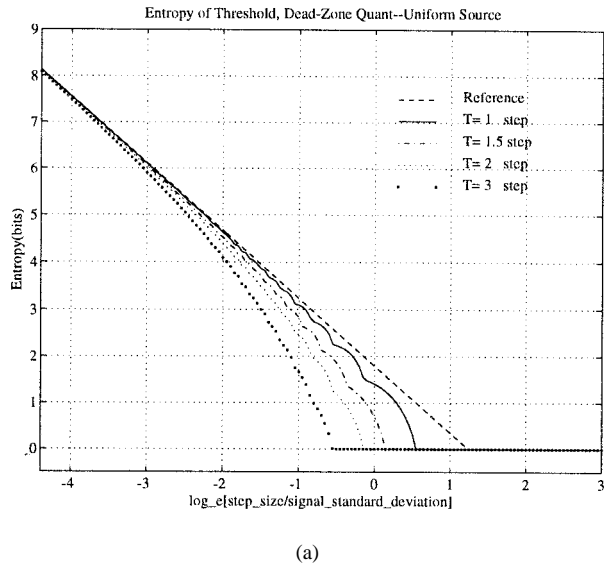


Fig. 5. Entropy of quantizers with dead-zone for (a) uniform, (b) Gaussian, and (c) Laplacian sources.

Fig. 6. MSE of quantizers with dead-zone for (a) uniform, (b) Gaussian, and (c) Laplacian sources.

Reference curves shown on those plots are the asymptotic formulas described by (9) and (10).

In image coding, the quantization step sizes of significant frequency components are usually smaller than the signal standard deviation; therefore, (10) and (12) are reasonably good approximations if α , β , and ϵ^2 are adjusted appropriately. The values of these parameters, in general, depend upon T and σ/Δ . However, for a fixed T and a certain range of step sizes, they can be approximated by constants. When the step size gets larger, (10) and (12) become less accurate. But in these cases the bits produced by those coefficients are very small and thus do not change very much the total bits. Extensions and modifications of these parameters are described in the next section.

Assuming that the bits on the average needed to encode an above-threshold transform coefficient are represented by [from (12)]

$$b_i(\Delta_i) = \frac{1}{\alpha_i(\Delta_i, T_i)} \log_e \left[\frac{\beta_i(\Delta_i, T_i) \cdot \epsilon^2 \cdot \sigma_i^2}{\Delta_i^2} \right] \quad (21)$$

where $i \in A_2$, T_i is the threshold value used in picking up the i th transform coefficient, and $\beta_i(\cdot)$ and ϵ^2 are source-dependent parameters. The average bits number of a pixel becomes

$$\bar{b} = \frac{1}{L} \left[\sum_{i \in A_2} b_i(\Delta_i) + b_{\text{EOB}} \right] \quad (22)$$

where b_{EOB} are the bits for the end-of-block signal.

If the weighting matrix in (17) is adopted, the average bits and the average distortion can be made more explicit

$$\bar{b}(q_s) = \frac{1}{L} \left[\sum_{i \in A_2} \frac{1}{\alpha_i(q_s, T_i)} \log_e \left(\frac{\beta_i(q_s, T_i) \cdot \epsilon^2 \cdot \sigma_i^2}{W_i^2 \cdot q_s^2} \right) + b_{\text{EOB}} \right] \quad (23)$$

and

$$\bar{D} = \frac{q_s^2}{L} \sum_{i \in A_2} \left[\frac{W_i^2}{\beta_i(q_s, T_i)} \right] + \frac{1}{L} \sum_{i \in B_2} \sigma_i^2. \quad (24)$$

Since the threshold transform coding with weighting matrix is invented to match the subjective distortion criterion, the mean square error distortion calculation, (24), is not as useful as the bits calculation, (23), which can be used to adjust the quantization step for regulating encoder buffer and controlling picture quality.

If T_i is chosen to be $[constant \cdot W_i \cdot q_s]$ with roughly the same *constant* value for all the coefficients, then α , β , and ϵ^2 can be expressed as functions of q_s and W_i only. Also, in most image coding cases, the ac components have approximately Laplacian distribution and the dc component, uniform or Gaussian distribution [12]. Hence, the $\alpha_i(\cdot)$ values are similar for all the frequency components as indicated by Fig. 5, in which $\alpha_i(\cdot)$'s are proportional to the slopes of

these curves and have about the same value for all the three distributions. Therefore, replacing $\alpha_i(\cdot)$ in (23) by $\alpha(\cdot)$, we obtain

$$\begin{aligned} \bar{b}(q_s) &= \frac{1}{L\alpha(q_s)} \sum_{i \in A_2} \log_e \left[\frac{\beta_i(q_s) \cdot \epsilon^2 \cdot \sigma_i^2}{W_i^2} \right] \\ &\quad - \frac{1}{L\alpha(q_s)} \sum_{i \in A_2} \log_e q_s^2 + \frac{b_{\text{EOB}}}{L} \\ &= -\frac{1}{\alpha(q_s)} \log_e q_s^2 + \frac{1}{\alpha(q_s)} \\ &\quad \cdot \left\{ \frac{1}{L} \sum_{i \in A_2} \log_e \left[\frac{\beta_i(q_s) \cdot \epsilon^2 \cdot \sigma_i^2}{W_i^2} \right] \right. \\ &\quad \left. + \alpha(q_s) \cdot \frac{b_{\text{EOB}}}{L} \right\} \\ &= -\frac{1}{\alpha(q_s)} \log_e q_s^2 + \frac{G(q_s)}{\alpha(q_s)} \end{aligned} \quad (25)$$

where

$$G(q_s) = \frac{1}{L} \sum_{i \in A_2} \log_e \left[\frac{\beta_i(q_s) \cdot \epsilon^2 \cdot \sigma_i^2}{W_i^2} \right] + \alpha(q_s) \cdot \frac{b_{\text{EOB}}}{L}. \quad (26)$$

Or

$$q_s^2 = F(q_s) \cdot e^{-\alpha(q_s) \cdot \bar{b}} \quad (27)$$

with $F(q_s) = \exp \{G(q_s)\}$. Note that we denote $\beta(\sigma_i, q_s)$ by $\beta_i(q_s)$ in the above equations for simplicity.

The direct use of (25) seems to be fairly complicated—it needs to estimate a number of parameters, $\alpha_i(\cdot)$'s, $\beta_i(\cdot)$'s, and ϵ^2 , computed from image data. If, however, we could assume that the picture to be coded is not much different from the picture that has already been coded in the sense that the α_i , β_i , and ϵ^2 remain about the same in the neighborhood of q_s that we are dealing with, then the F parameter in (27) can be estimated from the \bar{b} and q_s of the coded pictures. Typically, the α value is less picture-dependent, only the F value has to be estimated from image data. Consequently, the entire model identification procedure can be relatively simple.

V. MODEL PARAMETERS

For a practical application, the parameters of the above model have to be adjusted to cope with the specific video coder used and the real picture characteristics. The meaning of the parameters in our model (23) and (24) suggests the following modifications. First, since ϵ^2 is a factor mainly determined by probability distribution, we assume ϵ^2 is a constant (≈ 1.2 for Laplacian source, say) for the rest of analysis. Second, the value of $\beta(\sigma/\Delta, T)$ is no longer constant for small σ/Δ ; however, those values can be precalculated and stored in a table for real-time applications. Third, the constant $\alpha(\approx 1.386)$ is replaced by a parametric function $\alpha(q_s) = 1.386 \cdot \alpha_s(q_s)$ to compensate for the mismatch between the ideal model and a practical video coder. Although these parameters (α , ϵ , and β) may be somewhat related, for simplicity we study them separately. We will elaborate on the second and the third items below.

A. Model Parameter β

We wish to find the model parameter β for different T and (σ/Δ) under the assumption that ϵ^2 is a constant ($=1.2$). If the quantization error is i.i.d. and is uniformly distributed, the theoretical value of β is 12. This is approximately true when the quantization step is much smaller than the signal variance (the left portion of Fig. 7). However, the above assumption is not valid for large quantization step sizes. Since the high frequency transform coefficients are roughly Laplacian distributed, we are interested in the Laplacian cases. Combining (21) and (A9) in the Appendix, we obtain the $\beta(\Delta/\sigma, T)$ value for the Laplacian probability distribution

$$\beta\left(\frac{\Delta}{\sigma}, T\right) = \frac{1}{\epsilon^2 \cdot \left(\frac{\sigma}{\Delta}\right)^2} \cdot e^{\alpha \cdot H_Q[\Delta/\sigma, T+\Delta/2]}. \quad (28)$$

The β values calculated based on this formula are plotted in Fig. 7. From Fig. 7, we find that β increases abruptly after $\log_e(\Delta/\sigma)$ exceeds a threshold value (≈ 1.3). This is because when the quantization step size is very large (comparing to the signal variance), quantization errors can no longer be treated as uniformly distributed. Therefore, the distortion goes up at a slower pace than the asymptotic formula predicts. Eventually, the distortion saturates when it approaches the signal variance (Fig. 6). However, in order to use the same distortion function, (10), for the cases that their distortion is approaching signal variance, the corresponding β values are made growing exponentially (Fig. 7). In real-time applications, it is not likely to compute $\beta(\Delta/\sigma, T)$ directly based upon the above equations. For a specific system, MPEG say, we choose a fixed T ($T = 1 \cdot \Delta$ in this instance) and build a look-up table for real-time use. We did the same thing in the following simulations. One may note from Fig. 5(c) that entropy ≈ 0 when $\log_e(\Delta/\sigma) \geq 1.5$. Hence the look-up table needs only to store the β values for $\log_e(\Delta/\sigma)$ smaller than 1.5. For $\log_e(\Delta/\sigma) \geq 1.5$, the corresponding distortion value is σ^2 , and in this case, based on the original definition of β in (10), β becomes $(q_s^2 \cdot W_i^2)/(\epsilon^2 \cdot \sigma_i^2)$. On the other extreme for $\log_e(\Delta/\sigma) \leq -4$, a constant of 12 is assigned to β .

B. Nonideal Factors in a Practical Coder and Parameter α

Mismatches between the theoretical entropy model and the real VLC coded bits are discussed in this section. Several scaling factors are observed and estimated from experimental data to compensate for the mismatches. In Section II, α is initially treated as constant ($=1.386$), but later analysis suggests that α varies depending upon the value of quantization step size.

First, the probability density functions (PDF's) of the ac coefficients are not ideal Laplacian distribution of equal variance. This PDF mismatch, shown in Fig. 8, results in a smaller entropy for real transform coefficients than the predicted value based on Laplacian assumption. As the quantization step increases, the probability of the quantized values around zero grows much larger than that of the other intervals, and the PDF mismatch problem becomes even more serious. We thus introduce a multiplicative factor $S_p(\geq 1)$ to compensate for the difference. Second, the transform coefficients within a block

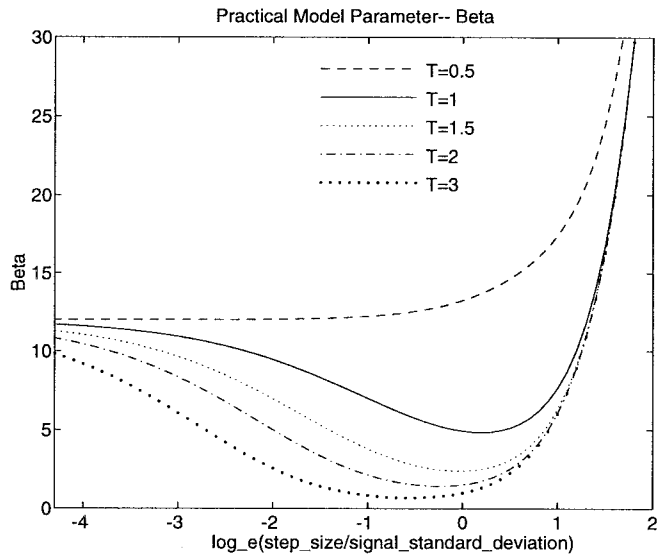


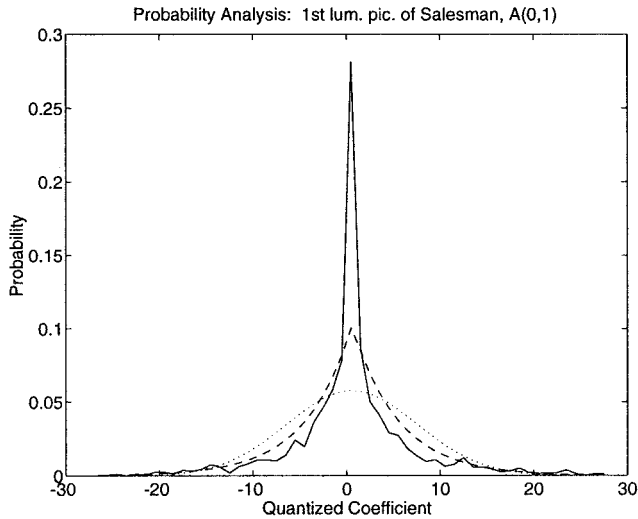
Fig. 7. Model parameter β for various dead-zone values (T) and quantization step sizes (Δ).

are somewhat correlated; hence, another multiplicative factor, $S_d(\geq 1)$, is introduced to correct the original i.i.d. assumption. Third, the VLC table in image coding standards is built based on the probability of zero coefficient runs and coefficient levels, $p(\text{run}, \text{level})$, rather than on $p(X_0, X_1, \dots, X_{L-1})$. Therefore, the average encoded bits per block are higher than the theoretical block entropy. We denote this inefficiency of VLC table by the third multiplicative parameter $S_{vlc}(\leq 1)$. Combining these factors together as a whole, we obtain

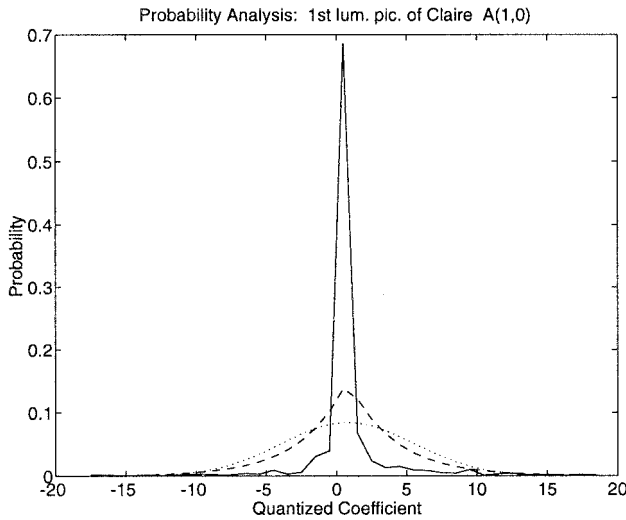
$$\begin{aligned} \sum_{t=0}^{L-1} H_{\text{ideal}}(X_t) &= S_p \sum_{t=0}^{L-1} H_{\text{practical}}(X_t) \\ &= S_p \cdot S_d \cdot H_{\text{practical}}(X_0, \dots, X_{L-1}) \\ &= S_p \cdot S_d \cdot S_{vlc} \cdot \text{bits}_{vlc} \\ &= \frac{1}{S} \cdot \text{bits}_{vlc} \end{aligned} \quad (29)$$

where bits_{vlc} represents the real coded bits. All the above multiplicative scaling factors are functions of quantization scale (q_s). Hence, the overall factor S is, in general, a function of q_s . Practically, we only need to estimate the overall factor $S(q_s)$ from the real data. Therefore, (27) is still valid when \bar{b} is replaced by bits_{vlc}/L , as long as the new $\alpha(q_s)$ includes the nonideal factor S .

In summary, the original α [$\alpha(q_s)$ in (25) and (27)] is now replaced by $1.386 \cdot \alpha_s(q_s)$, where $\alpha_s(q_s)$ is a function of q_s , and it includes the nonideal factors in entropy coding. Once we decide the β value (Section V-A), (27) can be used to compute $\alpha_s(q_s)$ from data. An H.261-type coding structure is used as an example. Since the dc coefficients are coded with fixed length codewords and their coding is independent of the ac coefficients coding, we concentrated on the statistical analysis of the ac coefficients. Fig. 9 shows the $\alpha_s(q_s)$ values computed from the intracoded frames of several video sequences at different q_s values. In this experiment, the β values are taken from Fig. 7, W_i is set to "1" for all the frequency components, and b_{EOB} is neglected.



(a)



(b)

Fig. 8. Probability density functions (solid line) of the first ac component of transform coefficients for (a) Salesman and (b) Claire. The dash lines are Laplacian density functions of the same variance, and the dotted lines are Gaussian density functions of the same variance.

It can be seen that the $\alpha_s(q_s)$ values are getting larger and approaching one when the quantization scales are getting small. This is because the quantized coefficients are less correlated at small quantization scales. Hence, the encoded bits are well estimated by our original entropy model. For larger quantization step sizes, the original model predicts fewer bits than the true bits. To compensate for this fact, $\alpha_s(q_s)$ is significantly smaller than one for large q_s .

In order to use our model in predicting coded bits, we look for a simple arithmetic expression of $\alpha(q_s)$. A first-order linear curve fitting is obtained from experimental data as follows:

$$\alpha_s(q_s) = a_\alpha \cdot q_s + b_\alpha \quad (30)$$

where a_α and b_α are two picture dependent constants. Since $\sigma_{ave} = \sum_{i=2}^N \sigma_i / (N-1)$ is an indication of image complexity, we derive the following empirical formulas for typical values of a_α and b_α : $a_\alpha = \sigma_{ave} \cdot 7 \times 10^{-4} + \lfloor \sigma_{ave}/10 \rfloor \cdot 0.0018 -$

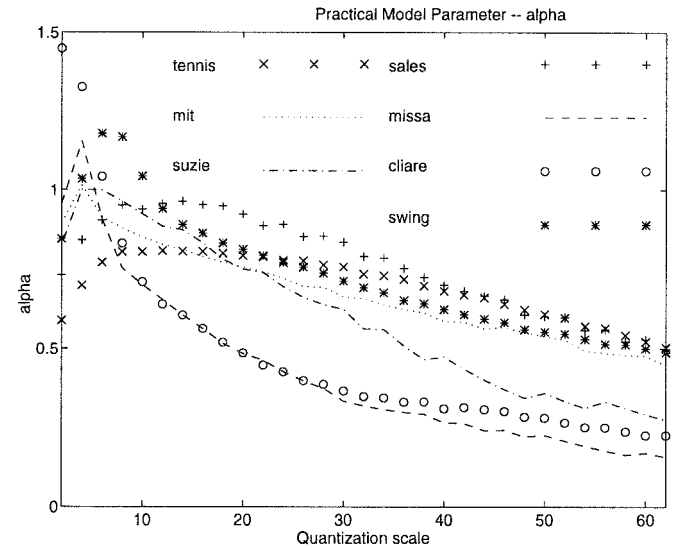


Fig. 9. Model parameter $\alpha_s(q_s)$ for various pictures.

$$\lfloor \sigma_{ave}/20 \rfloor \cdot 0.0096 - 0.02 \text{ and } b_\alpha = \lfloor \sigma_{ave}/10 \rfloor \cdot 0.3 - \lfloor \sigma_{ave}/20 \rfloor \cdot 0.37 + 0.80.$$

Thus far, we have obtained the parametric formulas of $\alpha(q_s)$ [$= 1.386 \cdot \alpha_s(q_s)$] and $\beta(\sigma/\Delta)$. These expressions can be used to estimate the bits needed to encode a picture with a prechosen quantization scale.

C. Bits Prediction

If the coded bits can be predicted prior to the quantization and VLC operations, then the coding process can be well controlled. Based on (25), (27), and the parameters $\alpha(\cdot)$ and $\beta(\cdot)$ described in the preceding subsection, bits needed to encode a picture for a given distortion (quantization step) can be estimated if the variances of transform coefficients are available. The picture characteristics from coding point of view are completely specified by the function $F(q_s)$ in (27). This function, $F(q_s)$, can thus be considered as the measure of picture coding complexity and therefore is named “coding complexity function.” If we keep $\alpha(q_s)$ in (27) as a constant and allow $F(q_s)$ to vary to accommodate for the varying $\alpha(q_s)$, the computation in quantizer control procedure can then be simplified somewhat. In this case, the resultant $F(q_s)$ is called “modified $F(q_s)$ ” and is denoted by $F^m(q_s)$. For easy distinction, we call $F(q_s)$ the “original $F(q_s)$ ” denoted by $F^o(q_s)$ if it is computed using (27) directly based on the variances of block transform coefficients. Some examples of $F^m(\cdot)$ and $F^o(\cdot)$ are shown in Fig. 10. An interesting observation is that both $F^m(\cdot)$ and $F^o(\cdot)$ can be approximated rather well by a linear equation (for each picture separately) when q_s 's are relatively small (between 10 and 30), which are the most frequently used values in practice. In other words

$$F(q_s) = a_F \cdot q_s + b_F \quad (31)$$

where a_F and b_F are two picture-dependent constants. We can see from Fig. 10(a) and (b) that the coded bits of different pictures are reflected on their corresponding coding complexity functions. If the picture contents do not change significantly

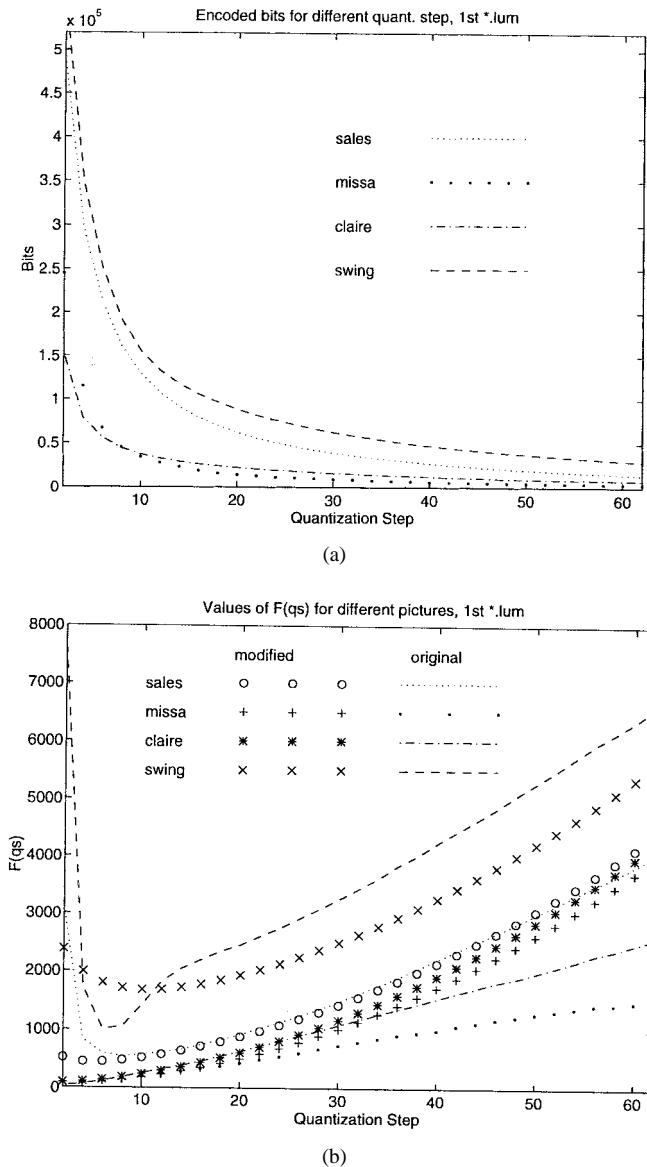


Fig. 10. The coding complexity function for various pictures (the first frame in those picture sequences). (a) Coded bits per pixel for various quantization step sizes and (b) the modified $F(q_s)$ ($F^m(q_s)$) and the original $F(q_s)$ ($F^o(q_s)$) values.

between two nearby frames, the $F(q_s)$ computed from the previous frame can still be used to estimate the bits of the current frame at similar step sizes.

We should keep in mind that the above bits model is derived based on several assumptions of the statistical behavior of data. For real pictures and picture sequences, the stationarity and ergodicity assumptions are not completely satisfied. In addition, this simple model uses essentially only $F(q_s)$ to estimate the bits needed to encode an entire picture. Although $\alpha(q_s)$ is made a function of q_s , it is not possible to cover all the nonstationarity and data-dependency fully by only a couple of parameters. Therefore, a small amount of estimation errors cannot be avoided.

VI. CONCLUSIONS

In this paper, we derive a source model that describes the relationship between bits, distortion, and quantization

step sizes for block-transform coders. This source model is initially derived based on rate-distortion theory. The realistic constraints such as quantizer dead-zone and threshold coefficient selection are included in our formulation. The picture complexity from a coding point of view can be measured and computed from the entropy variance which is the product of all the component's variances of image signals. When the model is used in real video coding, image characteristics and nonideal factors of a practical video coder are accommodated by parameterizing the model constants. If the parameter values are properly chosen, the coding behavior can be well estimated by the proposed model. In brief, there are two aspects of our contribution. One, we extend the existing results of rate-distortion theory to the practical video coders, and two, the nonideal factors in real signals and systems are identified and their mathematical expressions are derived from experimental data. One application of this source model, as shown in the second part of this paper, is the buffer/quantizer control in video coders with the advantage that the picture quality is kept nearly constant over the entire picture sequence.

APPENDIX ENTROPY AND DISTORTION OF UNIFORMLY QUANTIZED UNIFORM AND LAPLACIAN SOURCES

A. Uniform Quantizer

The behavior of a uniform quantizer can be analyzed by the following equations for inputs with known probability distributions. Assuming that the probability density function of a zero-mean i.i.d. source X is $p_X(x)$, the entropy bits produced by this uniform quantizer can be calculated by

$$H_Q(\Delta) = \sum_{k=-N}^N P_X[k] \cdot \log P_X[k] \quad (\text{A1})$$

where

$$P_X[k] = \int_{(k-1/2)\Delta}^{(k+1/2)\Delta} p_X(x) dx \quad (\text{A2})$$

and the mean square quantization error is

$$d_Q(\Delta) = \sum_{k=-N}^N \int_{(k-1/2)\Delta}^{(k+1/2)\Delta} (x - k\Delta)^2 p_X(x) dx \quad (\text{A3})$$

where N decides the total number of quantization levels ($2N + 1$), chosen sufficiently large to eliminate almost all the overloaded quantization errors.

B. Uniform Distribution

Assuming that a zero-mean uniform distribution i.i.d. source with variance σ_U^2 is quantized by a uniform quantizer described above. Then, the entropy of the outputs is [5]

$$H_Q(\Delta) = \frac{1}{1.3863} \log_e \left(\frac{12\sigma_U^2}{\Delta^2} \right) \quad (\text{A4})$$

and the mean square quantization error is

$$d_Q(\Delta) = \frac{\Delta^2}{12}. \quad (\text{A5})$$

C. Gaussian Distribution

The probability density function of a zero-mean Gaussian source with variance σ_G^2 is

$$p_G(x) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{x^2}{2\sigma_G^2} \right\}.$$

Unfortunately, there are no closed form formulas of (A1) and (A3) for Gaussian distribution. However, in this case, it is not difficult to compute their numerical values using (A1) and (A3). The results are shown in Figs. 5(b) and 6(b). When Δ/σ is reasonably small, the PDF $p_X(x)$ is approximately constant in a Δ -width interval. We then obtain the asymptotical behavior of such a quantizer at high bit rates as indicated by (9) and (10).

D. Laplacian Distribution

The probability density function of a zero-mean Laplacian source with variance σ_L^2 is

$$p_L(x) = \frac{1}{\sqrt{2}\sigma_L} e^{-\sqrt{2}(|x|/\sigma_L)}.$$

Thus

$$\begin{aligned} P_L[k] &= \int_{(k-1/2)\Delta}^{(k+1/2)\Delta} \frac{1}{\sqrt{2}\sigma_L} e^{-\sqrt{2}(|x|/\sigma_L)} dx \\ &= \frac{1}{2} e^{-[(k-1/2)\Delta/\sqrt{2}\sigma_L]} [1 - e^{-(\sqrt{2}\Delta/\sigma_L)}]. \end{aligned} \quad (\text{A6})$$

The general formulas of entropy and distortion of uniformly-quantized Laplacian source with dead-zone are given below.

E. Dead Zone

Assume that the first reconstruction level of a quantizer with dead-zone is g_0 , then the $2N + 1$ reconstruction levels are

$$y[k] = \begin{cases} 0, & k = 0 \\ g_0 + (k-1)\Delta, & k > 0 \\ -g_0 + (k+1)\Delta, & k < 0. \end{cases}$$

If the corresponding input decision regions are selected as

$$(l[k], h[k]) = \begin{cases} \left(-g_0 + \frac{\Delta}{2}, g_0 - \frac{\Delta}{2} \right), & k = 0 \\ \left(y[k] - \frac{\Delta}{2}, y[k] + \frac{\Delta}{2} \right), & k \neq 0 \end{cases}$$

then (A2) and (A3) are modified to

$$P_X[k] = \int_{l[k]}^{h[k]} p_X(x) dx \quad (\text{A7})$$

and

$$d_Q(\Delta) = \sum_{k=-N}^N \int_{l[k]}^{h[k]} (x - y[k])^2 p_X(x) dx. \quad (\text{A8})$$

For *Laplacian distribution*, let $\lambda = \sqrt{2}/\sigma_L$, $\rho = e^{-\lambda\Delta}$, and $\rho_1 = e^{-\lambda(g_0 - \Delta/2)}$.

The entropy (in bits) and distortion can be derived

$$\begin{aligned} H_Q \left[\frac{\Delta}{\sigma_L}, g_0 \right] &= -(1 - \rho_1) \cdot \log_2 \cdot (1 - \rho_1) \\ &\quad - \rho_1 \left[\log_2 \rho_1 \cdot (1 - \rho) + \frac{\rho}{1 - \rho} \cdot \log_2 \rho - 1 \right] \end{aligned} \quad (\text{A9})$$

and

$$\begin{aligned} d_Q[\sigma_L, \Delta, g_0] &= g_0 \rho_1 \left(\Delta - \frac{2}{\lambda} - g_0 \right) \\ &\quad + \frac{2}{\lambda^2} - \left(\frac{2 \cdot \Delta}{\lambda} \right) \left(\frac{\rho \cdot \rho_1}{1 - \rho} \right). \end{aligned} \quad (\text{A10})$$

If we take the centroid of the decision region $[l[k], h[k]]$ as the reconstruction level, we obtain

$$\begin{aligned} d_{Q(\min)}[\sigma_L, \Delta, g_0] &= -\rho_1 \cdot \left(g_0 - \frac{\Delta}{2} + \frac{1}{\lambda} \right)^2 \\ &\quad + \frac{2}{\lambda^2} - \frac{\Delta^2 \cdot \rho \cdot \rho_1}{(1 - \rho)^2}. \end{aligned} \quad (\text{A11})$$

A uniform quantizer without dead zone may be considered as a special case of the above with $g_0 = \Delta$. In this case, the corresponding entropy and distortion are

$$H_Q \left[\frac{\Delta}{\sigma_L} \right] = H_2(\sqrt{\rho}) + \sqrt{\rho} \left[1 + \frac{H_2(\rho)}{1 - \rho} \right] \quad (\text{A12})$$

and

$$d_{Q(\min)}[\sigma_L, \Delta] = -\rho_1 \left(\frac{\Delta}{2} + \frac{1}{\lambda} \right)^2 + \frac{2}{\lambda^2} - \frac{\Delta^2 \cdot \rho \cdot \rho_1}{(1 - \rho)^2} \quad (\text{A13})$$

where

$$H_2(\rho) = \rho \log_2 \frac{1}{\rho} + (1 - \rho) \log_2 \frac{1}{1 - \rho}. \quad (\text{A14})$$

F. Threshold

If a dead-zone quantizer also has a threshold T and $T > (g_0 - \Delta/2)$, then (A7) and (A8) are modified to

$$P_X[k] = \begin{cases} \int_{-T}^T p_X(x) dx, & k = 0 \\ \int_{l[k]}^{-T} p_X(x) dx, & k = -k_T \\ \int_T^{h[k]} p_X(x) dx, & k = k_T \\ \int_{l[k]}^{h[k]} p_X(x) dx, & \text{otherwise} \end{cases} \quad (\text{A15})$$

and

$$\begin{aligned}
 d_Q(\Delta) = & \int_{-T}^T x^2 p_X(x) dx \\
 & + \int_{l[-k_T]}^{-T} (x - y[-k_T])^2 p_X(x) dx \\
 & + \int_T^{h[k_T]} (x - y[k_T])^2 p_X(x) dx \\
 & + \sum_{k=-N}^{-k_T-1} \int_{l[k]}^{h[k]} (x - y[k])^2 p_X(x) dx \\
 & + \sum_{k=k_T+1}^N \int_{l[k]}^{h[k]} (x - y[k])^2 p_X(x) dx \quad (\text{A16})
 \end{aligned}$$

where

$$k_T = \max \left(\left\lceil \left(\frac{T - g_0}{\Delta} + \frac{1}{2} \right) \right\rceil, 1 \right)$$

and $\lceil a \rceil$ represents the smallest integer that is greater than or equal to a . When $T \leq (g_0 - \Delta/2)$, the threshold operation is not effective. It then simply becomes a regular uniform quantizer.

REFERENCES

- [1] CCITT, Working Party XV/1, "Draft of recommendation H.261: video codec for audiovisual services at $P \times 64$ k bits/s," July 1990.
- [2] W. B. Pennebaker and J. L. Mitchell, *JPEG—Still Image Data Compression Standard*. New York, NY: Van Nostrand Reinhold, 1993.
- [3] MPEG (Motion Picture Experts Group, ISO-IEC JTC1/SC29/WG11), "Coding of Moving Pictures and Associated Audio," CD 11172-2 Nov. 1991.
- [4] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 676–683, Sept. 1968.
- [5] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice Hall, 1984.
- [6] T. Berger, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice Hall, 1971.
- [7] H.-M. Hang *et al.*, "Digital HDTV compression using parallel motion-compensated transform coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 210–221, June 1991.
- [8] A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 351–361, Dec. 1991.

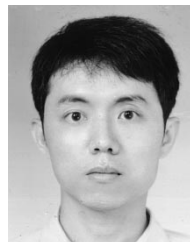
- [9] W.-Y. Sun, H.-M. Hang, and C. B. Fong, "Scene adaptive parameters selection for MPEG syntax based HDTV coding," in *Signal Processing of HDTV*, L. Stenger *et al.*, Eds. Elsevier Science, 1994, vol. V.
- [10] A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding*. New York, NY: McGraw-Hill, 1979.
- [11] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [12] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York, NY: Plenum, 1988.
- [13] R. E. Blahut, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1991.
- [14] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA: Academic, 1990.



Hsueh-Ming Hang (S'79–M'84–SM'91) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1984 to 1991, he was with AT&T Bell Laboratories, Holmdel, NJ, where he was engaged in digital image compression algorithm and architecture researches. He joined the Electronics Engineering Department of National Chiao Tung University, Hsinchu, Taiwan, in December 1991. His current research interests include digital video compression, image/signal processing algorithm and architecture, and digital communication theory.

Dr. Hang was a conference co-chair of Symposium on Visual Communications and Image Processing (VCIP), 1993, and the Program Chair of the same conference in 1995. He guest co-edited two *Optical Engineering* special issues on visual communications and image processing in July 1991 and July 1993. He was an Associate Editor of *IEEE TRANSACTIONS ON IMAGE PROCESSING* from 1992 to 1994 and a Co-Editor of the book *Handbook of Visual Communications* (Academic Press, 1995). He is currently an Editor of *Journal of Visual Communication and Image Representation*, Academic Press. He is a member of Sigma Xi.



Jiann-Jone Chen was born in Tai-Chong, Taiwan, R.O.C., in 1966. He received the B.S. and M.S. degrees in electrical engineering from National Cheng Kung University, Tainan, in 1989 and 1991, respectively. He is currently working toward the Ph.D. degree in electronics engineering at National Chiao Tung University, Hsinchu, Taiwan.

His research interests include medical image processing and digital video coding.