

# Novel Statistical Modeling, Analysis and Implementation of Rate-Distortion Estimation for H.264/AVC Coders

Xin Zhao, Jun Sun, Siwei Ma, and Wen Gao, *Fellow, IEEE*

**Abstract**—In H.264/advanced video coding, the encoder employs the rate-distortion optimization (RDO) to select the optimal coding mode of each block. Although it is effective to employ the RDO technique for mode decision, the computation load increases drastically. To reduce the computation complexity of the RDO technique, in this paper, we propose efficient algorithms for the estimation of block-level rate and distortion. For rate estimation, we model the transform coefficients with accurate generalized Gaussian distributions, and the weighted sum of absolute quantized transform coefficients is proposed as an efficient rate estimator, where the weights provide an implicit mechanism for evaluating different contributions of different frequency components to the coding bits. For distortion estimation, we first analyze the origins of distortion thoroughly. Then a direct relationship between the discarded bits in quantization and the distortion is explored. According to this investigation, a simple and efficient algorithm is proposed for the distortion estimation. With above proposed algorithms, the RDO technique can be efficiently implemented in a low-complexity way. Extensive experimental results demonstrate that, compared with the original RDO implementation, the proposed algorithms achieve about 32% reduced total encoding time with ignorable coding performance degradation.

**Index Terms**—Distortion estimation, GGD, H.264/AVC, rate estimation, rate-distortion optimization, video coding.

## I. INTRODUCTION

**T**O IMPROVE the coding efficiency of block-based hybrid video coding structure, coding tools and strategies become more and more flexible during the development of international video coding standards such as MPEG-1 [1], MPEG-2 [2], MPEG-4 [3], H.263 [4], and the latest H.264/advanced video coding (AVC) [5]. One of these important developing

Manuscript received March 17, 2009; revised June 22, 2009 and August 24, 2009. First version published March 15, 2010; current version published May 5, 2010. This work was supported in part by the National Science Foundation (60833013, 60902004) and the National Basic Research Program of China (973 Program, 2009CB320903, 2009CB320904). This paper was recommended by Associate Editor R. C. Lancini.

X. Zhao is with the Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China and also with the Graduate School of Chinese Academy of Science, Beijing, 100049, China (e-mail: xzhao@jdl.ac.cn).

J. Sun is with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: jsun@pku.edu.cn).

S. Ma and W. Gao are with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: swma@jdl.ac.cn; wgao@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2010.2045803

flexibilities, which significantly improve the coding efficiency, is to increase the number of candidate coding modes. For evaluating the efficiency of each candidate mode in the high-compression video coding, the encoder usually employs the Lagrangian multiplier optimization technique [6], which is expressed by

$$\min\{J\} \quad J = D + \lambda \cdot R \quad (1)$$

where  $J$  is the Lagrangian rate-distortion (R-D) cost function to be minimized,  $\lambda$  is the Lagrangian multiplier,  $D$  and  $R$  are the reconstruction distortion and entropy coding bits of a coding unit with a certain candidate mode. During the mode decision process, all available candidate modes are evaluated by the rate-distortion cost, and the one with the minimum R-D cost is selected as the optimal coding mode. The minimization process of the R-D cost is the well-known rate-distortion optimization (RDO). Although it is efficient to employ the RDO technique for choosing the best mode, the computation complexity is extremely high. For each candidate mode, the encoder has to fully perform transform, quantization, entropy coding, inverse quantization, inverse transform, and pixel reconstruction to obtain the accurate  $R$  and  $D$ . Therefore, the R-D cost calculation is very time-consuming, and the overall complexity of RDO increases linearly with the number of available candidate modes.

To reduce the computation complexity of RDO, many fast mode-decision algorithms were proposed. Typically, these algorithms can be classified into two categories. In the first category, the implicit relationship between the optimal coding mode and the corresponding spatial features or temporal features of a block is studied with empirical experimental results. In this way, unnecessary candidate modes are excluded from the mode-decision process, and thus the times of calculating the R-D cost are reduced. For intra-mode decision, a local edge map direction histogram has been established and successfully used in [8] to predict the most probable intra-prediction direction, and the remaining prediction directions can be safely skipped. For inter-mode decision, the spatial homogeneity and the temporal stationarity of a macroblock were also properly evaluated and successfully employed in [9] to skip unnecessary mode trials.

In the second category, the mode decision process retains the “try all and select the best” [9] philosophy and no candidate mode is skipped as the algorithms do in the first category.

Instead, efforts are dedicated to reducing the complexity of calculating each R-D cost value. More specifically, algorithms in this category attempt to simplify the calculation of R-D cost by estimating the R and/or D part(s) in (1) with a low complexity way. Related to this category, rate models observed from quantizer (Q)-domain in [10] and  $\rho$ -domain in [11] were established and theoretically justified for rate control in video coding. In [12], R-D models based on Cauchy density approximation to the transform coefficient distribution were developed for frame bit allocation in video coding, and the proposed models were verified to be more accurate than the Laplacian-pdf-based models. A quadratic R-D model based on the quantization scheme of H.264 and predicted mean absolute difference was provided in [13] for an efficient adaptive rate-control scheme in H.264/AVC video coding. To resolve the inter-dependence of RDO and rate control, enhanced R-D models for H.264 video were built in [14] with a two-stage encoding scheme, and an enhanced header rate model was also proposed due to the increased importance of header bits in H.264. In [15], a rate model using a linear function of the number and the absolute magnitudes of nonzero quantized transform coefficients was verified in the MPEG-4 rate-control scheme.

For the H.264/AVC mode decision process, the above model in [15] was modified in [16] to achieve accurate mode decision for different quantization parameters and sophisticated coding options. However, this model was designed only for inter-mode decisions and validated for the context adaptive variable length coding (CAVLC) entropy coding method. Rate estimation algorithms specifically designed for CAVLC entropy coder were also proposed in [17]–[19], these algorithms mainly utilized different linear combinations of features (e.g., the magnitude of nonzero coefficients, the number of trailing ones, the number of nonzero coefficients) of quantized transform block as efficient bit-rate estimators. Recently, a novel rate estimation model was proposed in [20] for mode decision of H.264/AVC, and the model employed a weighting form for evaluating different contributions of different frequency bands to the entropy coding bits. For the D part in R-D cost, a distortion measurement obtained in transform domain was proposed and combined with a proposed rate estimation model in [16]. With the proposed method, inverse transform and the reconstruction processes can be saved, but the inverse quantization and some additional mathematical manipulations were also needed. To further reduce the computation complexity, in [21], a new fast sum of squared difference computation algorithm based on an iterative table-lookup method was proposed. This algorithm saves arithmetic operations at the cost of increasing memory needed for storing the look-up table, and the assumption of the algorithm becomes violated when the quantization offsets are adaptively changed [33]. Moreover, a new R-D estimation algorithm was obtained in [22] to save the transform and quantization computation time by modeling the transform coefficients with spatial-domain variance, and the proposed method was employed in both mode decision and rate-control schemes.

To obtain a general rate estimator efficient for both CAVLC and context adaptive binary arithmetic coding (CABAC) and

to further simplify the implementation of distortion estimation, we propose novel block-level rate and distortion estimation algorithms in this paper, which belong to the second category introduced above. The three main contributions of this paper are listed as follows.

- 1) For rate estimation, a novel weighted sum of quantized transform coefficients is deduced and proposed as an efficient rate estimator. The proposed method improves the accuracy of existing methods by weighting the transform coefficients because of their different contributions to the coding bits.
- 2) For distortion estimation, the origin of distortion in H.264/AVC video coding is first analyzed thoroughly. Then a new transform-domain distortion (TDD) estimation method is proposed using the discarded lower bits in the quantization process. With the proposed method, the distortion estimation in [16] is further simplified without loss of accuracy.
- 3) A low-complexity mode decision scheme is proposed based on the novel rate and distortion estimation algorithms. In the proposed mode decision scheme, the processing of entropy coding, inverse quantization, inverse transform and pixel reconstruction in the conventional RDO is skipped.

The remainder of this paper is organized as follows. In Section II, a mathematical analysis of the relationship between self-information and magnitude of a single transform coefficient is given. The proposed rate and distortion estimation algorithms are introduced in Sections III and IV, respectively. Extensive experimental results and analysis are shown in Section V. Finally, we give conclusion in Section VI.

## II. STATISTICAL MODELING AND ANALYSIS OF BLOCK-LEVEL RATE ESTIMATION

In this section, we start with two motivating observations, which provide useful guidelines for modeling the rate. Then we employ a zero-mean generalized Gaussian distribution (GGD) to model the distribution of transform coefficients, and the mathematical relationship between the self-information and the magnitude of a single quantized transform coefficient is deduced. Based on the deduction, a novel block-level rate estimation model is constructed. Finally, we give comparisons of the effectiveness between our proposed rate estimator and some previous estimators.

### A. Motivating Observations of Actual Quantized Blocks and Corresponding Coding Bits

The  $4 \times 4$  matrices shown in Figs. 1 and 2 are two groups of quantized transform blocks of *Foreman* with common intermediate format (CIF). In Fig. 1, the  $l_1$ -norms of the two blocks are both 25, but the entropy coding bits of the left block is 67, while the right one, where the power mainly distributes to the low frequency components results in only 30 coding bits. In Fig. 2, the two blocks share the same number of nonzero coefficients, but the left block results in 85 coding bits, while the right one of which the power is much lower results in only 27 coding bits.

$$\begin{bmatrix} 1 & 0 & 1 & 3 \\ 2 & 2 & 4 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 14 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 1. Two quantized transform block from *Paris* in CIF format with  $l_1$ -norm both equal to 25. The entropy coding bits of the two blocks are 67 and 30 bits, respectively.

$$\begin{bmatrix} 20 & 13 & 2 & 0 \\ 20 & 14 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 6 & 6 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Fig. 2. Two quantized transform block from *Paris* in CIF format with number of nonzero coefficients both equal to 10. The entropy coding bits of the two blocks are 85 and 27 bits, respectively.

These two motivating observations demonstrate the limitations of existing rate estimation models that even when the  $l_1$ -norms or the number of nonzero coefficients are exactly the same, the actual coding bits can be still quite different. The difference originates from the different power distributions of quantized transform coefficients, i.e., different frequency components contribute differently to the entropy coding bits. Therefore, the quantized transform coefficients need to be weighted before being used to estimate the rate. However, to the best of our knowledge, there is no proposed algorithm in the literature that weights the quantized transform coefficients for rate estimation purposes.

### B. Generalized Gaussian Modeling for Integer Transform Coefficients of H.264/AVC

To get a theoretically justified weighting form of the quantized transform coefficients, we start with modeling a single transform coefficient  $C_{uv}$  with zero-mean GGD [23] described as follows:

$$f_{uv}(x) = \frac{\eta_{uv}\alpha_{uv}(\eta_{uv})}{2\sigma_{uv}\Gamma(1/\eta_{uv})} \exp\left\{-\left[\alpha_{uv}(\eta_{uv})\frac{|x|}{\sigma_{uv}}\right]^{\eta_{uv}}\right\} \quad (2a)$$

with

$$\alpha_{uv}(\eta_{uv}) = \sqrt{\frac{\Gamma(3/\eta_{uv})}{\Gamma(1/\eta_{uv})}} \quad (2b)$$

where  $f_{uv}(x)$  indicates the probability density function (PDF) of  $C_{uv}$ ,  $\Gamma(\cdot)$  is the gamma function,  $\eta_{uv}$  and  $\sigma_{uv}$  are positive real-valued distribution parameters which control the shape and scale of the GGD, respectively. We employ the GGD for analyzing the distribution of transform coefficients because it is a flexible PDF which covers a wide range of symmetrical distributions. As shown in Fig. 3, for the particular cases  $\eta = 1$ ,  $\eta = 2$ , and  $\eta = \infty$ , the GGD becomes Laplacian, Gaussian, and uniform PDF, respectively. Due to the flexibility, GGD is efficient for simulating the various distributions of discrete cosine transform (DCT) coefficients and has been efficiently used in different research areas for analyzing the DCT coefficients [24]–[27].

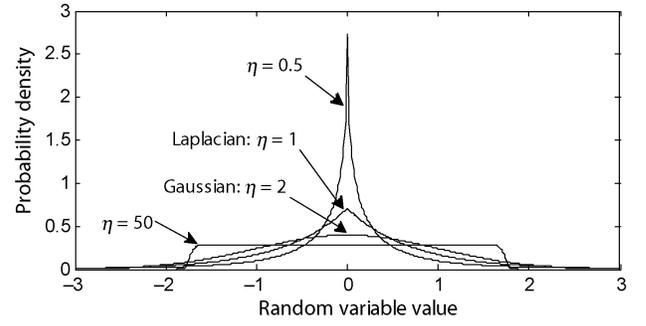


Fig. 3. Generalized Gaussian distribution with shape parameter  $\eta = 0.5, 1, 2,$  and  $50$ .

There are several methods [28]–[30] in the literature for estimating the GGD parameters  $\eta$  and  $\sigma$ . These methods are mainly based on the mathematical relationship between the moments and the parameters given by

$$R(\eta) = \frac{\Gamma^2(2/\eta)}{\Gamma(1/\eta)\Gamma(3/\eta)} = \frac{E^2\{|X|\}}{E\{X^2\}} \quad (3a)$$

$$\sigma^2 = E\{X^2\} \quad (3b)$$

where  $R(\eta)$  is the reciprocal of the so-called generalized Gaussian ratio function introduced in [28],  $E\{X\}$  and  $E\{X^2\}$  represent the first-order moment and the second-order moment of a random variable  $X$ , respectively. With (3a) and (3b), the GGD parameters  $\eta$  and  $\sigma$  are estimated by

$$\hat{\eta} = R^{-1}\left(\frac{\left(\frac{1}{N}\sum_{i=1}^N|X_i|\right)^2}{\frac{1}{N}\sum_{i=1}^N X_i^2}\right) \quad \hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^N X_i^2}. \quad (4)$$

In [30], a simple function  $H(x) = 0.2718/(0.7697 - x) - 0.1247$  is used to directly approximate the inverse function of  $R(x)$ , and we utilize this method in our simulations due to its simplicity and efficiency. The gamma function in (2a) and (2b) is calculated using the method introduced in [31].

### C. Relationship Between Self-Information and Magnitude of Quantized Transform Coefficient

According to the design rules of optimal entropy coders [35], the entropy coding bits of a symbol  $x$  with occurrence probability  $P_x$  is directly dependent on the self-information ( $-\log_2 P_x$ ). Therefore, it is important to relate the entropy coding bits of a single quantized transform coefficient to its self-information. Let  $C_{uv}$  be the transform coefficient located at position  $(u, v)$  in the  $4 \times 4$  transform block, and its quantized value is represented by  $\hat{C}_{uv}$ , where  $u, v \in [0, 3]$ . Then the self-information of  $\hat{C}_{uv}$  being quantized as  $\hat{x}$  is calculated by

$$r_{uv} = -\log_2 P\{\hat{C}_{uv} = \hat{x}\} \quad (5)$$

where  $P\{\cdot\}$  represents the occurrence probability of an event.

As the former video coding standards, H.264/AVC also adopts a uniform scalar quantizer [32] formulated as

$$\hat{C}_{uv} = \text{sign}(C_{uv}) \cdot (|C_{uv}| \cdot Q + f \cdot 2^{\text{qbits}}) \gg \text{qbits} \quad (6)$$

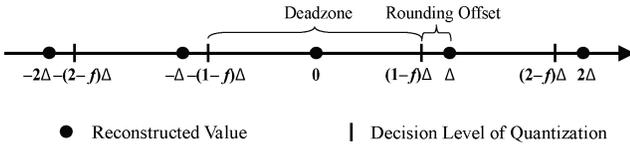


Fig. 4. Quantizer structure in the H.264/AVC encoder.

where  $Q$  is the multiplication factor,  $f$  is the parameter controlling the rounding offset, and  $\gg$  denotes right shift and rounding operation,  $sign(x)$  is  $-1$  with negative  $x$ , and  $1$  for positive  $x$ . A visualized description of the quantizer formulated in (6) is shown in Fig. 4.

With the quantization structure in (6), the probability of  $C_{uv}$  being quantized as  $\hat{x}$  is calculated by

$$P\{\hat{C}_{uv} = \hat{x}\} = \begin{cases} \int_{(|\hat{x}|-f) \cdot Q_{\text{step}}}^{(|\hat{x}|+1-f) \cdot Q_{\text{step}}} f_{uv}(x) dx & \hat{x} \neq 0 \\ 2 \cdot \int_0^{(1-f) \cdot Q_{\text{step}}} f_{uv}(x) dx & \hat{x} = 0 \end{cases} \quad (7)$$

where  $Q_{\text{step}}$  represents the quantization step size equaling to  $2^{\text{qbits}}/Q$ , and  $\hat{x} = 0$  is a special case in (7), because the size of quantization interval becomes  $2(1-f)Q_{\text{step}}$  for the deadzone shown in Fig. 4.

Although there is no closed-form solution for calculating the probability in (7), we can approximate  $P\{\hat{C}_{uv} = \hat{x}\}$  using the characteristics of the GGD function and the design rule of the quantizer in H.264/AVC. According to the first mean value theorem for integration [34], there exists  $x^*$  in the quantization interval satisfying

$$P\{\hat{C}_{uv} = \hat{x}\} = \begin{cases} Q_{\text{step}} f_{uv}(x^*) & \hat{x} \neq 0 \\ 2(1-f)Q_{\text{step}} f_{uv}(x^*) & \hat{x} = 0. \end{cases} \quad (8)$$

That is,  $f_{uv}(x^*)$  equals the average probability in the corresponding quantization interval, and  $\hat{x} = 0$  is a special case in (8), because the size of quantization interval becomes  $2(1-f)Q_{\text{step}}$ . Furthermore, in the Appendix, we prove that for the shape parameter  $\eta < 1$ , which is the most cases [26],  $x^*$  lies in the first half of the quantization interval, i.e.,  $x^* \in [(\hat{x}-f) \cdot Q_{\text{step}}, (\hat{x}-f) \cdot Q_{\text{step}} + 0.5 \cdot Q_{\text{step}}]$ . According to [32] and [33], the rounding parameter  $f$  is set as a constant below  $1/2$  or even adaptively adjusted during encoding to better locate the expectation value of  $\hat{C}_{uv}$  inside a quantization interval, and the quantized value  $\hat{x}$  also lies in the first half of the quantization interval. Therefore, we use  $\hat{x}$  to approximate  $x^*$  when  $\hat{x} \neq 0$ . For the particular case of  $\hat{x} \neq 0$ , we simply approximate  $x^*$  using  $f$  which is the rounding offset defined in (6). Then the probability of  $\hat{C}_{uv} = \hat{x}$  is approximated as

$$P\{\hat{C}_{uv} = \hat{x}\} \approx \begin{cases} Q_{\text{step}} \cdot f_{uv}(\hat{x} \cdot Q_{\text{step}}) & \hat{x} \neq 0 \\ 2(1-f)Q_{\text{step}} \cdot f_{uv}(f \cdot Q_{\text{step}}) & \hat{x} = 0. \end{cases} \quad (9)$$

With (2a), (5), and (9), for  $\hat{x} \neq 0$ ,  $r_{uv}$  is approximated as

$$\begin{aligned} r_{uv} &\approx -\log_2 \left\{ Q_{\text{step}} \cdot \frac{\eta_{uv} \alpha_{uv}(\eta_{uv})}{2\sigma_{uv} \Gamma(1/\eta_{uv})} \exp \left\{ - \left[ \alpha_{uv}(\eta_{uv}) \left| \frac{Q_{\text{step}} \cdot \hat{x}}{\sigma_{uv}} \right| \right]^{\eta_{uv}} \right\} \right\} \\ &= -\log_2 \left\{ Q_{\text{step}} \cdot \frac{\eta_{uv} \alpha_{uv}(\eta_{uv})}{2\sigma_{uv} \Gamma(1/\eta_{uv})} \right\} \\ &\quad - \log_2 \left\{ \exp \left\{ - \left[ \alpha_{uv}(\eta_{uv}) \left| \frac{Q_{\text{step}} \cdot \hat{x}}{\sigma_{uv}} \right| \right]^{\eta_{uv}} \right\} \right\}, \\ &= \log_2(e) \cdot \left[ \alpha_{uv}(\eta_{uv}) \left| \frac{Q_{\text{step}}}{\sigma_{uv}} \right| \right]^{\eta_{uv}} \cdot |\hat{x}|^{\eta_{uv}} \\ &\quad - \log_2 \left\{ Q_{\text{step}} \cdot \frac{\eta_{uv} \alpha_{uv}(\eta_{uv})}{2\sigma_{uv} \Gamma(1/\eta_{uv})} \right\} \\ &= a_{uv} \cdot |\hat{x}|^{\eta_{uv}} + b_{uv} \end{aligned} \quad (10)$$

where

$$\begin{cases} a_{uv} = \log_2(e) \cdot [Q_{\text{step}} \cdot \alpha(\eta_{uv})/\sigma_{uv}]^{\eta_{uv}} \\ b_{uv} = -\log_2[Q_{\text{step}} \cdot \eta_{uv} \alpha(\eta_{uv})/(2\sigma_{uv} \Gamma(1/\eta_{uv}))]. \end{cases} \quad (11)$$

For the case of  $\hat{x} = 0$ ,  $r_{uv}$  is approximated as

$$\begin{aligned} r_{uv} &\approx -\log_2 \\ &\quad \left\{ 2(1-f)Q_{\text{step}} \cdot \frac{\eta_{uv} \alpha_{uv}(\eta_{uv})}{2\sigma_{uv} \Gamma(1/\eta_{uv})} \right. \\ &\quad \left. \exp \left\{ - \left[ \alpha_{uv}(\eta_{uv}) \left| \frac{Q_{\text{step}} \cdot f}{\sigma_{uv}} \right| \right]^{\eta_{uv}} \right\} \right\} \\ &= \log_2(e) \cdot \left[ \alpha_{uv}(\eta_{uv}) \left| \frac{Q_{\text{step}} \cdot f}{\sigma_{uv}} \right| \right]^{\eta_{uv}} \cdot f^{\eta_{uv}} \\ &\quad - \log_2 \left\{ 2(1-f)Q_{\text{step}} \cdot \frac{\eta_{uv} \alpha_{uv}(\eta_{uv})}{2\sigma_{uv} \Gamma(1/\eta_{uv})} \right\} \\ &= a_{uv}^0 \cdot f^{\eta_{uv}} + b_{uv}^0 \end{aligned} \quad (12)$$

where

$$\begin{cases} a_{uv}^0 = \log_2(e) \cdot [Q_{\text{step}} \cdot \alpha(\eta_{uv})/\sigma_{uv}]^{\eta_{uv}} \\ b_{uv}^0 = -\log_2[2(1-f)Q_{\text{step}} \cdot \eta_{uv} \alpha(\eta_{uv})/(2\sigma_{uv} \Gamma(1/\eta_{uv}))]. \end{cases} \quad (13)$$

Therefore, for a transform coefficient with Generalized Gaussian distributions, the self-information is a power function of its quantized magnitude, and the power item is exactly the shape parameter of the GGD function.

It is observed that for the case  $\eta_{uv} = 1$ , the GGD becomes a Laplacian distribution, and the power function in (10) and (12) becomes a linear function with  $a_{uv} = \log_2(e) \cdot [\sqrt{2} \cdot Q_{\text{step}}/\sigma_{uv}]$ . Because  $1/\sigma_{uv}$  is always larger with a larger index of  $u$  and  $v$ , it is shown that the high frequency coefficients with larger  $a_{uv}$  affect the bit rate more than the low frequency coefficients. If we denote  $a_{uv}$  in a matrix form as  $\mathbf{A}$ , i.e.,  $\mathbf{A}(u, v) = a_{uv}$ , it is then observed that  $\mathbf{A}$  is a scaling matrix, which emphasizes the high frequency coefficients. Furthermore, the effect of Subsection II-A that coefficients with different indexes affect the bit rate differently is also verified by this result.

#### D. Proposed Rate Estimator and Comparison With Other Estimators

If the components of a quantized transform block are independently distributed, the self-information of a quantized transform block is equal to the sum of self-information of all its components. According to [36] and [37], the above conditions are satisfied to a great extent and have been successfully utilized in [38] for establishing a source model for transform video coder. Due to the high correlation between the

actual entropy coding bits and the corresponding block self-information, we propose to use the sum of coefficient self-information as a novel block-level rate estimator. Therefore, with (10) and (12), our proposed block-level rate estimator is formulated as

$$r_B \approx \sum_u \sum_v r_{uv} \quad \text{where } r_{uv} \approx \begin{cases} a_{uv} \cdot |\hat{x}_{uv}|^{\eta_{uv}} + b_{uv} & \hat{x}_{uv} \neq 0 \\ a_{uv}^0 \cdot f^{\eta_{uv}} + b_{uv}^0 & \hat{x}_{uv} = 0 \end{cases} \quad (14a)$$

$$\hat{R}_B = \alpha \cdot r_B + \beta \quad (14b)$$

where  $f$  denotes the quantization offset defined in (6),  $r_B$  represents the symbol which approximates the self-information of the quantized transform block,  $\hat{R}_B$  represents the estimated coding bits,  $\alpha$  and  $\beta$  are model parameters. Due to the varying spatial frame statistics and different entropy coding methods, the proposed self-information  $r_B$  in (14a) is improper to be directly used to estimate the entropy coding bits. However, as shown in Fig. 5(c), the proposed self-information  $r_B$  is verified to be strongly linearly correlated with the actual entropy coding bits. The two model parameters are designed to set up the linear mapping relationship between the self-information and the actual entropy coding bits. These two model parameters are adjusted during the mode decision process to make the rate estimation model adaptive to different entropy coding methods and variously changing video contexts.

In previous works, the number of nonzero coefficients and the  $l_1$ -norm of the quantized transform coefficients [15]–[17] are two frequently-used symbols for rate estimation. In Fig. 5, in order to provide intuitive evaluations on the effectiveness of these symbols, we draw a large number of samples from the actual mode decision process for each of the symbols. From Fig. 5 it is observed that the samples of the estimated coding bits using proposed self-information (14a) are more compactly and linearly distributed compared with the other two symbols. It means that our proposed rate estimator provides higher level of confidence when used to estimate the entropy coding bit rate.

### III. IMPLEMENTATION OF PROPOSED RATE ESTIMATION ALGORITHM

In this section, we introduce the detail implementation of the proposed rate estimation algorithm including three main modules: rate estimation module, estimation of the distribution parameters, initialization and updating of model parameters.

#### A. Implementation of Proposed Rate Estimation Algorithm

To integrate the proposed algorithm with the actual mode decision process, we replace the original entropy coding with the proposed rate estimation module. The input of the rate estimation module is the quantized transform block and the output is the estimated coding bits.

To avoid the frequent time-consuming calculation of the power function in (14a), we accelerate the proposed algorithm by using look-up tables. The values of self-information for most input cases are pre-calculated to avoid redundant computation in the mode decision process. For example, before

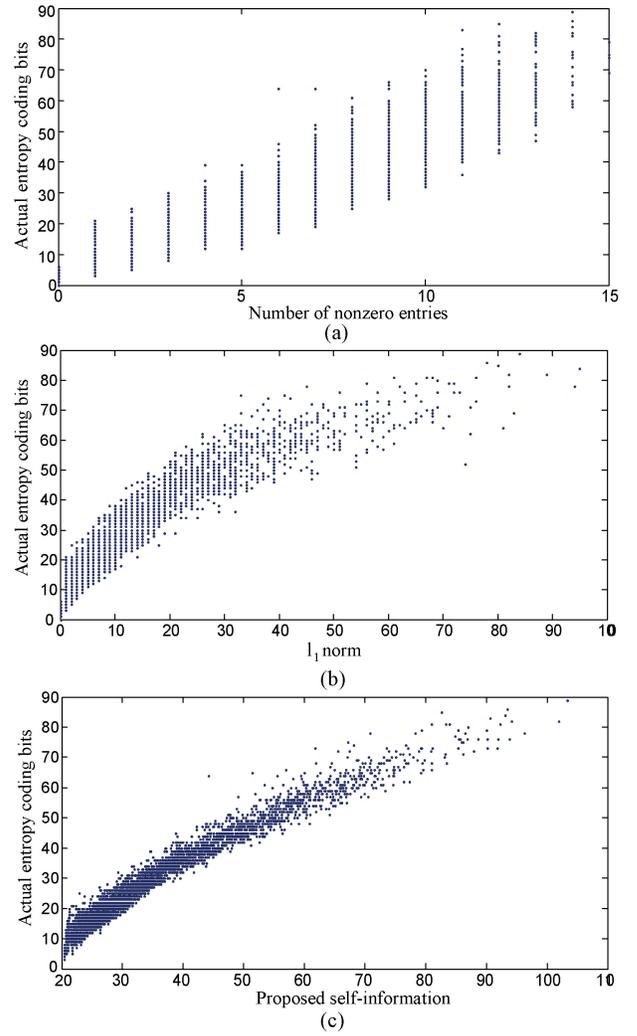


Fig. 5. Efficiency comparisons of different bit-rate estimators in *Football* with CIF format. (a) Number of nonzero coefficients versus actual entropy coding bits. (b)  $l_1$ -norm versus actual entropy coding bits. (c) Proposed self-information versus actual entropy coding bits.

encoding the current frame, the self-information for all the 16 quantized transform coefficients with magnitudes lower than a given threshold value 200 are pre-calculated. Then a  $16 \times 200$  lookup table is generated for the input quantized transform coefficient values below the threshold. The threshold value is empirically set as 200, because it covers most of the possible input cases. However, if the input quantized transform coefficient exceeds the threshold value, which is a rare case in the coding process, the calculation of (14a) becomes necessary. Therefore, to calculate the self-information of current transform block, we simply look up the table for each quantized transform coefficient and sum up all the 16 values to get  $r_B$  in (14a). Then the linear function in (14b) is utilized to map the self-information  $r_B$  to the final estimated entropy coding bits.

For the particular case of the I16MB mode [7], there is one  $4 \times 4$  direct coefficient (DC) transform block and 16  $4 \times 4$  alternating coefficient (AC) transform blocks without the DC coefficients. Therefore, the numbers of encoded coefficients in the AC transform block and the ordinary  $4 \times 4$  transform

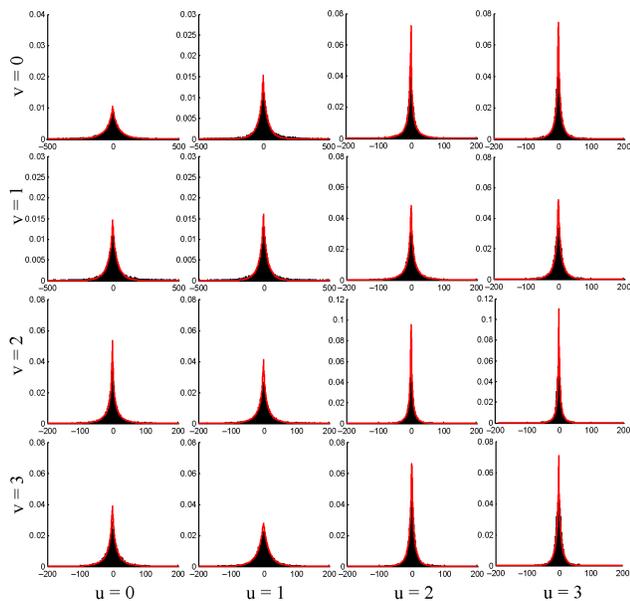


Fig. 6. Normalized histogram of the transform coefficients and corresponding discretized PDF of GGD with estimated distribution parameters.

block are different. To handle this specific issue, two different sets of model parameters are used for the  $4 \times 4$  DC transform block and the  $16 \ 4 \times 4$  AC transform blocks, respectively.

### B. Estimation of GGD Parameters

We utilize the method of [30] introduced in Section II-B to estimate the GGD parameters. In Fig. 6, the actual distribution of transform coefficients and the corresponding discretized GGD function with the estimated parameters  $\eta$  and  $\sigma$  are shown. From Fig. 6 it is observed that the estimated GGD parameters are efficient for simulating the distribution of transform coefficients.

Note that the GGD parameters of current frame become available only when the entire frame has been coded, and we propose to use the GGD parameters of previous frame of the same type for current frame. For example, at the beginning of coding the  $i$ th P frame, the GGD distribution parameters are initialized with the values obtained from previous P frame.

### C. Initialization and Updating of Model Parameters

In our simulations, the model parameter  $\alpha$  is initialized with a default value 1, which can be observed from the slope of the samples presented in Fig. 5(c). To initialize  $\beta$ , a prior knowledge that a zero block will probably be coded using only one bit is used. The self-information  $r^0_B$  of a zero block in (14a) is first calculated, after that, with the initialized alpha and  $r^0_B$ , a good initialization value for beta can be obtained by solving (14b).

Linear regression analysis is used to update the model parameters in our proposed algorithm. The linear regression process is formulated as follows:

$$R_i = \alpha r_i + \beta + \varepsilon_i \quad (15)$$

where  $i = 0, 1, \dots, n-1$ ,  $R_i$  represents the actual entropy coding bits,  $r_i$  represents the self-information calculated with

(14a),  $\varepsilon_i$  denotes the prediction error of the  $i$ th block. Then the least-mean-square-error estimates of  $\alpha$  and  $\beta$  are given by

$$\alpha = \frac{n \sum_i R_i r_i - \sum_i R_i \sum_i r_i}{n(\sum_i r_i)^2 - \sum_i r_i^2}$$

$$\beta = \frac{n(\sum_i r_i)^2 \sum_i R_i - \sum_i R_i r_i \sum_i r_i}{n(\sum_i r_i)^2 - \sum_i r_i^2}. \quad (16)$$

Two practical issues need to be handled for updating the mode parameters. First, as the number of data pairs of  $R_i$  and  $r_i$  utilized for updating the model parameters increases, the contributions of subsequent data pairs gradually decrease. As a result, the adaptivity of the model parameters becomes low when the number of input data pairs becomes large. To handle this issue, we set an updating period for this linear regression process. When the number of data pairs has reached the given updating period  $T_{\text{update}}$ , all the parameters used in linear regression are initialized with zero, and the former data pairs will not contribute to the subsequent updating process. Second, it is noted that when only a small number like 3 or 5 of data pairs are available after the initialization of the linear regression progress, the calculated model parameters could be unstable, and the prediction error will be large for the subsequent data pairs. To overcome this problem, we propose to control the linear regression process with an updating threshold. The updating of model parameters will not start until the number of available data pairs has reached  $T_{\text{threshold}}$ . With this  $T_{\text{threshold}}$ , the fitted model parameters become stable and trustable to be utilized for the rate estimation process. The  $T_{\text{update}}$  and  $T_{\text{threshold}}$  are empirically obtained and good performance is obtained by setting  $T_{\text{update}} = 100$ ,  $T_{\text{threshold}} = 15$  in our simulations.

## IV. PROPOSED DISTORTION ESTIMATION ALGORITHM

To estimate the distortion for complexity-reduced mode decision, in this section, we first analyze the origin of the distortion in H.264/AVC video coding thoroughly, then we develop a direct mathematical relationship between the discarded lower bits in quantization and the resulting distortion. Based on this investigation, a novel block-level distortion estimation algorithm is proposed.

### A. Origin of Distortion in H.264/AVC Video Coding

Distortion in a lossy video coding system is often measured by the sum of squared spatial difference between the reference and the reconstructed version. Besides the quantization process, we will show that the origin of distortion in H.264/AVC also includes the transform, inverse transform, and reconstruction processes. The forward transform and inverse transform for a  $4 \times 4$  residual block in H.264/AVC are formulated [39] as follows:

$$T(X) = C_f X C_f^T \otimes E_f \quad T^{-1}(Y) = C_i^T (Y \otimes E_i) C_i \quad (17)$$

where  $T(\cdot)$  and  $T^{-1}(\cdot)$  represent the forward transform and inverse transform, respectively,  $C_f$  and  $C_i$  are called the core matrices with integer entries,  $E_f$  and  $E_i$  are called the scaling matrices absorbed in the quantization process and the symbol

$\otimes$  indicates the element-by-element multiplication. Both  $T(\cdot)$  and  $T^{-1}(\cdot)$  are orthogonal transforms [40], and they satisfy that

$$\|T(X)\|^2 = \|T^{-1}(X)\|^2 = \|X\|^2 \quad (18)$$

where  $\|\cdot\|^2$  indicates the  $l_2$ -norm of a matrix. However, the actual forward scaling matrix  $E_f$  adopted in the reference software [42] is slightly modified from  $E_f$  due to a resulting improvement of perceptual quality at the decoder [40]. Therefore, the orthogonality in (18) is not strictly satisfied and the forward transform  $T(\cdot)$  also contribute to the spatial-domain distortion. For example, an actual  $4 \times 4$  residual block obtained from the encoding process of *Bus* with CIF format is

$$\begin{bmatrix} -11 & -2 & 4 & -1 \\ -8 & -7 & 8 & 6 \\ 4 & -10 & -9 & 3 \\ -1 & -6 & -11 & -6 \end{bmatrix}$$

then the H.264/AVC forward transform and quantization are sequentially performed on this block with the theoretical scaling matrix in (17) and actual scaling matrix in the reference software, respectively. For the case of  $QP = 18$ , the two scaling matrices are

$$\begin{bmatrix} 0.2500 & 0.1581 & 0.2500 & 0.1581 \\ 0.1581 & 0.1000 & 0.1581 & 0.1000 \\ 0.2500 & 0.1581 & 0.2500 & 0.1581 \\ 0.1581 & 0.1000 & 0.1581 & 0.1000 \end{bmatrix}$$

$$\begin{bmatrix} 0.2500 & 0.1538 & 0.2500 & 0.1538 \\ 0.1538 & 0.1000 & 0.1538 & 0.1000 \\ 0.2500 & 0.1538 & 0.2500 & 0.1538 \\ 0.1538 & 0.1000 & 0.1538 & 0.1000 \end{bmatrix}.$$

The outputs of the quantization process using the above two different scaling matrices are

$$\begin{bmatrix} -2 & -2 & 1 & 2 \\ 1 & -2 & -2 & 1 \\ -1 & 1 & -1 & -1 \\ 2 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} -2 & -1 & 1 & 1 \\ 1 & -2 & -2 & 1 \\ -1 & 1 & -1 & -1 \\ 2 & 1 & 1 & 0 \end{bmatrix}$$

respectively. It is noted that these two quantized blocks are different, i.e., the forward transform using the actual scaling matrix results in additional distortion. Let  $D_f$  be the part of distortion caused by forward transform, then based on (18) and the linearity of  $T(\cdot)$ , we obtain

$$D_f = \|(E_f - E'_f) \otimes C_f X C_f^T\|^2 \quad (19)$$

where  $X$  is the residual block,  $E_f$  and  $E'_f$  are the theoretical and the actual forward scaling matrices used in H.264/AVC reference software, respectively.

In the inverse transform  $T^{-1}(\cdot)$ , the multiplication operations with the  $C_i$  composed of  $1/2$  and  $-1/2$  need rounding operations which also result in slight distortion, and  $D_i$  is defined to represent this part of distortion. After the sequential processing of transform, quantization, inverse quantization, and inverse transform, the reconstructed pixel value may exceed the dynamic range of the 8-bit pixel intensity representation, and another clipping operation is needed to restrict

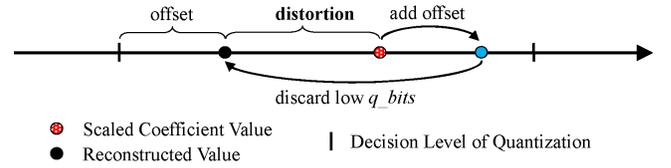


Fig. 7. Quantization process of a single transform coefficient in H.264/AVC video coding.

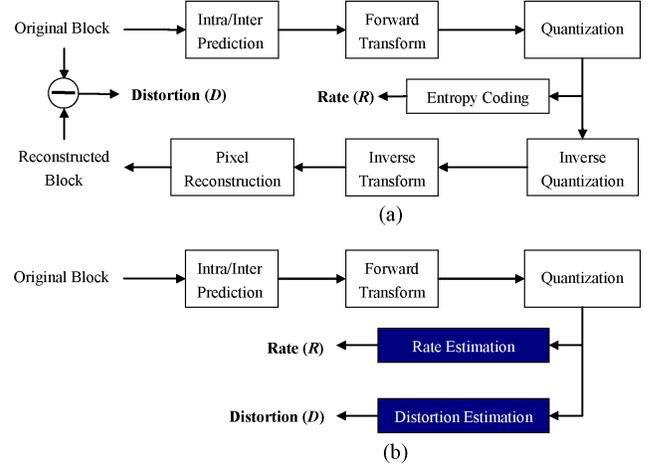


Fig. 8. Implementation schemes of the original and the proposed rate-distortion calculation. (a) Original. (b) Proposed.

the reconstructed pixel value to be lying in the range of 0 and 255. The distortion resulted from this clipping operation is denoted by  $D_r$ .

Finally, let the distortion resulting from quantization be  $D_q$ , then the total distortion  $D$  in H.264/AVC video coding is caused by the contribution of four independent parts  $D_f$ ,  $D_i$ ,  $D_r$ , and  $D_q$ . But  $D$  is not equal to the sum of  $D_f$ ,  $D_i$ ,  $D_r$ , and  $D_q$  due to the possible counteraction between these four parts. Empirical results show that  $D_q$  takes the majority part of  $D$ , and we use  $D_q$  as an estimation of  $D$  for better complexity-performance tradeoff.

### B. Proposed Block-Level Distortion Estimation Algorithm for H.264/AVC Video Coding

It is observed that the only loss in the quantization process for the post-scaled transform coefficients is the low  $q$  bits, which are discarded by the rounding operation. It means that the discarded low  $q$  bits determine the corresponding distortion  $D_q$ . To make the relationship clear, the process of quantizing a single post-scaled transform coefficient is illustrated in Fig. 7. First, the *scaled\_coeff* represented by the red point is shifted to the blue point by adding the *offset*, and then by discarding the low  $q$  bits with the rounding operation, the blue point which falls into a quantization interval is moved to the black point representing *recon\_coeff*. The final distortion  $D_q$  is then measured by the distance between *scaled\_coeff* and *recon\_coeff*, and the distance is calculated by  $d_q = |\text{offset} - \text{low}_q\text{bits}|$ . With  $d_q$ , the corresponding distortion in transform domain is calculated as

$$D_q = [(d_q/2^q - \text{bits}) \cdot Q_{\text{step}}]^2. \quad (20)$$

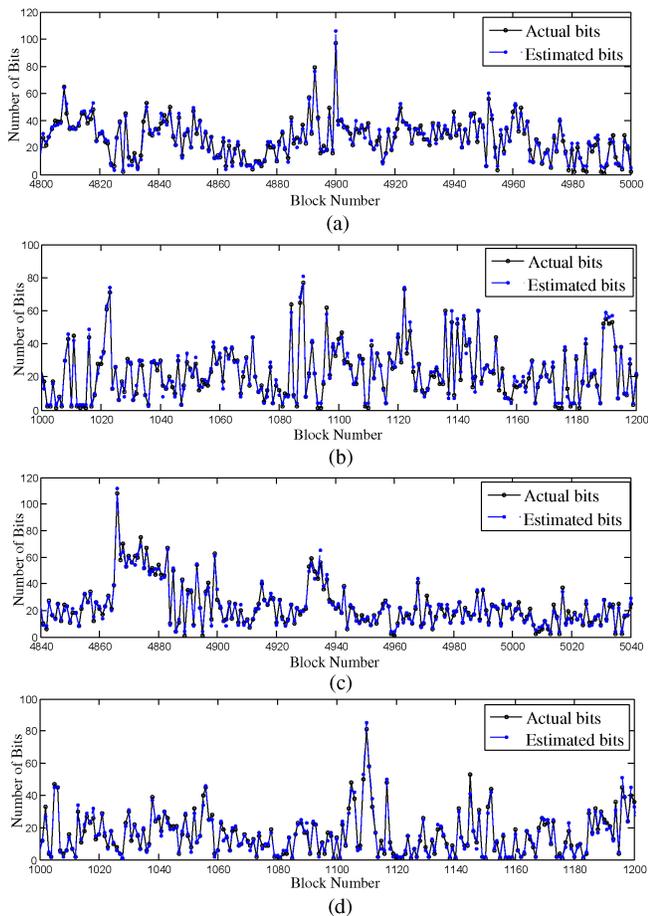


Fig. 9. Estimated and actual coding bits of randomly selected 200 blocks of different sequences with CIF format. (a) *Tempete* (CABAC). (b) *Football* (CABAC). (c) *Bus* (CAVLC). (d) *Carphone* (CAVLC).

With (20), the distortion of a single transform coefficient with index  $(u, v)$  is then approximated by the distortion caused by quantization as

$$D_{uv} \approx [(\text{offset}_{uv} - \text{low\_qbits}_{uv}) / 2^{q\_bits}] \cdot Q_{\text{step}}]^2 \quad (21)$$

and the distortion of a transform block in spatial domain is estimated as

$$D \approx \sum_u \sum_v D_{uv} = \sum_u \sum_v \left[ \left( \frac{|\text{offset} - \text{low\_qbits}_{uv}|}{2^{q\_bits}} \right) \cdot Q_{\text{step}} \right]^2 \quad (22)$$

To save the number of multiplication operations, we manipulate the function in (22) as

$$D \approx [d \cdot Q_{\text{step}}^2 + r] \gg (2 \cdot \text{qbits}) \quad (23)$$

where  $d = \sum_u \sum_v |\text{offset} - \text{low\_qbits}_{uv}|^2$ , and  $r$  is a rounding factor which equals  $2^{2 \cdot \text{qbits} - 1}$ . Finally, a new distortion estimation method is obtained by building a bridge between the discarded bits and the corresponding distortion in (23). The proposed distortion estimation algorithm in (23) can be easily integrated with the quantization module without any additional mathematical manipulations. The original 16 multiplications in the inverse quantization process for a  $4 \times 4$  transform block

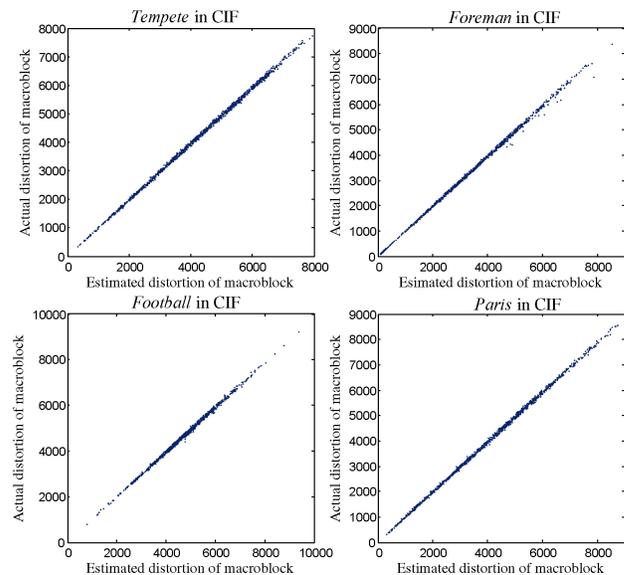


Fig. 10. Estimated and actual distortion of 4000 macroblocks for sequences with CIF format.

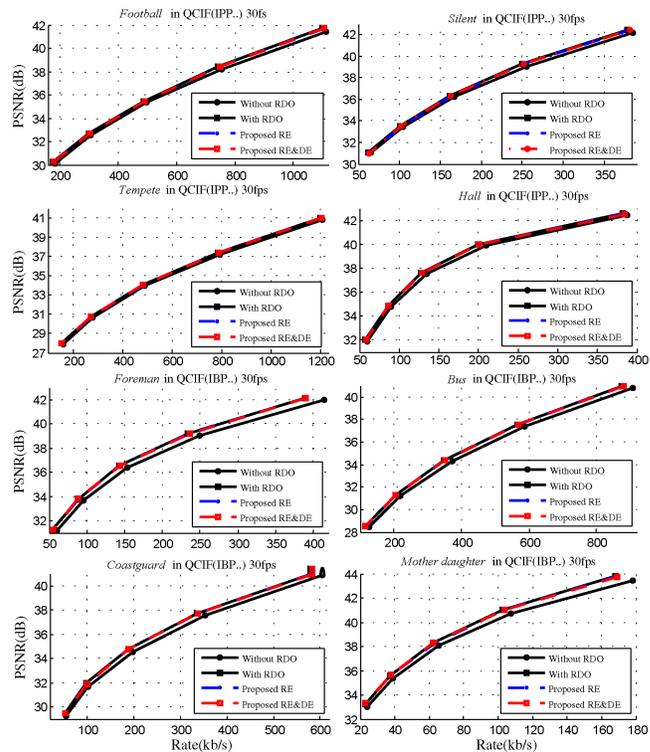


Fig. 11. Performance comparisons of different mode decision algorithms for sequences with QCIF format and CABAC entropy coding method.

are reduced to a single multiplication with  $Q_{\text{step}}^2$  in (23), which can be alternatively absorbed in the  $\lambda$  in (1).

The proposed distortion estimation algorithm can cooperate with the proposed rate estimation algorithm to save more computation time of the overall encoding process. The difference between the original and the proposed implementation structure of the rate-distortion calculation is illustrated in Fig. 8. The major difference is that the entropy coding is

replaced with the low-complexity rate estimation module, and the inverse quantization, inverse transform and pixel reconstruction are replaced with a simple distortion estimation module.

## V. EXPERIMENTAL RESULTS

To validate the accuracy and efficiency of proposed block-level rate and distortion estimation algorithms, we integrated our proposed algorithms to the H.264/AVC reference software JM12.0 [42]. In this section, we design three experiments: 1) comparison between estimated values and the corresponding actual values; 2) the complexity reduction and the coding performance compared with the conventional RDO; and 3) Comparisons with Tu *et al.*'s rate-distortion estimation algorithm in [16].

### A. Comparison Between Estimated and Actual Values

In this subsection, comparisons of the estimated and actual values are made for different sequences under different quantization parameters (QPs) with both CAVLC and CABAC entropy coding methods. In Fig. 9, the estimated and actual coding bits of randomly selected 200 blocks for each test are shown. We see from Fig. 9 that the proposed rate estimation algorithm is accurate and robust for different video contexts and different coding configurations, and the estimated bits are closely matched with the actual coding bits for both low-bits and high-bits blocks. To validate the accuracy of the proposed distortion estimation algorithm, the estimated and the actual distortion values of 4000 macroblocks for different sequences with CIF format are shown in Fig. 10, and it is observed that our algorithm outputs accurate distortion estimations.

Actually, the mode decision process presents a certain level of tolerance for the accuracy of rate and distortion estimations, because the only issue that matters the result of mode decision is the relative order of the R-D cost values of candidate modes. Therefore, small prediction errors do not affect the mode decision result until it is large enough to change the relative order of R-D cost values.

### B. Performance of the Proposed Block-Level Rate and Distortion Estimation Algorithms

In this experiment, we compare the coding performance and complexity between our proposed low-complexity mode decision algorithm and original RDO in JM12.0. When comparing the coding performance difference, we utilize the popular method proposed in [41] for calculating the average peak signal-to-noise ratio (PSNR) differences between R-D curves. When evaluating the complexity reduction, we calculate  $\Delta T$  in [16] and [19] defined as

$$\Delta T = \frac{T_{\text{pro\_RDO}} - T_{\text{org\_RDO}}}{T_{\text{org\_RDO}}} \times 100\% \quad (24)$$

where  $T_{\text{org\_RDO}}$  and  $T_{\text{pro\_RDO}}$  indicate the coding time with the original implementation and with the proposed algorithm, respectively. When comparing  $\Delta T$ , we indicate the total encoding time including the non-RDO processes. Some important encoding configurations are set as follows: all available

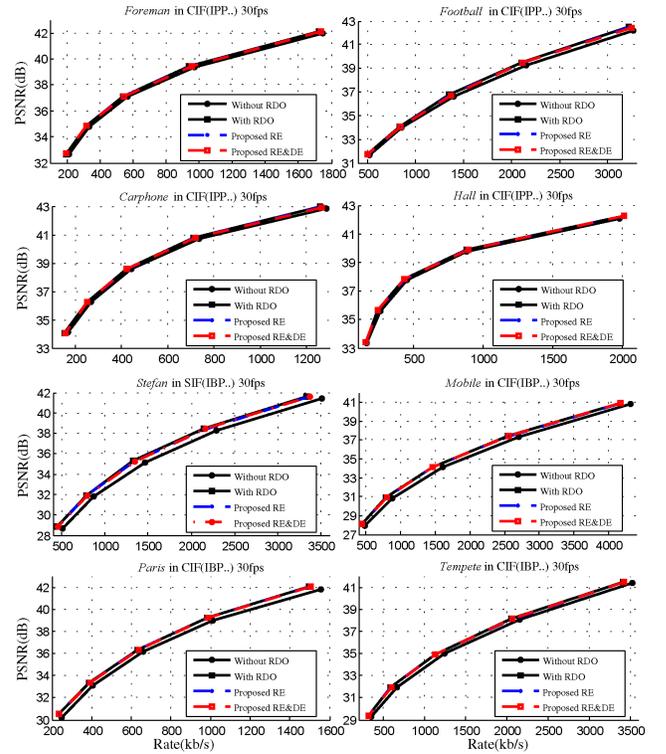


Fig. 12. Performance comparisons of different mode decision algorithms for sequences with CIF format and CABAC entropy coding method.

Inter and Intra modes for both Inter frame and Intra frame are enabled; fast motion estimation algorithm is used; motion search range is 33 by 33; the number of reference frames is 1 for IPP coding type and 2 for IBP coding type; Intra Period is set as 8; the frame rate is 30frames/s for both CIF and quarter common intermediate format (QCIF) sequences; FastCrIntraDecision is disabled.

To verify the robustness of the proposed rate-distortion estimation algorithm, extensive experiments were performed on standard sequences with QCIF and CIF formats. For both CIF and QCIF sequences, we employ two different kinds of mode-decision strategies. One is the original RDO with only the proposed rate estimation algorithm (proposed RE), and the other is the original RDO using both the proposed rate and distortion algorithms (proposed RE and DE). The experimental results for CIF and QCIF sequences are shown in Tables I and II, respectively. The experimental results tabulated in Tables I and II show that the average PSNR loss compared with the original RDO is ignorable, while about average 32% of the total encoding time is saved for both CABAC and CAVLC. The R-D performance comparisons of the proposed algorithm to the default RDO are also shown in Fig. 11 and Fig. 12, and it is seen that, the PSNR loss with the proposed algorithm over the full range of QP values is ignorable.

### C. Comparisons With Tu *et al.*'s Rate-Distortion Estimation Algorithm

In this experiment, the proposed algorithms are compared with Tu *et al.*'s rate and distortion estimation algorithms

TABLE I  
PERFORMANCE OF THE PROPOSED ALGORITHMS (COMPARED WITH THE ORIGINAL RATE-DISTORTION OPTIMIZATION TECHNIQUE) FOR CIF  
SEQUENCES AT 30 FRAMES/S

Entropy Coding Sequences		CABAC				CAVLC			
		$\Delta PSNR^*$	$\Delta T_{Total}^*$	$\Delta PSNR^{**}$	$\Delta T_{Total}^{**}$	$\Delta PSNR^*$	$\Delta T_{Total}^{**}$	$\Delta PSNR^*$	$\Delta T_{Total}^{**}$
<i>Stefan</i> (352 × 240)	IPP..	-0.0382	-34.8%	-0.0494	-50.7%	-0.0551	-33.9%	-0.0618	-38.2%
	IBP..	-0.0539	-29.6%	-0.0811	-32.6%	-0.0682	-27.4%	-0.0848	-31.2%
<i>Coastguard</i>	IPP..	-0.0341	-32.6%	-0.0388	-35.2%	-0.0417	-33.9%	-0.0485	-37.8%
	IBP..	-0.0361	-26.0%	-0.0395	-30.5%	-0.0431	-26.0%	-0.0499	-28.7%
<i>Bus</i>	IPP..	-0.0350	-33.9%	-0.0444	-36.4%	-0.0500	-34.5%	-0.0549	-38.3%
	IBP..	-0.0324	-27.3%	-0.0382	-31.2%	-0.0492	-26.7%	-0.0519	-30.1%
<i>Foreman</i>	IPP..	-0.0363	-24.6%	-0.0447	-27.1%	-0.0386	-27.4%	-0.0493	-31.5%
	IBP..	-0.0345	-11.5%	-0.0385	-14.1%	-0.0286	-21.6%	-0.0429	-24.9%
<i>Football</i>	IPP..	-0.0491	-25.3%	-0.0530	-35.9%	-0.0643	-28.6%	-0.0714	-40.7%
	IBP..	-0.0455	-20.2%	-0.0596	-25.8%	-0.0641	-20.8%	-0.0730	-24.7%
<i>Mobile</i>	IPP..	-0.0293	-37.7%	-0.0366	-40.3%	-0.0473	-37.3%	-0.0565	-41.6%
	IBP..	-0.0277	-33.9%	-0.0379	-37.9%	-0.0454	-25.4%	-0.0544	-28.9%
<i>Paris</i>	IPP..	-0.0387	-33.1%	-0.0427	-35.5%	-0.0714	-33.9%	-0.0730	-38.2%
	IBP..	-0.0414	-28.0%	-0.0445	-33.4%	-0.0571	-28.8%	-0.0610	-32.4%
<i>Tempeete</i>	IPP..	-0.0367	-46.8%	-0.0381	-48.3%	-0.0525	-34.6%	-0.0592	-38.8%
	IBP..	-0.0396	-39.7%	-0.0397	-43.1%	-0.0614	-27.3%	-0.0629	-30.5%
<i>Flower</i>	IPP..	-0.0441	-40.8%	-0.0522	-42.1%	-0.0835	-36.4%	-0.0928	-40.2%
	IBP..	-0.0317	-56.1%	-0.0383	-26.2%	-0.0664	-28.3%	-0.0726	-31.9%
<i>Akiyo</i>	IPP..	-0.0328	-19.0%	-0.0493	-22.3%	-0.0391	-23.9%	-0.0479	-28.8%
	IBP..	-0.0256	-14.1%	-0.0395	-36.2%	-0.0006	-16.6%	-0.0218	-21.0%
<i>Carphone</i>	IPP..	-0.0392	-20.5%	-0.0468	-23.0%	-0.0306	-26.4%	-0.0426	-30.8%
	IBP..	-0.0429	-18.1%	-0.0433	-22.7%	-0.0471	-20.3%	-0.0477	-23.4%
<i>Container</i>	IPP..	-0.0416	-27.5%	-0.0522	-29.5%	-0.0468	-30.8%	-0.0645	-35.5%
	IBP..	-0.0379	-22.0%	-0.0427	-27.5%	-0.0422	-23.1%	-0.0584	-26.1%
<i>Hall</i>	IPP..	-0.0167	-21.6%	-0.0328	-24.8%	-0.0237	-27.0%	-0.0386	-31.3%
	IBP..	-0.0182	-20.3%	-0.0242	-25.5%	-0.0304	-20.8%	-0.0339	-24.9%
<i>Silent</i>	IPP..	-0.0436	-27.2%	-0.0578	-29.2%	-0.0713	-29.6%	-0.0810	-34.3%
	IBP..	-0.0426	-24.8%	-0.0611	-38.1%	-0.0583	-24.7%	-0.0763	-27.8%
Average	IPP..	-0.0368	-30.4%	-0.0456	-34.3%	-0.0511	-31.3%	-0.0601	-36.2%
	IBP..	-0.0364	-26.6%	-0.0449	-30.4%	-0.0473	-24.1%	-0.0565	-27.6%

\*Only rate estimation enabled.

\*\*Both rate and distortion estimations enabled.

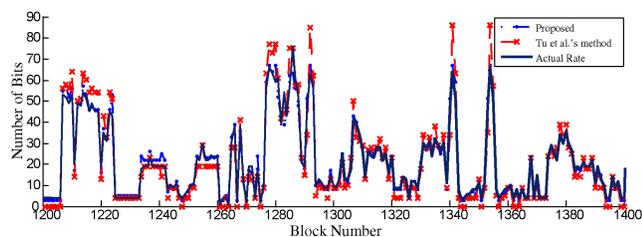


Fig. 13. Comparison of our proposed method with Tu *et al.*'s method in [16] for the rate estimation.

described in [16]. For the rate estimation, the method in [16] is designed for inter-mode decision and Intra modes are disabled for coding Inter frames in the experiments. In our experiments, Tu *et al.*'s algorithm is implemented and it works efficiently when only the Inter modes are enabled. However, for the intra-mode decision, the rate estimation becomes a more challenging task due to the high energy of

the residual. Fig. 13 shows the actual coding bits of 200 actual blocks and the corresponding estimated values of both algorithms, and it is shown that the predicted values of our proposed algorithm are closer to the actual values.

For the distortion estimation, the first 20 actual distortion values in the mode decision process of coding *Football* with CIF format and the corresponding estimated values of both algorithms are tabulated in Table III. Theoretically, the proposed algorithm outputs exactly the same estimated distortion values of the algorithm in [16], but the implementation is simpler and the inverse quantization and some additional mathematical manipulations needed in [16] are also saved. From Table III, it is seen that the actual estimated values of these two algorithms present slight differences which are caused by the different rounding operations for some additional mathematical manipulations needed in Tu *et al.*'s method.

To compare the coding performance and complexity reduction, all the available Inter and Intra modes are enabled, and both rate and distortion estimation algorithms are used

TABLE II

PERFORMANCE OF THE PROPOSED ALGORITHMS (COMPARED WITH THE ORIGINAL RATE-DISTORTION OPTIMIZATION TECHNIQUE) FOR QCIF SEQUENCES AT 30 FRAMES/S

Entropy Coding Sequences		CABAC				CAVLC			
		$\Delta PSNR^*$	$\Delta T_{Total}^*$	$\Delta PSNR^{**}$	$\Delta T_{Total}^{**}$	$\Delta PSNR^*$	$\Delta T_{Total}^*$	$\Delta PSNR^{**}$	$\Delta T_{Total}^{**}$
Foreman	IPP..	-0.0427	-27.8%	-0.0521	-32.5%	-0.0370	-29.2%	-0.0472	-33.4%
	IBP..	-0.0286	-22.9%	-0.0401	-26.2%	-0.0447	-23.4%	-0.0409	-27.3%
Bus	IPP..	-0.0319	-35.7%	-0.0298	-40.2%	-0.0581	-35.7%	-0.0653	-39.9%
	IBP..	-0.0400	-30.0%	-0.0444	-32.3%	-0.0486	-27.2%	-0.0583	-30.6%
Football	IPP..	-0.0444	-29.6%	-0.0443	-34.1%	-0.0595	-31.3%	-0.0658	-34.9%
	IBP..	-0.0443	-24.2%	-0.0503	-26.7%	-0.0651	-24.3%	-0.0640	-27.2%
Tempete	IPP..	-0.0287	-37.9%	-0.0341	-42.3%	-0.0600	-38.2%	-0.0649	-42.1%
	IBP..	-0.0368	-31.9%	-0.0339	-34.2%	-0.0556	-29.7%	-0.0617	-34.2%
Coastguard	IPP..	-0.0348	-31.9%	-0.0395	-36.7%	-0.0411	-34.5%	-0.0452	-37.9%
	IBP..	-0.0379	-25.8%	-0.0426	-28.8%	-0.0408	-26.7%	-0.0435	-29.2%
Container	IPP..	-0.0651	-28.8%	-0.0789	-32.8%	-0.0776	-30.7%	-0.0915	-35.0%
	IBP..	-0.0489	-24.0%	-0.0547	-27.6%	-0.0675	-24.4%	-0.0868	-29.7%
Hall	IPP..	-0.0378	-27.9%	-0.0511	-31.9%	-0.0498	-29.6%	-0.0688	-34.0%
	IBP..	-0.0076	-23.9%	-0.0170	-26.6%	-0.0589	-25.1%	-0.0576	-29.6%
Mother...daughter	IPP..	-0.0439	-23.4%	-0.0516	-27.6%	-0.0438	-26.4%	-0.0555	-31.7%
	IBP..	-0.0186	-19.0%	-0.0350	-22.2%	-0.0273	-20.8%	-0.0352	-25.6%
Silent	IPP..	-0.0585	-28.8%	-0.0652	-33.7%	-0.0823	-31.9%	-0.0954	-37.1%
	IBP..	-0.0321	-23.9%	-0.0465	-27.9%	-0.0631	-25.4%	-0.0635	-29.5%
Average	IPP..	-0.0431	-30.2%	-0.0496	-34.6%	-0.0566	-32.0%	-0.0666	-36.2%
	IBP..	-0.0327	-25.1%	-0.0405	-28.1%	-0.0524	-25.3%	-0.0568	-29.2%

\*Only rate estimation enabled.

\*\*Both rate and distortion estimations enabled.

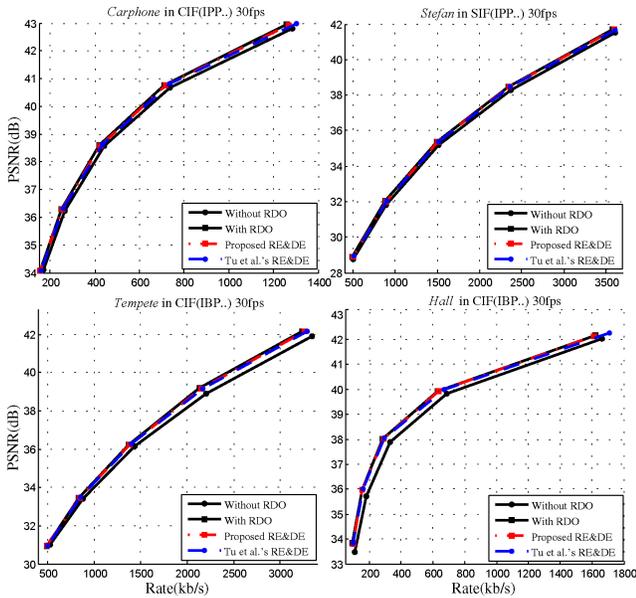


Fig. 14. Coding performance comparison with Tu *et al.*'s method in [16].

for mode decision. The experimental results for several standard test sequences are tabulated in Table IV, and the R-D performances are also shown in Fig. 14. From Table IV, it is shown that our proposed algorithm prevents more coding performance degradation, and the complexity reduction is also a little superior due to a simpler implementation of distortion

TABLE III

COMPARISON WITH TU *et al.*'s METHOD IN [16] FOR THE DISTORTION ESTIMATION

IPmode*	True Distortion	Tu <i>et al.</i> 's Method	Proposed
10 (2)	1821	1781	1781
9 (2)	87	81	82
9 (1)	106	107	108
9 (2)	104	103	104
9 (8)	122	123	123
9 (0)	86	89	90
9 (2)	108	95	96
9 (3)	138	137	137
9 (7)	93	86	87
9 (0)	158	156	156
9 (1)	190	178	178
9 (2)	166	171	172
9 (3)	179	159	160
9 (4)	203	190	191
9 (5)	111	117	117
9 (6)	171	166	165
9 (7)	172	156	155
9 (8)	150	137	139
9 (1)	101	87	87
9 (2)	115	106	105

\*IPmode indicates I4MB (9) or I16MB (10). The number in bracket denotes the prediction direction index (I4MB: 0-8, I16MB: 0-3).

estimation. The superior coding performance originates from the improved rate estimation algorithm, and the complexity reduction stems from the simplified distortion estimation. An enlarged version of the R-D curve in Fig. 14 is also shown in

TABLE IV  
CODING PERFORMANCE COMPARISON WITH TU *et al.*'S METHOD IN [16]

Sequences		Entropy Coding		CABAC				CAVLC			
				IPPPP		IBBPB		IPPPP		IBBPB	
		$\Delta PSNR$	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta T_{Total}$		
Carphone (CIF)	Tu <i>et al.</i> 's	-0.0889	-19.7%	-0.0755	-20.3%	-0.0995	-25.1%	-0.0969	-19.0%		
	Proposed	-0.0468	-23.0%	-0.0433	-22.7%	-0.0426	-30.8%	-0.0477	-23.4%		
Tempete (CIF)	Tu <i>et al.</i> 's	-0.1109	-44.4%	-0.0900	-42.1%	-0.1415	-33.6%	-0.1221	-26.3%		
	Proposed	-0.0381	-48.3%	-0.0397	-43.1%	-0.0592	-38.8%	-0.0629	-30.5%		
Hall (CIF)	Tu <i>et al.</i> 's	-0.0882	-20.6%	-0.0668	-36.4%	-0.1143	-26.3%	-0.0984	-20.1%		
	Proposed	-0.0328	-24.8%	-0.0242	-25.5%	-0.0386	-31.3%	-0.0339	-24.9%		
Coastguard (CIF)	Tu <i>et al.</i> 's	-0.1199	-31.4%	-0.1166	-27.5%	-0.1164	-32.9%	-0.1309	-24.6%		
	Proposed	-0.0395	-36.7%	-0.0426	-28.8%	-0.0485	-37.8%	-0.0499	-28.7%		
Mot_daught (QCIF)	Tu <i>et al.</i> 's	-0.1636	-23.9%	-0.1203	-19.2%	-0.1575	-26.4%	-0.1099	-20.4%		
	Proposed	-0.0516	-27.6%	-0.0350	-22.2%	-0.0555	-31.7%	-0.0352	-25.6%		
Football (QCIF)	Tu <i>et al.</i> 's	-0.0992	-29.9%	-0.0881	-24.5%	-0.1275	-29.4%	-0.1284	-23.8%		
	Proposed	-0.0443	-34.1%	-0.0503	-26.7%	-0.0658	-34.9%	-0.0640	-27.2%		
Foreman (QCIF)	Tu <i>et al.</i> 's	-0.1434	-29.1%	-0.1369	-23.7%	-0.1438	-28.0%	-0.1400	-23.6%		
	Proposed	-0.0521	-32.5%	-0.0401	-26.2%	-0.0472	-33.4%	-0.0409	-27.3%		

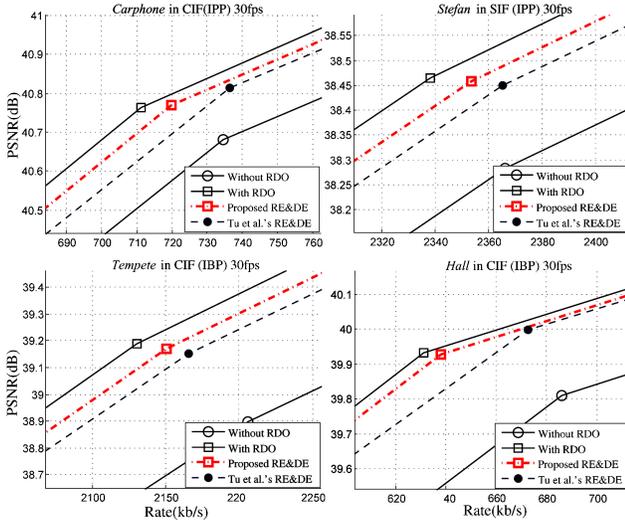


Fig. 15. Comparison of enlarged R-D curve in Fig.14.

Fig. 15, and it is observed that the R-D sample points of our method are more closely matched with the R-D sample points of the original RDO. From Fig. 14, it is also observed that for the cases of high bit-rate coding, the R-D sample points of our method are visibly closer with the R-D sample points of original RDO than Tu *et al.*'s method. This observation is also verified by the high bit-rate cases shown in Figs. 5 and 13.

## VI. CONCLUSION AND FUTURE WORK

In H.264/AVC, the RDO technique is important for choosing the optimal coding mode. However, the encoding complexity increases drastically because of the expensive calculation of R-D cost. In this paper, we propose novel rate and distortion estimation algorithms to accelerate the RDO process. For rate estimation, we utilize the flexible GGDs to model the transform coefficient and derive a block-level rate estimation algorithm, which employs the weighted sum of quantized

transform coefficients as an efficient rate estimator. For distortion estimation, the origin of distortion was first analyzed thoroughly, and then a novel TDD estimator was proposed with a compact implementation scheme. With proposed rate and distortion estimation algorithms, about 32% total encoding time is saved with ignorable coding performance degradation compared with conventional RDO. In the future work, efficient combination of R-D estimation and mode skipping will be studied to present more powerful fast mode decision algorithms.

## APPENDIX

In this appendix, a detailed derivation of the conclusion that  $x^*$  lies in the first half of the integration interval is given. First, we will prove that  $f_{uv}(x)$  is a convex function when  $x \geq 0$  and  $\eta_{uv} < 1$  by calculating its first derivative as

$$\begin{aligned}
 f'_{uv}(x) &= d \left\{ \frac{\eta_{uv}\alpha(\eta_{uv})}{2\sigma_{uv}\Gamma(1/\eta_{uv})} \exp \left\{ -[\alpha(\eta_{uv})\frac{x}{\sigma_{uv}}]^{\eta_{uv}} \right\} \right\} / dx \\
 &= -\frac{\eta_{uv}\alpha(\eta_{uv})}{2\sigma_{uv}\Gamma(1/\eta_{uv})} \cdot \exp \{-(vx)^{\eta_{uv}}\} \cdot \eta_{uv}v^{\eta_{uv}}x^{\eta_{uv}-1} \\
 &= -\kappa \cdot \exp \{-(Vx)^{\eta_{uv}}\} \cdot x^{\eta_{uv}-1}
 \end{aligned} \tag{25a}$$

where

$$v = \frac{\alpha(\eta_{uv})}{\sigma_{uv}} \quad \kappa = \frac{\eta_{uv}\alpha(\eta_{uv})}{2\sigma_{uv}\Gamma(1/\eta_{uv})} \cdot \eta_{uv} \cdot v^{\eta_{uv}}. \tag{25b}$$

With (25a) and (25b), the second derivative of  $f_{uv}(x)$  is then calculated as

$$\begin{aligned}
 f''_{uv}(x) &= d \left\{ -\kappa \cdot \exp \{-(vx)^{\eta_{uv}}\} \cdot x^{\eta_{uv}-1} \right\} / dx \\
 &= \kappa \{ \exp \{-(vx)^{\eta_{uv}}\} \eta_{uv}v^{\eta_{uv}}x^{2\eta_{uv}-2} \\
 &\quad + (1 - \eta_{uv}) \exp \{-(Vx)^{\eta_{uv}}\} \cdot x^{\eta_{uv}-2} \}
 \end{aligned} \tag{26}$$

It is observed that  $\kappa$  is positive-valued because  $\eta_{uv}$ ,  $\sigma_{uv}$  and  $\alpha(\eta_{uv})$  are all positive-valued, and  $(1 - \eta_{uv}) \exp \{-(vx)^{\eta_{uv}}\} \cdot x^{\eta_{uv}-2} > 0$  when  $\eta_{uv} < 1$ , then the second derivative of  $f_{uv}(x)$  is proved to be positive. Therefore, we obtain the conclusion

that  $f_{uv}(x)$  is a convex function when  $x \geq 0$  and  $\eta_{uv} < 1$ . Then the integration of  $f_{uv}(x)$  in a certain quantization interval  $[a, b]$  is reformulated as

$$\begin{aligned} \int_a^b f_{uv}(x)dx &= \int_a^{\frac{a+b}{2}} f_{uv}(x)dx + \int_{\frac{a+b}{2}}^b f_{uv}(x)dx \\ &= \int_a^{\frac{a+b}{2}} f_{uv}(x)dx + \int_{\frac{a+b}{2}}^a f_{uv}(a+b-x)d(a+b-x) \\ &= \int_a^{\frac{a+b}{2}} f_{uv}(x)dx + \int_a^{\frac{a+b}{2}} f_{uv}(a+b-x)dx \\ &= \int_a^{\frac{a+b}{2}} (f_{uv}(x) + f_{uv}(a+b-x))dx. \end{aligned} \quad (27)$$

Based on the derived conclusion that  $f_{uv}(x)$  is a convex function when  $x = 0$  and  $\eta_{uv} < 1$ , we have that

$$f_{uv}(x) + f_{uv}(a+b-x) > 2f_{uv}\left(\frac{a+b}{2}\right). \quad (28)$$

Because  $f_{uv}(x^*)$  in (8) represents the average probability value in a quantization interval  $[a, b]$ , i.e.,  $f_{uv}(x^*) = \frac{1}{b-a} \cdot \int_a^b f_{uv}(x)dx$ , with (27) and (28), we obtain that

$$\begin{aligned} (b-a)f_{uv}(x^*) &= \int_a^{\frac{a+b}{2}} (f_{uv}(x) + f_{uv}(a+b-x))dx \\ &> \int_a^{\frac{a+b}{2}} 2f_{uv}\left(\frac{a+b}{2}\right)dx \\ &= (b-a)f_{uv}\left(\frac{a+b}{2}\right) \end{aligned} \quad (29)$$

Then we have  $f_{uv}(x^*) > f_{uv}((a+b)/2)$ , with  $f_{uv}(x)$  being continuous monotonic decreasing when  $x \geq 0$ , we finally derive that  $x^* \in (a, \frac{a+b}{2})$ .

#### ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers for their thoughtful comments and suggestions which help to improve the technical presentation of this paper.

#### REFERENCES

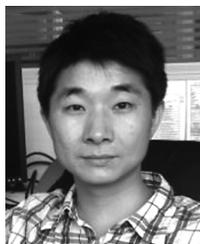
- [1] *Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s: Video*, ISO/IEC 11172-2 MPEG-1, 1993.
- [2] *Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video*, ITU-T Rec. H.262 and ISO/IEC 13818-2 MPEG-2, 1995.
- [3] *Information Technology-Generic Coding of Audio-Visual Objects Part 2: Visual*, ISO/IEC 14496-2 MPEG-4, 1999.
- [4] *Video Coding for Low Bitrate Communication*, ITU-T Rec. H.263 Version 1, 1995, Version 2, Sep. 1997.
- [5] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, document JVT-G050.doc, ITU-T Rec. H.264 and ISO/IEC 14496-10, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Mar. 2003.
- [6] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jun. 2003.
- [8] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.
- [9] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, R. Susanto, and C. C. Kuo, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 953–958, Jul. 2005.
- [10] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 246–250, Feb. 1997.
- [11] Z. He and S. K. Mitra, "A linear source model and a unified control algorithm for DCT video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 970–982, Nov. 2002.
- [12] N. Kamaci, Y. Altunbasak, and R. Mersereau, "Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 994–1006, Aug. 2005.
- [13] Z. Li, W. Gao, F. Pan, S. W. Ma, K. P. Lim, G. N. Feng, X. Lin, S. Rahardja, H. Q. Lu, and Y. Lu, "Adaptive rate control for H.264," *J. Visual Commun. Image Representation*, vol. 17, no. 2, pp. 376–406, Apr. 2006.
- [14] D.-K. Kwon, M.-Y. Shen, and C. C. J. Kuo, "Rate control for H.264 video with enhanced rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 517–529, May 2007.
- [15] S.-C. Chang, J.-F. Yang, C. F. Lee, and J. N. Hwang, "A novel rate predictor based on quantized DCT indices and its rate control mechanism," *Signal Process. Image Commun.*, vol. 18, pp. 427–441, Jul. 2003.
- [16] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Efficient rate-distortion estimation for H264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 600–611, May 2006.
- [17] Q. Chen and Y. He, "A fast bits estimation method for rate-distortion optimization in H.264/AVC," in *Proc. Picture Coding Symp. (PCS)*, San Francisco, CA, Dec. 2004, pp. 133–134.
- [18] C.-H. Tseng, H. M. Wang, and J.-F. Yang, "Enhanced intra 4 × 4 mode decision for H.264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 8, pp. 1027–1032, Aug. 2006.
- [19] M. G. Sarwer and L.-M. Po, "Fast bit rate estimation for mode decision of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1402–1407, Oct. 2007.
- [20] X. Zhao, J. Sun, and W. Gao, "A novel rate estimation model for mode decision of H.264/AVC," in *Proc. SPIE Visual Commun. Image Process.*, vol. 7257. 2009, 72570R.
- [21] L.-M. Po and K. Guo, "Transform-domain fast sum of the squared difference computation for H.264/AVC rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 765–773, Jun. 2007.
- [22] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Rate-distortion modeling for efficient H.264/AVC encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 530–543, May 2007.
- [23] F. Müller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electron. Lett.*, vol. 29, no. 22, pp. 1935–1936, Oct. 1993.
- [24] H. Gharavi and A. Tabatabai, "Sun-band coding of monochrome and color images," *IEEE Trans. Circuits Syst.*, vol. 35, no. 2, pp. 207–214, Feb. 1988.
- [25] J. Sun, W. Gao, D. Zhao, and Q. Huang, "Statistical model, analysis and approximation of rate-distortion function in MPEG-4 FGS videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 535–539, Apr. 2006.
- [26] J. Sun, W. Gao, D. Zhao, and W. Li, "On rate-distortion modeling and extraction of H.264/SVC fine-granular scalable video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 323–336, Mar. 2009.
- [27] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
- [28] K. Sharifi and A. L. Garcia, "Estimation of shape parameter for generalized Gaussian distribution in subband decomposition of video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 52–56, Feb. 1995.
- [29] J. A. Domínguez-Molina, G. González-Farías, and R. M. Rodríguez-Dagnino, "A practical procedure to estimate the shape parameter in the generalized Gaussian distribution," Tech. Rep. I-01-18, Sep. 2001 [Online]. Available: [http://www.cimat.mx/reportes/enlinea/I-01-18\\_eng.pdf](http://www.cimat.mx/reportes/enlinea/I-01-18_eng.pdf)

- [30] T.-Y. Wang, H.-W. Li, Z.-M. Li, and Z.-H. Wang, "A fast parameter estimation of generalized Gaussian distribution," in *Proc. Int. Conf. Signal Processing (ICSP)*, vol. 1. Guilin, China, 2006, pp. 101–104.
- [31] *Lanczos Approximation* [Online]. Available: [http://en.wikipedia.org/wiki/Lanczos\\_approximation](http://en.wikipedia.org/wiki/Lanczos_approximation)
- [32] *Rate-Distortion Constrained Estimation of Quantization Offsets*, document JVT-O066.doc, Joint Video Team (JVT), Apr. 2005.
- [33] *Adaptive Quantization Encoding Technique Using an Equal Expected-Value Rule*, document JVT-N011.doc, Joint Video Team (JVT), Jan. 2005.
- [34] *Mean Value Theorem* [Online]. Available: [http://en.wikipedia.org/wiki/Mean\\_value\\_theorem](http://en.wikipedia.org/wiki/Mean_value_theorem)
- [35] T. M. Cover and J. A. Thomas, "Optimal codes," *Elements of Information Theory*. New York: Wiley, 1991, Chapter 5, pp. 84–86.
- [36] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York: Plenum, 1988.
- [37] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advances, Applications*. San Diego, CA: Academic, 1990.
- [38] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application. I. Fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 287–298, Apr. 1997.
- [39] *Low Complexity Transform and Quantization*, documents JVT-B038.doc and JVT-B039.doc, Joint Video Team (JVT), Jan. 2002.
- [40] I. E. G. Richardson, "The baseline profile," *H.264 and MPEG-4 Video Compression Video Coding for Next-Generation Multimedia*. Chichester, U.K.: Wiley, 2003, Chapter 6.4, pp. 187–193.
- [41] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," in *Proc. ITU-T Q.6/SG16 VCEG 13th Meeting*, Austin, TX, Apr. 2001, document VCEG-M33.
- [42] *Joint Video Team (JVT) Reference Software* [Online]. Available: [http://iphome.hhi.de/suehring/tml/download/old\\_jm](http://iphome.hhi.de/suehring/tml/download/old_jm)



**Xin Zhao** received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006. He is currently working toward the Ph.D. degree in computer science at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

His current research interests include image and video coding, video transcoding, and video processing.



**Jun Sun** received the B.S. degree in computer science from the University of Science and Technology, Beijing, China, in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2006.

From 2006 to 2008, he was a Post-Doctorate with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. In 2008, he joined the Institute of Computer Science and Technology, Peking

University, where he is currently an Associate Professor. His current research interests include scalable video coding, rate-distortion analysis, video transcoding, and video streaming.



**Siwei Ma** (S'03) received the B.S. degree in computer science from Shandong Normal University, Jinan, China, in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

From 2005 to 2007, he was a Post-Doctorate with the University of Southern California, Los Angeles. Then he joined the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, where he is currently an Associate Professor. He published over 30 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission. His current research interests include image and video coding, video streaming, and transmission.



**Wen Gao** (M'92–SM'05–F'09) received the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1985, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Professor in computer science with the Harbin Institute of Technology from 1991 to 1995, a Professor in computer science with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, from 1996 to 2005. He is currently a Professor with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He has been leading research efforts to develop systems and technologies for video coding, face recognition, sign language recognition and synthesis, and multimedia retrieval. He published four books and over 500 technical articles in refereed journals and proceedings in the areas of signal processing, image and video communication, computer vision, multimodal interface, pattern recognition, and bioinformatics. His current research interests include signal processing, image and video communication, computer vision, and artificial intelligence.

Dr. Gao received many awards, including five national awards for research achievements and activities. He did many services to academic society, such as the General Co-chair of the IEEE International Conference on Multimedia and Expo in 2007, and the Head of Chinese delegation to the Moving Picture Expert Group of International Standard Organization since 1997. He is also the Chairman of the working group responsible for setting a national Audio Video Coding Standard for China.