A Frame-Level Rate Control Scheme Based on Texture and Nontexture Rate Models for High Efficiency Video Coding

Bumshik Lee, *Member, IEEE,* Munchurl Kim, *Member, IEEE,* and Truong Q. Nguyen, *Fellow, IEEE*

***Abstract*—In this paper, a frame-level rate control scheme is proposed based on texture and nontexture rate models for High Efficiency Video Coding (HEVC). Due to more complicated coding structures and the adoption of new coding tools, the statistical characteristics of transform residues are significantly different depending on the depth levels of coding units (CUs) from which the residues are obtained. A new texture rate model is constructed for the transform residues, which are categorized into three types of CUs: low-, medium- and high-textured CUs. One single Laplacian probability PDF model is used for each residue category to derive a rate-quantization model. Based on the Laplacian PDF, a simplified rate model for texture bits is derived using entropy. In addition, an analytic rate model for nontexture bits is proposed, which also takes into account the different characteristics of nontexture bits occurring in various depths of CUs in HEVC. The nontexture bitrates are modeled based on the linear relation between the total nontexture data and the dominant nontexture data in each CU category. Based on the proposed rate models for the texture and nontexture bits, accurate rate control can be achieved owing to more precise rate estimation. The experimental results show that the proposed rate control scheme achieves the average PSNR with 0.44 dB higher and the average PSNR standard deviation of 0.32 point lower with the buffer status levels maintained very close to target buffer levels, compared to the conventional methods. Finally, the proposed rate control scheme remarkably outperforms the conventional schemes especially for the sequences of complex texture and large motion.**

***Index Terms*—High Efficiency Video Coding (HEVC), Laplacian distribution, rate control, rate-quantization model.**

I. Introduction

SINCE the H.264/AVC [1] video coding standard was developed, many studies have been conducted for further improvement of coding efficiency toward next generation video codecs. Recently, the Joint Collaborative Team on Video Coding (JCT-VC), co-established by ISO/IEC and ITU-T, has been working on the development of a next generation video coding standard, called High Efficiency Video Coding (HEVC) [2], which has reached almost the final stage for standardization. HEVC aims at achieving the coding efficiency improvement of about 50% or more compared to H.264/AVC. Due to the expanded prediction and transform block sizes with a flexible coding structure, HEVC can efficiently encode the video sequences from low to high picture resolutions.

In the current HEVC, more flexible hierarchical-block structures are adopted with quadtree partitions and higher depth levels, which are composed of coding unit (CU), prediction unit (PU), and transform unit (TU) [3]. The CU is a basic processing unit for encoding and decoding, which includes motion estimation (ME) and motion compensation (MC), transform, quantization, and entropy coding, etc. The CU with the maximum size is called the coding tree block (CTB) for which its size and the number of predefined depths are signaled in a sequence level. The PU in a CU block is a block unit for ME/MC whose block sizes include $2N \times 2N$, $2N \times N$ and $N \times 2N$ for the $2N \times 2N$ CU block, where $2N$ is horizontal or vertical pixel size of the CU. The block sizes of TU range from 32×32 to 4×4 pixels. Quadtrees for TUs are constructed in the leaf nodes of CUs. Fig. 1 shows an example of HEVC block partitions of the CU, PU, and TU. As shown in Fig. 1(a), the CUs can be hierarchically partitioned in a CTB, each of which is similar to an MB in the conventional video codecs. Within each CU block, the ME/MC is performed in its PU blocks of various prediction block sizes to produce residual signals. Then, for the obtained residual signals, the transform and quantization of the TU blocks are performed in quadtree-partitioned manners within each CU block. Fig. 1(b) shows an example of the quadtree partitions for CU and TU blocks.

In Fig. 1, CU_k indicates CU in depth level k. Therefore, CU_0 has the same block size as CTB. When CTB size is set to 64×64 with the maximum depth level of four, the CU_0, CU_1, CU_2 and CU_3 have their block sizes 64×64, 32×32, 16×16 and 8×8, respectively. Note that the CTB size in HEVC can be predefined up to the maximum size of 64×64 and the smallest CU (SCU) size can be predefined down to 8×8. Due to such flexible coding structures with a large block prediction and transform as shown in Fig. 1, it is more difficult to model the residual signals for rate control applications.

Manuscript received April 21, 2013; revised July 4, 2013; accepted July 18, 2013. Date of publication August 6, 2013; date of current version March 4, 2014. This work was supported in part by the Industrial Strategic Technology Development Program of MKE/KEIT [10039214, Video Codec SoC for Ultra High Definition], and in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology under Grant 2012R1A6A3A03039732. This paper was recommended by Associate Editor D. O. Wu.

B. Lee and T. Q. Nguyen are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093 USA (e-mail: bsl001@eng.ucsd.edu; nguyent@ece.ucsd.edu).

M. Kim is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Yuseong-gu, Daejeon 305-701, Korea (e-mail: mkim@ee.kaist.ac.kr).

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.2013.2276880

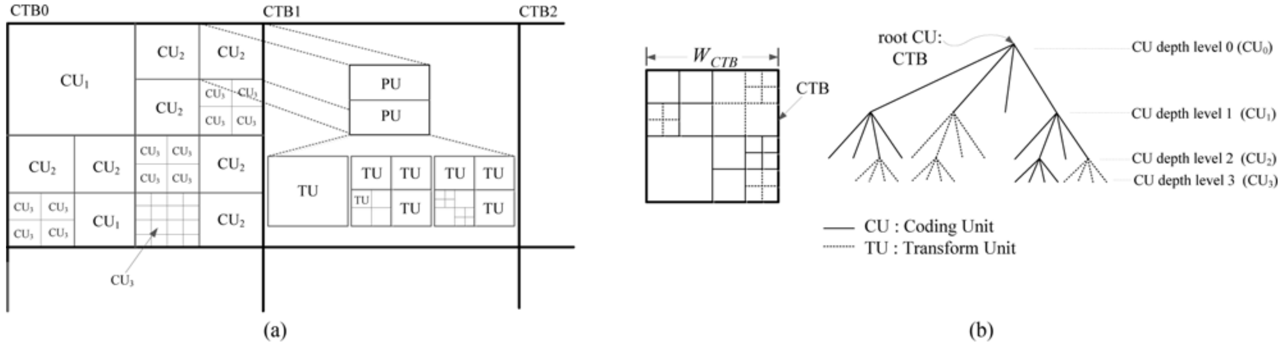

CTB0
CTB1
CTB2
CU_1
CU_2
CU_3
PU
PU
TU
(a)
W_{CTB}
CTB
root CU: CTB
CU depth level 0 (CU_0)
CU depth level 1 (CU_1)
CU depth level 2 (CU_2)
CU depth level 3 (CU_3)
CU : Coding Unit
TU : Transform Unit
(b)

Fig. 1. Example of CU, PU, and TU block structure in the HEVC coding standard. (a) Example of CU, PU, and TU. (b) Quadtree of CU and TU blocks.

For example, various block sizes of prediction and transform in hierarchical structures often result in very different statistical characteristics of the residual signals depending on the block sizes or the coding depths. So, it is often difficult to capture various statistical characteristics of residual signals only with one single rate-quantization model. Unfortunately, there have been very few works on the rate control for HEVC-like video codecs. It is worthy but challenging to investigate an appropriate rate control scheme with a new rate model suitable for HEVC of such hierarchical quadtree structure. Furthermore, due to many advanced coding tools adopted in HEVC, the relative portion of texture bits has been reduced compared to that of nontexture bits. Therefore, it becomes important to separately model the texture and nontexture rates for rate control. Especially, for the nontexture rate prediction, the occurrence characteristics of nontexture bits must be carefully investigated with respect to the hierarchical coding structures of CU and TU in HEVC.

There have been a number of investigations for rate models and their applications to rate control. Most of the existing schemes are designed for certain specific video coding standards, such as the MPEG-2 [4], H.263 [5], or H.264/AVC [1]. Their rate-quantization models for rate control have also been developed based on the statistics of residues or the frame or block complexity [6]–[14]. Chiang *et al.* [6] proposed a quadratic rate-quantization model, assuming that the predicted residues follow the Laplacian PDF. In this model, the complexity of basic coding units is estimated using the mean absolute difference (MAD) of the residual signals in order to determine an appropriate quantization parameter value. This model has been popularly used for rate control. Consequently, many variants of the quadratic rate-quantization model in [6] are proposed in [7]–[14] for various video coding standards, such as H.264/AVC or H.263. In [7], a quadratic rate model is used for H.264/AVC rate control and has been adopted in H.264/AVC JM reference software [31]. In [8], a linear rate-quantization model, which is simplified from the quadratic model, is applied for H.264/AVC rate control. An improved frame complexity estimation scheme is used for a more accurate rate estimation. It is reported that this scheme is more advantageous for abrupt changes in frame complexity. In [9], a simplified linear rate-quantization model for H.264/AVC was also proposed based on TMN5 in MPEG-2 to estimate the rate of coded-video frames. Dong *et al.* [10] used a simplified linear rate-quantization model with an estimated MAD value, where a more context adaptive MAD prediction method was employed. In [11] and [12], for frame or MB complexity measure, the peak signal-to-noise ratio (PSNR) is used instead of MAD. In [13], a scheme to obtain the ratio of frame MAD for frame complexity prediction is proposed for rate control of H.264/AVC. In addition to spatial domain MAD for the frame complexity, Kwon *et al.* [14] used the sum of absolute transformed difference (SATD) as a complexity measure instead of MAD.

In [15], a rate-distortion model has been proposed for rate control for H.264/AVC. Based on the relations between the pixel domain and the transform domain, a Laplacian model parameter is estimated. However, this method cannot be used for multiple types of transform blocks in HEVC in spite of high accuracy of the rate model in H.264/AVC where a 4×4 transform block size is only used in this scheme [15].

In [16] and [17], a Cauchy PDF is used for rate control schemes. Kamaci *et al.* [16] showed that the Cauchy PDF is more suitable to represent the residual signal in H.264/AVC, and they incorporated it into a frame-based rate control algorithm. Using the Cauchy PDF, a rate control algorithm for a basic unit layer is proposed in [17], which achieves higher coding efficiency with similar buffer occupancy compared to [16], which is a frame-level rate control scheme. In [18] and [19], ρ-domain-based rate control schemes using the percentage of zero-quantized coefficients are proposed. Despite higher accuracy of their rate models, it is difficult to map ρ to quantization step sizes. In addition, novel rate control schemes were investigated using adaptive rounding offset (ARO) [20], Lagrangian multiplier adjustment [21], histogram of difference frames (HOD) [22], gradient for intraframe rate control [23], etc.

Recently, a rate control scheme [24] was adopted in HEVC reference software, for which the same quadratic rate-quantization model in [6] was used with the MAD estimation as complexity measurement. However, the proposed rate control scheme shows some limitations to obtain accurate rate control results due to its inaccurate rate-quantization model and unpredictable nature of nontexture bits. A rate control scheme using an R-λ model [25] has recently been adopted to the HEVC test model reference software version 10.0 (HM10.0) [30] by replacing the previous method in [24].

Although the aforementioned rate models and their applications to rate control schemes are useful for the existing

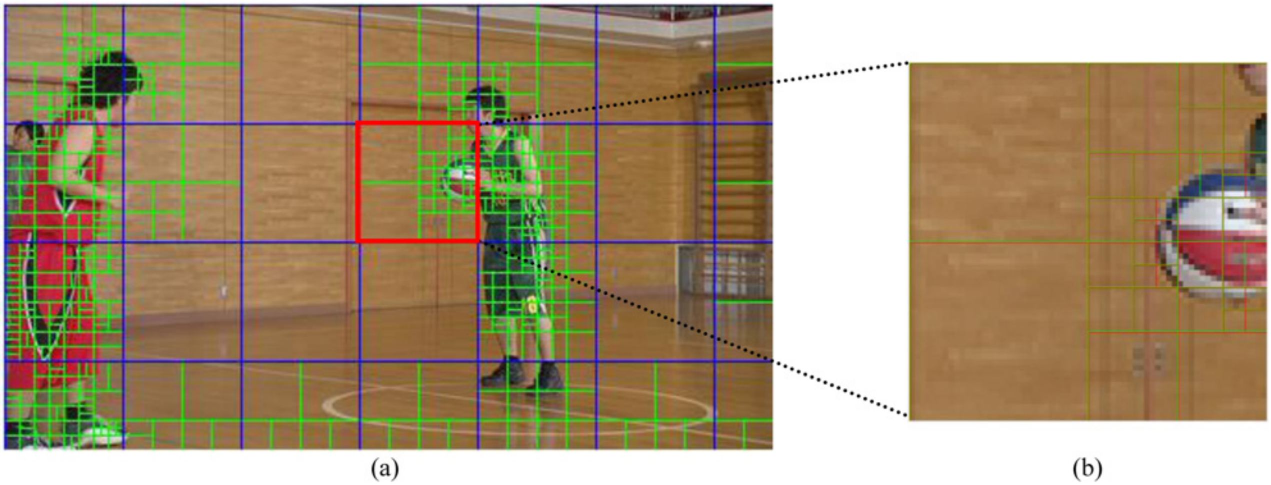


Fig. 2. Quadtree partitions of CU and TU. (a) CU partitions for the 30th frame of *BasketballPass* (416 × 240)—Blue lines: CTB, Green lines: CU and PU blocks. (b) Quadtree partitions of CU and TU block in a CTB—Red lines: TU blocks, Yellow lines: CU blocks.

hierarchical coding structures with various transform block sizes in HEVC.

More importantly, the rate control for such a complicated coding structure with quadtree coding and large transform kernels have not been thoroughly investigated yet, which is more challenging for emerging HEVC video coding standard. In this paper, a frame-level rate control scheme for HEVC is proposed based on our new rate models for texture and nontexture bits, which take into account the different statistical characteristics of transform residues and header bits in various coding depths of CU. For the rate model for texture bits, the Laplacian PDF-based rate model in our previous work [3] is used with a little modification for the rate control application. The texture rate model for rate control proposed in this paper is based on multiple Laplacian PDFs, each of which is separately applied for the transform residues in a CU category. Note that each CU is categorized into one of low-, medium- and high- texture type of CU. Thus, the proposed rate control scheme is well adaptive to diverse signal characteristics. The entropy for the texture rate model is used and derived to a simplified rate-quantization model for each CU category. As a result, the proposed overall texture rate model becomes a linear combination of the simplified rate-quantization models for three CU categories. In addition, an analytic rate model for nontexture bits is proposed, which also takes into account the different characteristics of nontexture bits occurring in various depths of CUs in HEVC. The nontexture bitrates are modeled based on the linear relation between the total nontexture data and the dominant nontexture data in CU categories. Based on the proposed rate models for the texture and nontexture bits, more accurate rate control can be achieved owing to more precise rate estimation.

This paper is organized as follows. In Section II, texture and nontexture rate models are proposed with statistical analysis for transformed residues and header bits with respect to the CU depth levels. In Section III, a rate control scheme for HEVC video codec is proposed. In Section IV, the experimental results are presented to show the effectiveness of the proposed frame-level rate control scheme for HEVC. Finally, the paper is concluded in Section V.

II. Texture and Nontexture Rate Models For HEVC

The HEVC has advantages to efficiently encode video sequences from small picture resolution to high picture resolution, such as full HD (1,920 × 1,080) or beyond owing to the flexible coding block structures. The optimal depth levels of CU and TU can be determined based on a rate-distortion optimization (RDO) sense. Fig. 2 shows an example of partitioned CU and TU blocks in quadtree for *BasksetballPass* (416 × 240) sequences encoded at QP = 28 by HM 10.0 [30]. As shown in Fig. 2(a), the homogeneous regions tend to be encoded in larger CU blocks with lower depth levels whereas smaller partitioned CU blocks with higher depth levels are used to encode the detailed regions or edges. Fig. 2(b) shows an enlarged CTB area for an area of the basketball where the CTB is quadtree-partitioned into the CU blocks of various sizes in RDO sense during encoding. Within each CU block, TU also shows similar patterns of quadtree transform partitions since the sizes of TU blocks are determined according to the texture characteristics in RDO sense. As a result, the variances of transformed coefficient values in different depths of CU are shown to be significantly different.

Table I shows the average variances of the transform coefficient values according to CU depth levels and coding types for the sequences of *BasketballPass*, *BQMall*, and *BQTerrace* with three different spatial resolutions encoded at QP = 28 and 33 using HM 10.0 [30]. As shown in Table I, the average variances tend to have higher values for the CU blocks of higher depth levels and for the intracoded CU. In particular, the variances of the transform coefficient values for the intracoded CU are significantly higher than those of the intercoded CU. We can notice that the statistical characteristics of transform residues for the CU blocks are substantially different according to depth levels and for their coding types. Therefore, it must be pointed out that the residues need to be differently treated or modeled according to the depth levels of the CU and the coding types of the CU blocks. Therefore, when a single-PDF model for predicted residues is used for rate models, the source residues cannot be accurately modeled because such a single

TABLE I

Average Variances of Transform Coefficients for Different Depth Levels of the CU and Coding Types of CU Blocks

Sequences	QP	CU-inter				CU-intra
		CU_0	CU_1	CU_2	CU_3	
BasketballPass (416×240)	28	7.14	11.55	33.56	69.20	124.00
	33	16.12	33.21	59.24	112.25	182.11
BQMall (832×480)	28	13.11	22.40	30.56	69.33	121.60
	33	25.12	36.89	64.00	122.11	125.58
BQTerrace (1920×1080)	28	11.64	22.15	35.40	55.09	111.97
	33	26.70	47.12	64.25	111.25	130.06

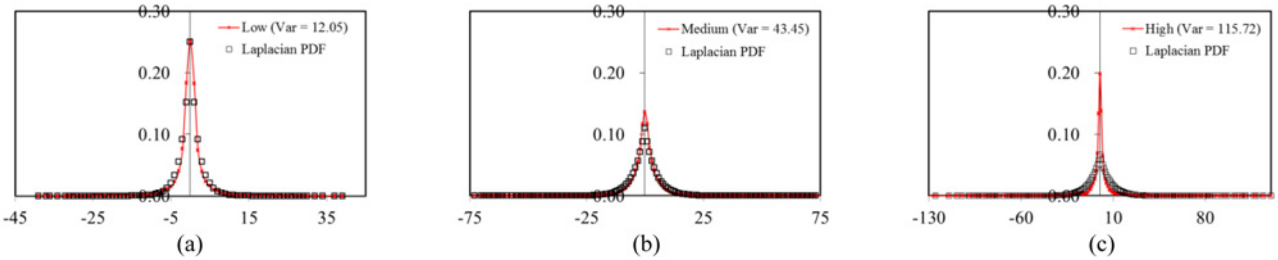


Fig. 3. Comparisons between histogram of DCT coefficient and Laplacian PDF for each CU depth level for *RaceHorses* (832 × 480) sequence encoded with QP = 28. (a) DCT distribution for CU_l. (b) DCT distribution for $\mathrm{CU}_{m,}$. (c) DCT distribution for CU_h.

model may fail to represent different statistical characteristics of residual sources obtained from various coding depth levels and the coding types of CU blocks.

A. Rate Modeling for Texture Bits

Motivated from the observations in Fig. 2 and Table I, we categorize CUs into three groups according to the signal characteristics in different CU depths and for different coding types. As shown in Fig. 2 and Table I, CU_0 and CU_1 contain relatively more homogeneous texture signals such that smaller variances of transform coefficient values are shown while CU_2 and CU_3 show medium values in average variance and the largest average variances are observed in CU-intra. Hence, CU_0 and CU_1 are categorized as CU_l, CU_2 and CU_3 are categorized into CU_m. Finally, intracoded CU (CU-intra) is assigned to CU_h category. Here, the subscript l, m, and h indicate the indexes of low-textured, medium-textured, and highly-textured CU categories, respectively. Fig. 3 shows comparisons between histogram of DCT coefficients and Laplacian PDF for each CU depth level for *RaceHorses*(832 × 480) sequence encoded with QP = 28. As shown in Fig. 3, the actual histogram in each CU category is well fitted into the Laplacian PDFs in the CU depth levels. In addition, we performed the hypothesis chi-square test in order to check the goodness-of-fit [26] for the Laplacian PDF. The histograms of the DCT coefficient values accept the chi-square test with a confidence interval 95%. Thus, we can justify that the DCT coefficient values well follow the Laplacian PDFs. A mathematical analysis that the distribution of DCT coefficients follows Laplacian PDF is provided in [27]. Thus, we model the source distribution of the transform residues for CU_C category ($C = l, m$ or h) as

$$f_C(l) = \frac{\lambda_C}{2} e^{-\lambda_C |l|}, \lambda_C \in \{\lambda_l, \lambda_m, \lambda_h\} \quad (1)$$

where l is the random variable for transform coefficient values. λ_C is the model parameters of the Laplacian PDF, and is obtained from residual transform coefficients for CU_C category which can be computed as [28]

$$\lambda_C = \sqrt{2} \Big/ \sigma_C \quad (2)$$

where σ_C is the average standard deviation of the residual transform coefficients for CU_l, CU_m or CU_h in a frame. Since the statistical characteristics of transform coefficients are considerably different between different CU categories as shown in Table I, the model parameter λ_C is obtained separately for each CU category. A rate model in a frame level is proposed by taking into account the resulting bit amounts from the CU blocks of different CU categories. Therefore, a rate model using entropy for HEVC is proposed as

$$R(q) = \alpha_l N_l H_l(q, \lambda_l) + \alpha_m N_m H_m(q, \lambda_m) + \alpha_h N_h H_h(q, \lambda_h) \quad (3)$$

where $H_l(q, \lambda_l)$, $H_m(q, \lambda_m)$, and $H_h(q, \lambda_h)$ refer to the entropies for CU_l, CU_m, and CU_h categories, respectively. q is a quantization step size. α_l, α_m, and α_h are the model parameters, which are computed with a linear regression scheme [26] between the actual rates and the estimated ones for each CU category, and are updated frame by frame. N_l, N_m, and N_h indicate the total numbers of pixels for CU_l, CU_m, and CU_h categories in a frame, respectively. Note that the sum of N_{SKIP} (sum of the total number of pixels for SKIP blocks), N_l, N_m, and N_h is equal to the frame size in the number of pixels. That is, frame size $= N_{SKIP} + N_l + N_m + N_h$. The entropy for the transform coefficients to be quantized in CU_C category is computed such as

$$H_C(q, \lambda_C) = -P_{0,C} \cdot \log_2 P_{0,C} - 2 \sum_{i=1}^{\infty} P_{i,C} \cdot \log_2 P_{i,C} \quad (4)$$

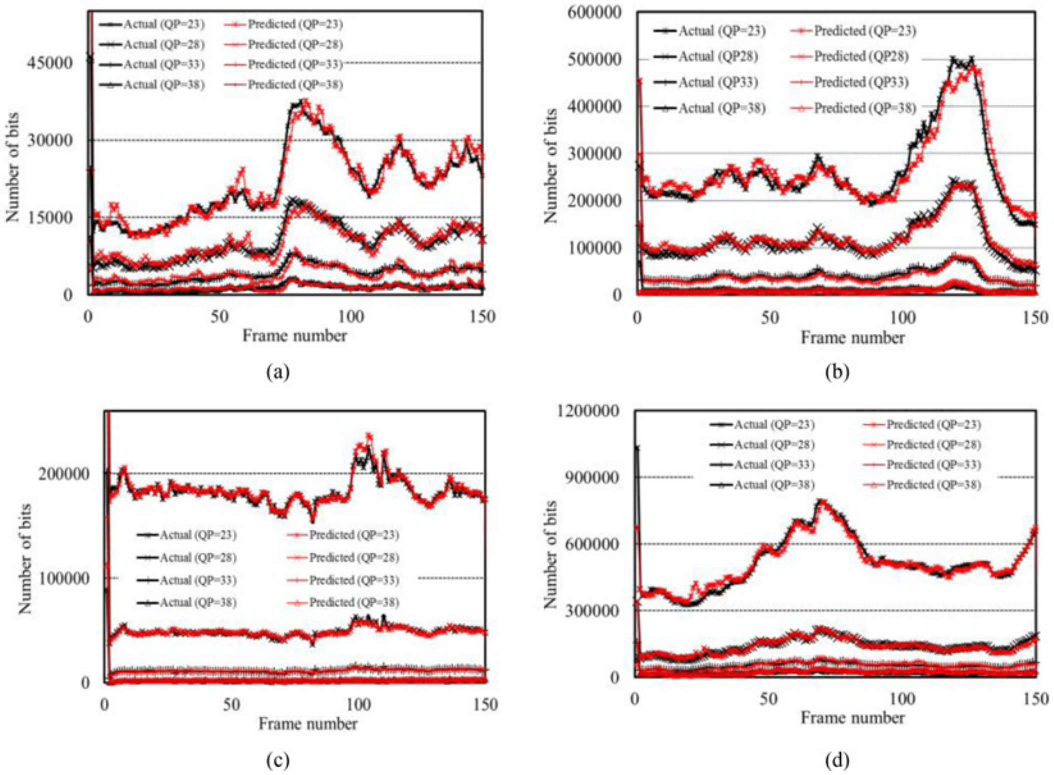


Fig. 4. Comparison of actual texture bits and estimated texture bits for various video sequences and QPs. (a) *BasketballPass* (416 × 240). (b) *RaceHorses* (832 × 480). (c) *BigShips* (1280 × 720). (d) *BasketballDrive* (1920 × 1080).

where $H_C(q, \lambda_C)$ is the entropy for CU$_C$ category, q is a quantization step size, $P_{0,C}$ and $P_{i,C}$ are the probabilities that the transform coefficients are quantized to zero and quantization interval i for CU$_C$ category, respectively, which are computed as

$$P_{0,C} = \int_{-(q-fq)}^{q-fq} f_C(l)dl = 1 - e^{-(1-f)q\lambda_C} \tag{5}$$

and

$$\begin{aligned} P_{i,C} &= \int_{(i-f)\cdot q}^{(i+1-f)\cdot q} f_C(l)dl \\ &= \begin{cases} 1/2 \cdot e^{-(i-f)q\lambda_C} \cdot \left(1 - e^{-q\lambda_C}\right), \text{for } i > 0 \\ 1/2 \cdot e^{(i-f)q\lambda_C} \cdot \left(1 - e^{-q\lambda_C}\right), \text{for } i < 0 \end{cases} \end{aligned} \tag{6}$$

In (5) and (6), q is a quantization step size and f is a rounding offset, which is set to 1/6 for intercoded CU and 1/3 for intracoded CU. The entropy can be expressed in closed form by substituting (5) and (6) into (4) and, is rewritten as

$$\begin{aligned} H_C(q, \lambda_C) &= -P_{0,C} \cdot \log_2 P_{0,C} - 2\sum_{i=1}^{\infty} P_{i,C} \cdot \log_2 P_{i,C} \\ &= -\left(1 - e^{-(1-f)\cdot q\cdot\lambda_C}\right) \cdot \log_2\left(1 - e^{-(1-f)\cdot q\cdot\lambda_C}\right) \\ &\quad - e^{-(1-f)q\lambda_C}\left\{\log_2\left(1 - e^{-q\cdot\lambda_C}\right) - 1 + \right. \\ &\quad \left. \frac{q\cdot\lambda_C}{\ln 2}\left(f - \frac{1}{1-e^{-q\cdot\lambda_C}}\right)\right\}. \end{aligned} \tag{7}$$

Using (5) and (6), $H_C(q, \lambda_C)$ can simply be expressed in terms of $P_{0,C}$ as

$$\begin{aligned} H_C(q, \lambda_C) &= \left\{ \begin{aligned} &\underbrace{\log_2\left(1 - e^{-(1-f)q\lambda_C}\right)}_{\equiv A} - \underbrace{\log_2\left(1 - e^{-q\lambda_C}\right)}_{\equiv B} \\ &+ \underbrace{\left(\frac{2 - f + (f-1)\,e^{-q\cdot\lambda_C}}{1 - e^{-q\cdot\lambda_C}}\,\frac{q\cdot\lambda_C}{\ln 2}\right)}_{\equiv C} \end{aligned} \right\}. \\ &\quad e^{-(1-f)q\lambda_C} \underbrace{-\log_2\left(1 - e^{-(1-f)q\lambda_C}\right)}_{\equiv D} \\ &= A\,(1-P_0) - B\,(1-P_0) + C\,(1 - P_0) + D \end{aligned} \tag{8}$$

where A, B, and D approach to zero when $q\lambda$ takes on a large enough value. In this case, (8) can be simplified to

$$H_C(q, \lambda_C) \approx a_C(1 - P_{0,C}) = a_C\left(e^{-(1-f)q\lambda_C}\right). \tag{9}$$

Finally, by substituting (9) into (3), the proposed rate model in (3) can be rewritten as

$$\begin{aligned} R(q) &= \alpha_l \hat{N}_l\left(e^{-(1-f)q\hat{\lambda}_l}\right) \\ &\quad + \alpha_m \hat{N}_m\left(e^{-(1-f)q\hat{\lambda}_m}\right) + \alpha_h \hat{N}_h\left(e^{-(1-f)q\hat{\lambda}_h}\right) \end{aligned} \tag{10}$$

where $\hat{N}_l$, $\hat{N}_m$, and $\hat{N}_h$ are the block sizes, and $\hat{\lambda}_l$, $\hat{\lambda}_m$, $\hat{\lambda}_h$ are the model parameter estimates, respectively, for low-, medium-, and high-complexity CU categories.

TABLE II
COMPARISONS OF ESTIMATED TEXTURE BITS

Test sequences	QP	Single-PDF based		Quadratic model [6]		Proposed	
		NRMSE	Correlation	NRMSE	Correlation	NRMSE	Correlation
BasketballPass (416×240)	23	0.285	0.961	0.412	0.879	0.078	0.973
	28	0.312	0.934	0.872	0.851	0.120	0.947
	33	0.354	0.861	2.116	0.812	0.179	0.948
	38	0.414	0.749	2.443	0.735	0.249	0.838
RaceHorses (832×480)	23	0.645	0.914	0.787	0.886	0.097	0.977
	28	0.066	0.931	0.848	0.843	0.080	0.985
	33	0.077	0.939	0.756	0.721	0.041	0.977
	38	0.352	0.811	0.878	0.711	0.322	0.828
BigShips (1280×720)	23	0.042	0.979	1.344	0.923	0.022	0.978
	28	0.031	0.989	1.680	0.899	0.041	0.997
	33	0.060	0.991	2.155	0.811	0.049	0.999
	38	0.148	0.974	2.321	0.800	0.131	0.995
BasketballDrive (1920×1080)	23	0.121	0.924	1.211	0.812	0.067	0.988
	28	0.592	0.911	1.380	0.777	0.067	0.975
	33	0.133	0.917	1.680	0.611	0.052	0.929
	38	0.819	0.883	1.732	0.676	0.312	0.965

In our proposed rate model, $\hat{N}_C$ and $\hat{\lambda}_C$ can be computed as the weighted moving average from the previous frames such as

$$\hat{\lambda}_{C,n+1} = \frac{1}{N_f}\sum_{i=0}^{N_f-1} w_i \lambda_{C,n-i} \text{ and } \hat{N}_{C,n+1} = \frac{1}{N_f}\sum_{i=0}^{N_f-1} w_i N_{C,n-i} \tag{11}$$

where $\hat{\lambda}_{C,n+1}$ and $\hat{N}_{C,n+1}$ are the model parameter estimate for the Laplacian PDF and the estimate for total number of pixels to be applied for $(n+1)$th frame, respectively. N_f is the number of the most recent frames to estimate $\hat{\lambda}_{C,n+1}$ and $\hat{N}_{C,n+1}$, which is empirically set to three with w_0= 0.533, w_1= 0.333, and w_2= 0.134 for all the test sequences in this paper. $N_{C,n-i}$ is the total block size that CU$_C$ category is selected in the (n-i)th frame, and is computed as the sum of the block sizes (pixels) of nonSKIP mode in CU$_C$ category. Fig. 4 shows the estimation accuracy of the proposed rate model in (10). The experimental results in Fig. 4 were obtained using HM 10.0 [30] for IPPP GOP structure. The test sequences with various spatial resolution and characteristics, such as *BasketballPass* (416 × 240), *RaceHorses* (832 × 480), *BigShips* (1,280 × 720), and *BasketballDrive* (1,920 × 1,080), are used with QP = 23, 28, 33, and 38. As shown in Fig. 4, the proposed rate model shows quite accurate estimation performances for the various test sequences and QP values. The rate estimation by the proposed model in (10) is very close to the actual amounts of bits over frames, which is conspicuously accurate for the sequence periods that contain abrupt changes in the scene over the frames. It indicates that the proposed rate model is well adaptive to the changes in signal characteristics.

Table II shows the performance comparisons for estimation accuracy on texture rate models. The proposed rate model is compared with a single-PDF based model and a quadratic model [6], which has been popularly used for rate control applications. Estimation accuracy is measured in terms of Pearson correlation [26] and normalized root mean square error (NRMSE), which is defined as

$$NRMSE = \frac{RMSE}{\bar{R}_{act}} = \frac{1}{\bar{R}_{act}}\sqrt{\frac{\sum_{i=0}^{N_f-1}(R_{act,i} - \tilde{R}_{\text{model},i})^2}{N_f}} \tag{12}$$

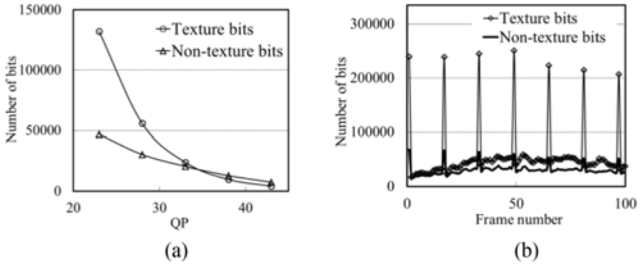


Fig. 5. Comparison of texture and nontexture bits for *BQMall* sequence. (a) Overall bits for texture and nontexture data for different QP values. (b) Frame-level bits for texture and nontexture data for QP = 28.

where $R_{act,i}$ is the number of actual bits, and $\tilde{R}_{\text{model},i}$ is the number of predicted bits by a rate model for the ith frame. Also, N_f is the number of encoded frames and $\bar{R}_{act}$ is the average number of actual bits for all encoded N_f frames. Smaller NRMSE values indicate that the rate models yield more accurate bit estimation results. As shown in Table II, the proposed rate model yields smaller RMSE and higher correlation values, compared to the conventional models based on single-PDFs or quadratic rate models. Especially for the test sequences with complex or various texture characteristics, the proposed model conspicuously outperforms the conventional models. This indicates that the proposed model is more locally adaptive to the temporal and spatial variations of texture signals over frames.

B. Analysis and Rate Modeling for Nontexture Bits

In HEVC, signaling of the side information for nontexture data, which includes motion vectors, quadtree split information for TU and CU, signaling information of prediction models, filter signaling information etc., becomes more significantly important due to the adoption of various coding tools compared to its previous video coding standards. Fig. 5 compares the amounts of texture and nontexture data for *BQMall* (832 × 480) sequences. As shown in Fig. 5(a), the relative portions of nontexture bits increases as QP values get higher and it becomes even larger than texture bit amount for QP = 34 and higher. In general, it is difficult to accurately predict

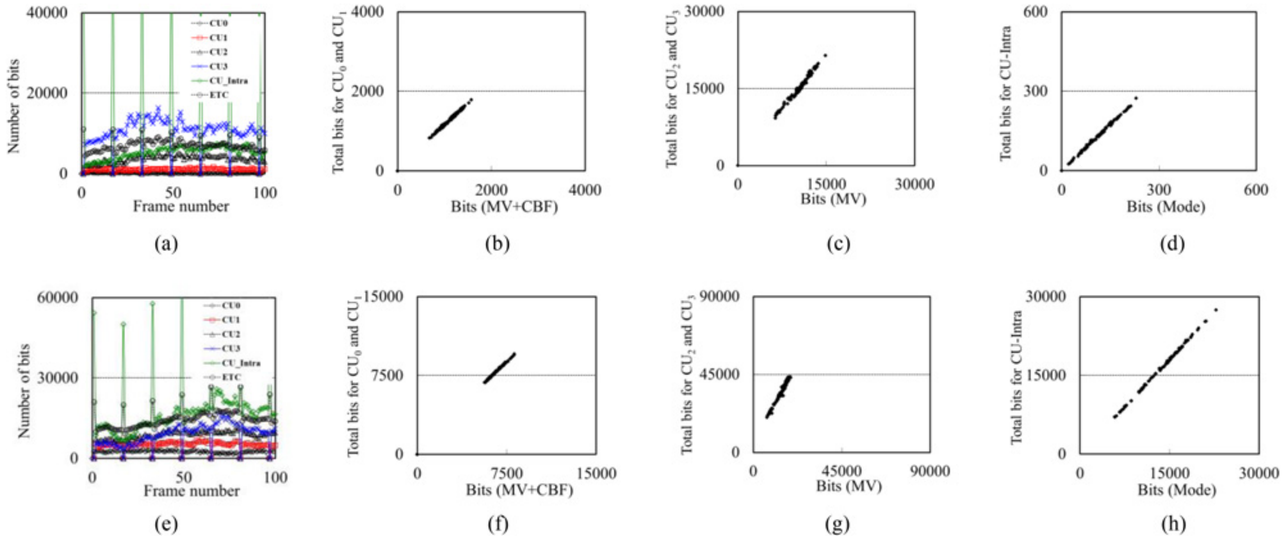


Fig. 6. Analysis of nontexture bit occurrences and relation between the total bits for nontexture and the dominant bits for each CU category. (a) *BQMall* (QP = 28). (b) Nontexture bits for CU$_l$ (*BQMall*, QP = 28). (c) Nontexture bits for CU$_m$, (*BQMall*,QP = 28). (d) Nontexture bits for CU$_h$, (*BQMall*,QP = 28). (e) *BasaketballDrive* (QP = 33). (f) Nontexture bits for CU$_l$ (*BasketballDrive*, QP = 33). (g) Nontexture bits for CU$_m$, (*BasketballDrive*,QP = 33). (h) Nontexture bits for CU$_h$, (*BasketballDrive*, QP = 33).

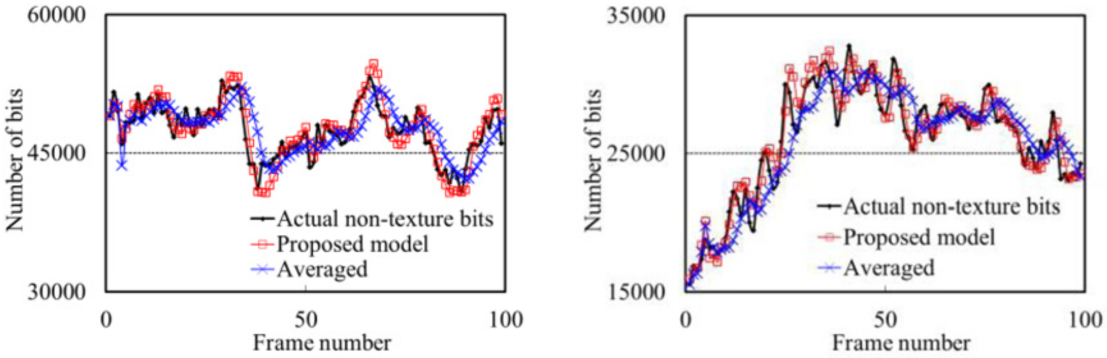
nontexture bits due to the high fluctuation in the number of header bits over frames.

Owing to the improvement of compression efficiency for texture bits by adopting many sophisticated coding tools, the handling of nontexture data becomes more important. Fig. 5(b) also shows the texture and nontexture bits in frames for *BQMall* sequences (832 × 480) with QP = 28 and the intraperiod of 16 frames. Therefore, for low bit rate ranges with higher QP values, the occurrence of nontexture data should carefully be dealt with for rate control. Most of the conventional rate control schemes employ a nontexture estimation method by simply averaging the nontexture data collected from a few previous frames [7], [9], [13], [17], [20], [21]. In [14], the nontexture bits are predicted by using the number of nonzero motion vectors. However, it is not effective for such HEVC of the hierarchical CU structure with deeper coding depths because the number of nonzero motion vectors alone cannot precisely represent the various nontexture bits occurring in different CU coding depth levels. In [15], a unified linear rate-quantization model that can estimate both the texture and nontexture bits was proposed where a single R_{total}-Q model is used to estimate the texture and nontexture rates. So, unpredictable nature of nontexture bits may significantly influence the entire rate control performance. Due to these reasons, it may cause unstable and highly fluctuated estimation for the nontexture data, thus resulting in inaccurate rate estimation for the total number of bits and fluctuating PSNR values over frames. In order to overcome the problem of the exiting schemes for the nontexture rate models, we thoroughly analyze the characteristics of nontexture data in various CU depths.

Fig. 6 shows nontexture bits and its relations with total nontexture bits in each CU category. The *BQMall* and *BasketballDrive* sequences are used and the period of intraframe coding is set to 16. Firstly, Fig. 6(a) and (e) show nontexture bit amounts in frames for each CU level during encoding. As shown in Fig. 6(a) and (e), the largest portion of nontexture bits for intercoded CU is from CU$_3$ while the CU blocks of lower depths result in relatively smaller portions of nontexture bits. This is due to the fact that, since the CU in the deepest coding depth level such as CU$_3$ has the smallest partitioned blocks, many motion vectors occur from the partitioned blocks. In addition, it is important to note that the dominant portions of nontexture data types are different depending on CU categories. For example, for the CU of low depth levels, such as CU$_0$ and CU$_1$, which are in the CU$_l$ category, the coded block flag (*cbf*) for TU blocks and the motion vectors together take the most significant portion of the nontexture data while the information for motion vectors becomes more dominant for the CU of higher depth levels, such as CU$_2$ and CU$_3$, which are in the CU$_m$ category. For intracoded CU blocks in the CU$_h$ category, the information of intraprediction modes mostly dominates the others in the nontexture data. Fig. 6(b), (c), (d), (f), (g), and (h) show the linear relations between the total nontexture bits and the most dominant nontexture bits for the three CU categories. As shown in Fig. 6, the total nontexture bits for CU$_2$ and CU$_3$ can be expressed only with the bit amounts of motion vectors. On other hand, the mode information for intracoded CU shows high correlation with the total nontexture bits due to a number of its intraprediction modes. Motivated from the observation in Fig. 6, we model the nontexture rate as

$$\begin{aligned}\tilde{R}_{nonTex} &= \alpha_l \tilde{R}_l(mv + cbf) \\ &\quad + \alpha_m \tilde{R}_m(mv) + \alpha_h \tilde{R}_h(\text{mode})\end{aligned} \tag{13}$$

where $\tilde{R}_{nonTex}$ is an estimate for the total number of nontexture bits, $\tilde{R}_l(mv + cbf)$ is an average number of bits for motion vectors plus *cbf* information in the CU$_l$ category, $\tilde{R}_m(mv)$ is an average number of bits for motion vectors for the CU$_m$ category, and $\tilde{R}_h(\text{mode})$ is the average bit amount for the CU$_h$ category. $\tilde{R}_l(mv + cbf)$, $\tilde{R}_m(mv)$, and $\tilde{R}_h(\text{mode})$ can be

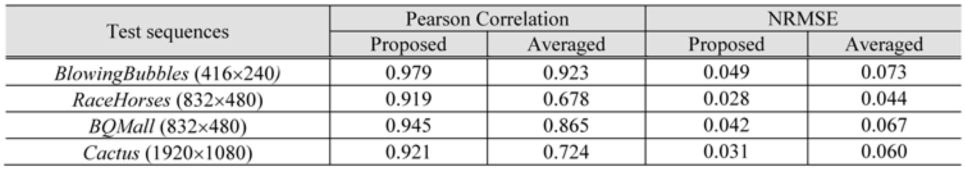


(a) (b)

Fig. 7. Estimation of Nontexture bits. (a) *RaceHorses* (832 × 480, QP = 28). (b) *BQMall* (832 × 480, QP = 28).

TABLE III

ACCURACY COMPARISON OF NONTEXTURE BIT ESTIMATIONS FOR THE PROPOSED NONTEXTURE RATE MODEL AND AN AVERAGE NONTEXTURE RATE MODEL.



Test sequences	Pearson Correlation		NRMSE	
	Proposed	Averaged	Proposed	Averaged
BlowingBubbles (416×240)	0.979	0.923	0.049	0.073
RaceHorses (832×480)	0.919	0.678	0.028	0.044
BQMall (832×480)	0.945	0.865	0.042	0.067
Cactus (1920×1080)	0.921	0.724	0.031	0.060

computed from several previous frames and, to remove high fluctuation during encoding, they can be computed as

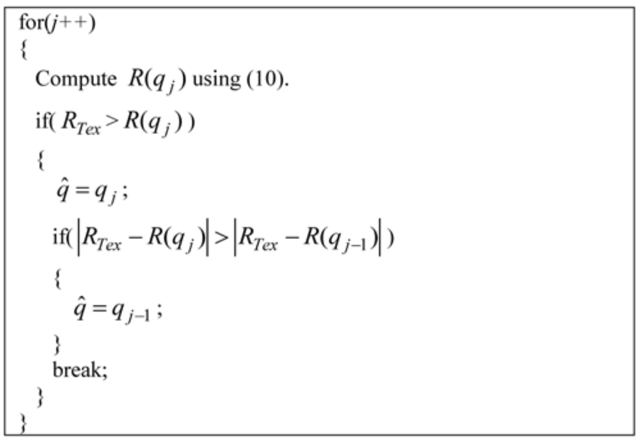
$$\tilde{R}_C = \frac{1}{N_w - 2}\sum_{i=1}^{N_w}\left(R^C_{nonTex,n_f-i} - R^C_{\min} - R^C_{MAX}\right) \qquad (14)$$

where $\tilde{R}_C \in \{\tilde{R}_l, \tilde{R}_m, \tilde{R}_h\}$, N_w is the window size to be averaged, n_f is the current frame number, R^C_{nonTex,n_f-i} is the nontexture bits in CU$_C$ category for $(n_f - i)$th frame, $R^C_{\min}$ and R^C_{MAX} are the minimum and maximum values of nontexture bits, respectively, in the window. In the frame window of size N_w, the average of the nontexture bits is computed by excluding the maximum value and the minimum value so that the nontexture bit amount can be more reliably estimated. N_w is set to eight in this paper. Fig. 7 shows the amounts of estimated nontexture bits by the proposed nontexture rate model in (13) for *RaceHorses* and *BQMall* sequences encoded at QP = 28. As shown in Fig. 7, the proposed nontexture model yields quite accurate estimation results. Compared to a simple average model [7], [9], [13], [17], [20], [21] with several previous frames, the proposed nontexture rate model shows its superiority in terms of estimation accuracy for nontexture bits.

Table III shows accuracy comparison of nontexture bit estimations for the proposed nontexture rate model and an average nontexture rate model. As similarly shown in Fig. 7, the proposed nontexture rate model outperforms the average model in estimation accuracy for nontexture bits. Especially for the test sequences with a number of abrupt changes in the single characteristics of frames such as *RaceHorses*, the proposed model shows considerably higher accuracy in estimation performances for nontexture bits. Based on the proposed texture rate model in (10) and the proposed nontexture rate model in (13), a new rate control scheme for quadtree structured HEVC is designed, which is described in the following section.

TABLE IV

PSEUDO CODE FOR COMPUTING q AND QP VALUES



```
for(j++)
{
  Compute $R(q_j)$ using (10).
  if( $R_{Tex} > R(q_j)$ )
  {
    $\hat{q} = q_j$;
    if( $|R_{Tex} - R(q_j)| > |R_{Tex} - R(q_{j-1})|$ )
    {
      $\hat{q} = q_{j-1}$;
    }
    break;
  }
}
```

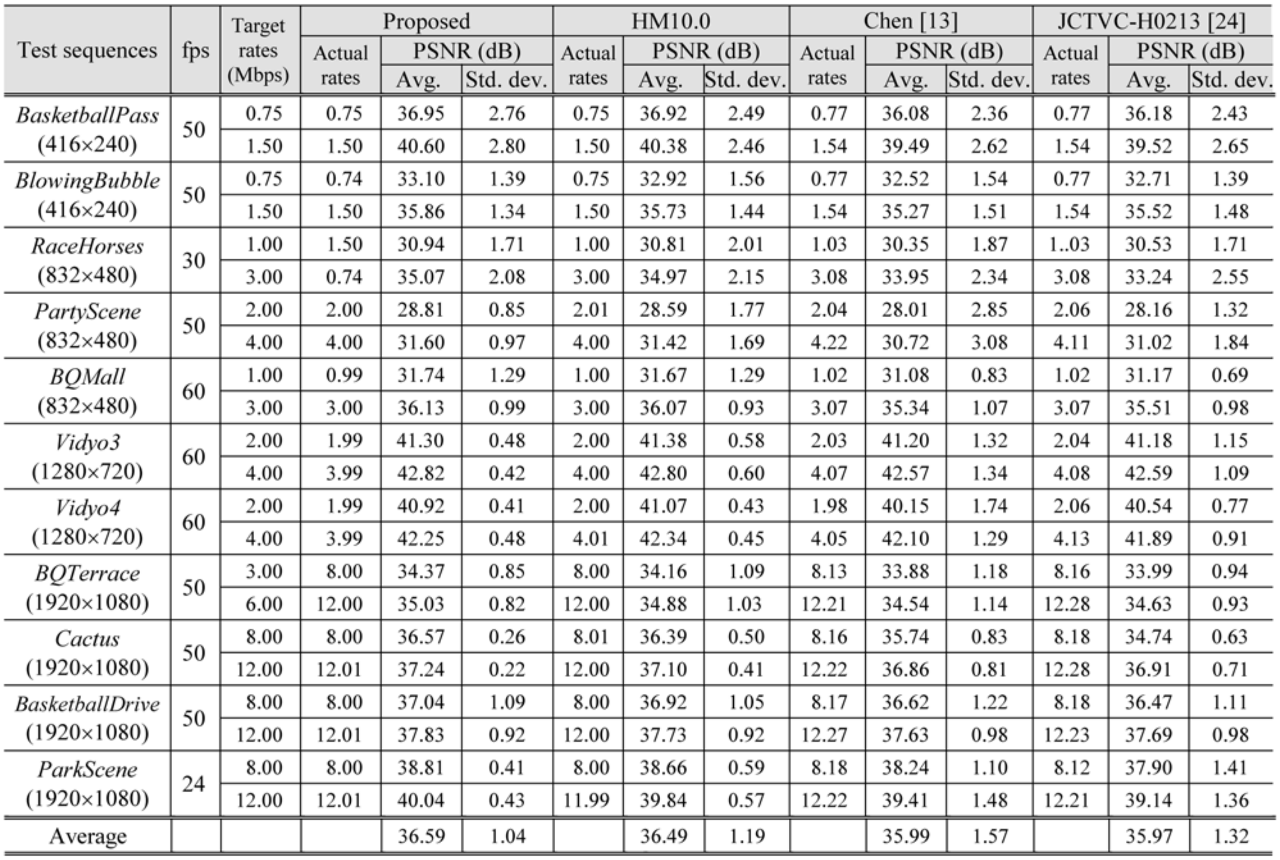
III. PROPOSED FRAME-LEVEL RATE CONTROL FOR HEVC

A. GOP- and Frame-Level Bit Allocation and QP Determination

In this section, a rate control scheme is proposed based on the proposed texture and nontexture rate models. For rate control, a GOP level bit allocation is first made as

$$T_{GOP,i} = \begin{cases} \left(R_T/f\right) \times N_{GOP}, \; fori = 1 \\ \left(R_T/f\right) \times N_{GOP} + R_{r,i-1}, \; otherwise \end{cases} \qquad (15)$$

TABLE V
EXPERIMENTAL RESULTS FOR THE PERFORMANCE COMPARISONS IN HEVC RATE CONTROL



Test sequences	fps	Target rates (Mbps)	Proposed			HM10.0			Chen [13]			JCTVC-H0213 [24]		
			Actual rates	PSNR (dB)		Actual rates	PSNR (dB)		Actual rates	PSNR (dB)		Actual rates	PSNR (dB)	
				Avg.	Std. dev.		Avg.	Std. dev.		Avg.	Std. dev.		Avg.	Std. dev.
BasketballPass (416×240)	50	0.75	0.75	36.95	2.76	0.75	36.92	2.49	0.77	36.08	2.36	0.77	36.18	2.43
		1.50	1.50	40.60	2.80	1.50	40.38	2.46	1.54	39.49	2.62	1.54	39.52	2.65
BlowingBubble (416×240)	50	0.75	0.74	33.10	1.39	0.75	32.92	1.56	0.77	32.52	1.54	0.77	32.71	1.39
		1.50	1.50	35.86	1.34	1.50	35.73	1.44	1.54	35.27	1.51	1.54	35.52	1.48
RaceHorses (832×480)	30	1.00	1.50	30.94	1.71	1.00	30.81	2.01	1.03	30.35	1.87	1..03	30.53	1.71
		3.00	0.74	35.07	2.08	3.00	34.97	2.15	3.08	33.95	2.34	3.08	33.24	2.55
PartyScene (832×480)	50	2.00	2.00	28.81	0.85	2.01	28.59	1.77	2.04	28.01	2.85	2.06	28.16	1.32
		4.00	4.00	31.60	0.97	4.00	31.42	1.69	4.22	30.72	3.08	4.11	31.02	1.84
BQMall (832×480)	60	1.00	0.99	31.74	1.29	1.00	31.67	1.29	1.02	31.08	0.83	1.02	31.17	0.69
		3.00	3.00	36.13	0.99	3.00	36.07	0.93	3.07	35.34	1.07	3.07	35.51	0.98
Vidyo3 (1280×720)	60	2.00	1.99	41.30	0.48	2.00	41.38	0.58	2.03	41.20	1.32	2.04	41.18	1.15
		4.00	3.99	42.82	0.42	4.00	42.80	0.60	4.07	42.57	1.34	4.08	42.59	1.09
Vidyo4 (1280×720)	60	2.00	1.99	40.92	0.41	2.00	41.07	0.43	1.98	40.15	1.74	2.06	40.54	0.77
		4.00	3.99	42.25	0.48	4.01	42.34	0.45	4.05	42.10	1.29	4.13	41.89	0.91
BQTerrace (1920×1080)	50	3.00	8.00	34.37	0.85	8.00	34.16	1.09	8.13	33.88	1.18	8.16	33.99	0.94
		6.00	12.00	35.03	0.82	12.00	34.88	1.03	12.21	34.54	1.14	12.28	34.63	0.93
Cactus (1920×1080)	50	8.00	8.00	36.57	0.26	8.01	36.39	0.50	8.16	35.74	0.83	8.18	34.74	0.63
		12.00	12.01	37.24	0.22	12.00	37.10	0.41	12.22	36.86	0.81	12.28	36.91	0.71
BasketballDrive (1920×1080)	50	8.00	8.00	37.04	1.09	8.00	36.92	1.05	8.17	36.62	1.22	8.18	36.47	1.11
		12.00	12.01	37.83	0.92	12.00	37.73	0.92	12.27	37.63	0.98	12.23	37.69	0.98
ParkScene (1920×1080)	24	8.00	8.00	38.81	0.41	8.00	38.66	0.59	8.18	38.24	1.10	8.12	37.90	1.41
		12.00	12.01	40.04	0.43	11.99	39.84	0.57	12.22	39.41	1.48	12.21	39.14	1.36
Average				36.59	1.04		36.49	1.19		35.99	1.57		35.97	1.32

where $T_{GOP,i}$ is the number of bits allocated for *i*th GOP, f is the frame rate [frame per second (fps)], R_T is the target bit rate, N_{GOP} is the number of frames in a GOP, and $R_{r,i-1}$ is the number of remaining bits in the (*i*-1)th GOP. In order to determine the target bits for each frame in a GOP, the bit allocation for *j*th frame in the GOP is computed as

$$T_{f,j} = w\hat{T}_j + (1-w)\tilde{T}_j \tag{16}$$

where

$$\hat{T}_j = \frac{R_T}{f} + \gamma\left(T_{bl,j} - B_{C,j}\right) \text{and} \tilde{T}_j = \frac{R_{j,GOP}}{n_{rem}}. \tag{17}$$

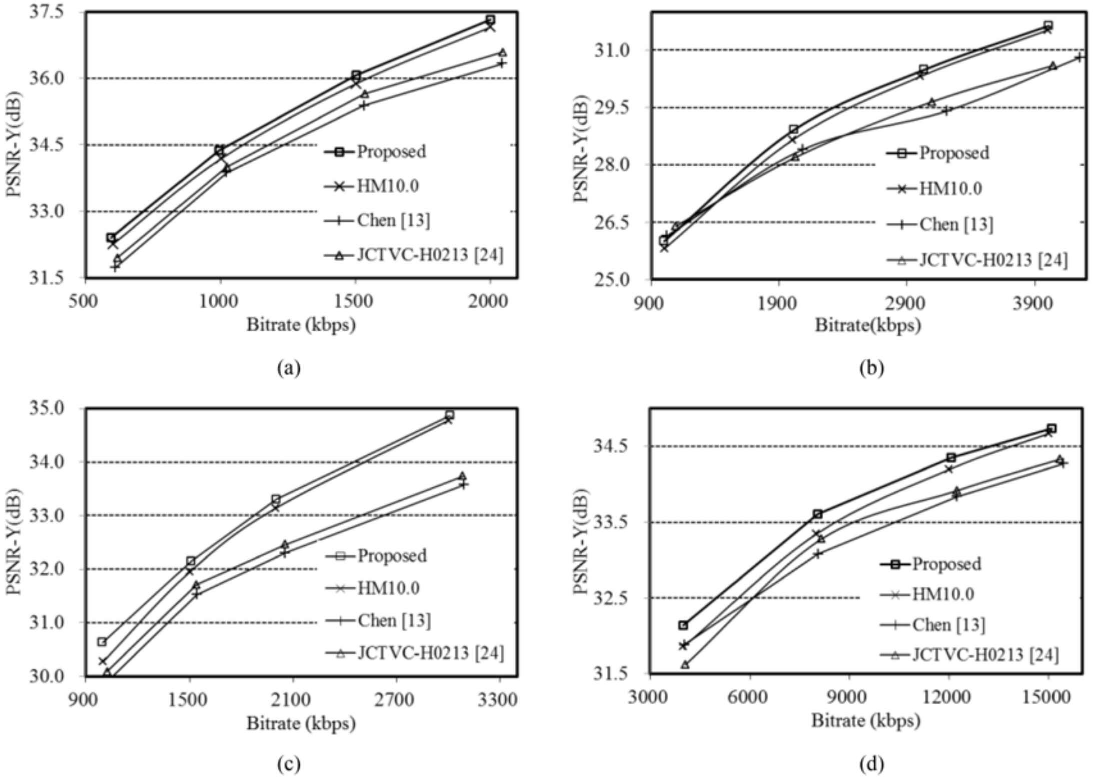
In (17), $T_{bl,j}$ and $B_{C,j}$ are the target buffer level and the current buffer level for *j*th frame, respectively. $R_{j,GOP}$ is the remaining bits in the GOP, and n_{rem} is the number of frames remained for encoding in the GOP. γ and w are constant values, which are set empirically to 0.25 and 0.5 in the proposed scheme for all test sequences, respectively.

Once the bit allocation for a frame is done, an appropriate QP value to encode the frame should be determined based on (10) and (13). The texture bit in the *j*th frame can be written as

$$R_{Tex,j} = T_{f,j} - R_{nonTex,j} \tag{18}$$

where $T_{f,j}$ is the total number of bits allocated in the *j*th frame in (17), $R_{nonTex,j}$ is the estimate of header bits including the motion vectors, the split information for quadtree CU and TU, the prediction partition information, the coded block flags, etc. The estimate for nontexture bits is obtained by (13) in this paper. From (10), (13), and (18), the quantization step size for a frame to be encoded can be determined. Table IV shows a pseudo code for computing q and QP values. In the Table IV, j is the QP index from 0 to 52 in HEVC, $R(q_j)$ is the proposed texture rate model in (10), $R_{Tex}$is the bit amount allocated in the frame according to (18), $\hat{q}$ is the estimate of the quantization step size, and q_j is a quantization step size corresponding to the *j*th QP value. From Table IV, quantization step size and QP value can be obtained using (10) and (18). In HEVC, the mapping of a quantization step size to a QP value is identical as H.264/AVC [1]. Thus, the corresponding QP values can easily be obtained.

It must be pointed out that it is very important to determine an initial QP value in each GOP. If a large QP value is selected, the subsequent frames in a GOP might be degraded due to coarse quantization of the first frame while larger amounts of bits are allocated in the following interframes. On the other hands, if a small QP value in the GOP is applied for the first frame, the visual qualities of the subsequent frames cannot be guaranteed due to the comparatively small amount of bit allocation into the following interframes. In [7], the first QP is empirically determined for H.264/AVC. However, since the coding efficiency of intraframes over the intercoded frames in HEVC is quite different from that of H.264/AVC, the initial QP determination for H.264/AVC in [7] is no longer applicable for



(a)

(b)

(c)

(d)

Fig. 8. Comparison of RD performances between the proposed rate control scheme, the HM rate control scheme, Chen's method and JCTVC-H0213. (a) *BlowingBubbles* (416 × 240). (b) *PartyScene* (832 × 480). (c) *RaceHorses* (832 × 480). (d) *BQTerrace* (1920 × 1080).

TABLE VI

BIT RATE REDUCTIONS FOR THE PROPOSED METHOD AGAINST THE EXISTING METHODS

Proposed vs.		*BlowingBubbles*	*PartyScene*	*RaceHorses*	*BQTerrace*
BDBR(%)	HM10.0	-4.8%	-5.2%	-4.4%	-10.1%
	Chen [13]	-16.1%	-16.3%	-22.4%	-22.8%
	JCTVC-H0213 [24]	-12.3%	-16.0%	-16.3%	-18.4%

HEVC. In the proposed method, we empirically set the initial QP values (QP_0) as

$$QP_0 = \begin{cases} 22, l_0 < BPP \\ 27, l_1 < BPP \leq l_0 \\ 32, l_2 < BPP \leq l_1 \\ 36, l_3 < BPP \leq l_2 \\ 42, BPP \leq l_3 \end{cases} \tag{19}$$

$$BPP = \frac{R}{f \times M \times N} \tag{20}$$

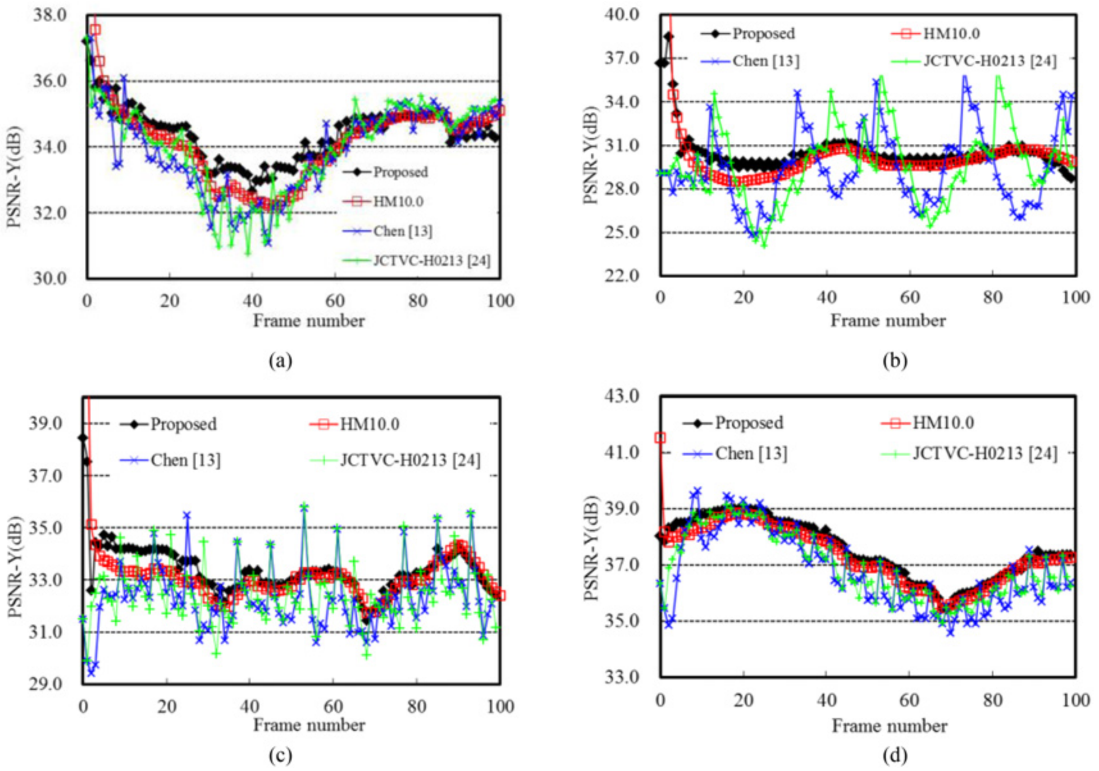
where R and f are a predefined bit rate and a frame rate, respectively. $M \times N$ is the spatial resolution of the sequence. l_0, l_1, l_2, and l_3 are empirically set to 0.211, 0.110, 0.070, and 0.030, respectively, for all test sequences and QP values.

B. Model Parameter Update

In the proposed texture and nontexture rate models in (10) and (13), the model parameters,α_l, α_m, and α_h, give significant influences on the accuracy and stability of rate control. If those values are not appropriately selected, significant fluctuations of output bits might occur during encoding. In the proposed rate control scheme, the model parameters α_l, α_m, and α_h can be updated frame by frame based on linear regression between the predicted bits and the actual bits in each CU category as

$$\alpha_C = \frac{\sum_{i=0}^{N-1} R_{C,n-i}\hat{R}_{C,n-i} - \frac{1}{N}\left(\sum_{i=0}^{N-1} R_{C,n-i}\right)\left(\sum_{i=0}^{N-1} \hat{R}_{C,n-i}\right)}{\sum_{i=0}^{N-1} \hat{R}^2_{C,n-i} - \frac{1}{N}\left(\sum_{i=0}^{N-1} \hat{R}_{C,n-i}\right)^2} \tag{21}$$

where α_C is a model parameter for each CU category. $\hat{R}_{C,n}$ is the rate estimate for CU_C category in nth frame, and is computed as $\hat{N}_C\left(e^{-(1-f)q\hat{\lambda}_C}\right)$ for texture bits in (10) and $\tilde{R}_C$ for nontexture bits in (13). Our proposed method has three model parameters for each CU category for texture and nontexture models, respectively. Hence, six model parameters are independently updated frame by frame. The parameter updates are performed based on the linear regression scheme in (21) for both texture and nontexture rate models. $R_{C,n}$ is the number of actual texture or nontexture bits for CU_C category, and N is the number of frames used for linear regression and is set to three in the proposed rate control scheme. By using



(a)

(b)

(c)

(d)

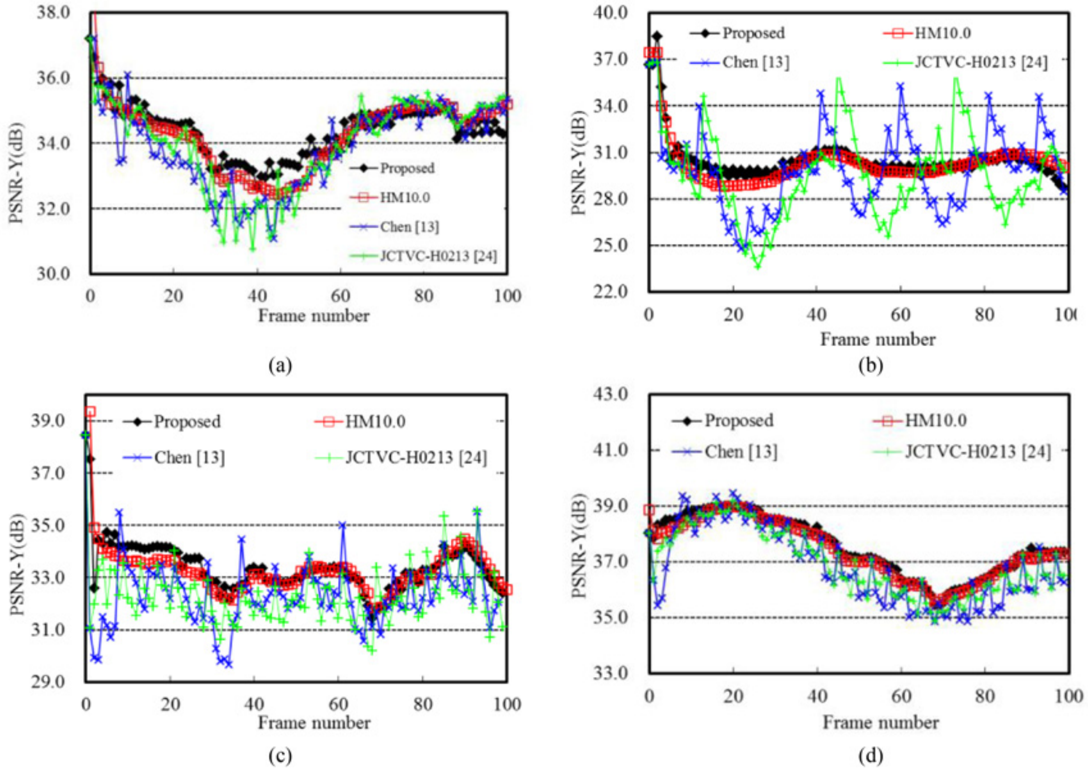
Fig. 9. Comparison of PSNR values and PSNR value fluctuations over frames when their own initial QPs are used for for the four rate control methods. (a) PSNR for *BlowingBubbles* (416 × 240) @1Mb/s Proposed: Avg. PSNR = 34.37 dB, PSNR std. dev.= 0.82 HM10.0: Avg. PSNR = 34.18 dB, PSNR std. dev. = 1.46 Chen: Avg. PSNR = 33.87 dB, PSNR std. dev. = 1.28 JCTVC-H0213: Avg. PSNR = 33.99 dB, PSNR std. dev. = 1.37. (b) PSNR for *PartyScene* (832 × 480) @3Mb/s Proposed: Avg. PSNR = 30.48 dB, PSNR std. dev.= 1.43 HM10.0: Avg. PSNR = 30.30, PSNR std. dev. = 2.37 Chen: Avg. PSNR = 29.40 dB, PSNR std. dev. = 2.57 JCTVC- H0213: Avg. PSNR = 29.64 dB, PSNR std. dev. = 2.58. (c) PSNR for *RaceHorses* (832 × 480) @2Mb/s Proposed: Avg. PSNR = 33.37 dB, PSNR std. dev. = 0.96 HM10.0: Avg. PSNR = 33.14 dB, PSNR std. dev. = 1.24 Chen: Avg. PSNR = 32.30 dB, PSNR std. dev. = 1.20 JCTVC- H0213: Avg. PSNR = 32.46 dB, PSNR std. dev. = 1.17. (d) PSNR for *BasketballDrive* (1920 × 1080) 8Mb/s Proposed: Avg. PSNR = 37.51 dB, PSNR std. dev. = 1.05 HM10.0: Avg. PSNR = 37.35 dB, PSNR std. dev. = 1.08 Chen: Avg. PSNR = 36.76 dB, PSNR std. dev. = 1.34 JCTVC- H0213: Avg. PSNR = 36.92 dB, PSNR std. dev. = 1.20.

such dedicated rate model parameters in (10) and (13) for each CU category, more accurate rate control can be achieved.

IV. Experimental Results

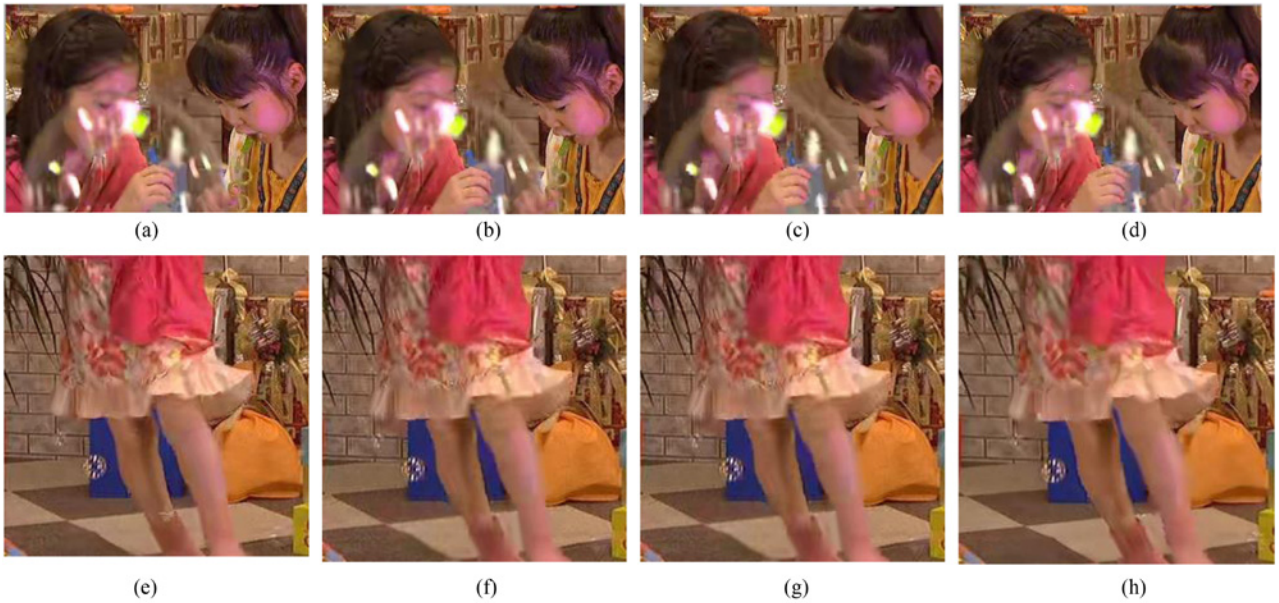
In order to verify the effectiveness of the proposed frame-level rate control scheme, we implemented it into HM 10.0 [30]. For the experiments, 11 test sequences of various resolutions with different signal characteristics are used. The GOP structure for low delay applications is IPPP. Each sequence is encoded at various target bitrates. The proposed rate control scheme is compared with the newly adopted HM rate control [30], Chen's method [13] and JCTVC-H0213 [24] in terms of average PSNR, PSNR standard deviation, accuracy of target bitrates, frame visual quality, and buffer status levels. Table V shows the experimental results for the proposed rate control scheme with comparison to the HM10.0 rate control, Chen's method and JCTVC-H0213. As shown in Table V, for the same target bitrates, the proposed rate control scheme shows the average PSNR with 0.1, 0.6, and 0.62 dB higher and the average PSNR standard deviation of 0.15, 0.53, and 0.28 points lower, compared to the HM rate control scheme, Chen's method and JCTVC-H0213, respectively.

This is due to the fact that the proposed texture rate model accurately estimates the texture output bits with multiple Laplacian PDFs for the transform residues in various CU depth levels. Furthermore, the proposed nontexture rate model can precisely estimate nontexture data based on the dominant nontexture bits in various CU depth levels as well. It is observed that PSNR improvements are more significant for the test sequences with complex texture and highly moving objects, such as *BlowingBubbles*, *PartyScene, RaceHorses* and *ParkScene* sequences. These test sequences include abrupt changes in signal characteristics over frames and tend to be encoded in various CU depth levels and with various coding types. Therefore, the proposed rate control scheme is able to more effectively control the output bitrates of HEVC codec with higher PSNR values by performing content-adaptive rate estimation. Fig. 8 and Table VI show the RD performance and bit rate reductions for the proposed rate control scheme with comparison to the other three methods. We obtain the experimental results at four target bitrate points for each test sequence by comparing the proposed rate control scheme with the other three methods in terms of Bjøntegaard delta bit rate (BDBR) (%) [32] in Table VI. As shown in Fig. 8, the proposed scheme outperforms the other three methods in BDBR. For all bitrate ranges, the proposed scheme shows remarkable superiorities of RD performances, especially for the test sequences with various texture types. In Table VI, the proposed method shows the 22.8% bit rate reduction



(a) (b) (c) (d)

Fig. 10. Comparison of PSNR values and PSNR value fluctuations over frames when the same indential initial QPs with (19) are used for the four rate control methods. (a) PSNR for *BlowingBubbles* (416 × 240) @1Mb/s Proposed: Avg. PSNR = 34.37 dB, PSNR std. dev.= 0.82 HM10.0: Avg. PSNR = 34.26 dB, PSNR std. dev. = 1.04 Chen: Avg. PSNR = 33.87 dB, PSNR std. dev. = 1.28 JCTVC-H0213: Avg. PSNR = 33.99 dB, PSNR std. dev. = 1.37. (b) PSNR for *PartyScene* (832 × 480) @3Mb/s Proposed: Avg. PSNR = 30.48 dB, PSNR std. dev.= 1.43 HM10.0: Avg. PSNR = 34.32 dB, PSNR std. dev. = 1.51 Chen: Avg. PSNR = 29.90 dB, PSNR std. dev. = 2.61 JCTVC-H0213: Avg. PSNR = 29.61 dB, PSNR std. dev. = 2.90. (c) PSNR for *RaceHorses* (832 × 480) @2Mb/s Proposed: Avg. PSNR = 33.37 dB, PSNR std. dev. = 0.96 HM10.0: Avg. PSNR = 33.25 dB, PSNR std. dev. = 0.87 Chen: Avg. PSNR = 32.31 dB, PSNR std. dev. = 1.28 JCTVC-H0213: Avg. PSNR = 32.32 dB, PSNR std. dev. = 1.14. (d) PSNR for *BasketballDrive* (1920 × 1080) @8Mb/s Proposed: Avg. PSNR = 37.51 dB, PSNR std. dev. = 1.05 HM10.0: Avg. PSNR = 37.40 dB, PSNR std. dev. = 1.04 Chen: Avg. PSNR = 36.84 dB, PSNR std. dev. = 1.34 JCTVC-H0213: Avg. PSNR = 36.95 dB, PSNR std. dev. = 1.17.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. PSNR and visual quality comparisons during encoding. (a) Proposed rate control (30.68 dB) (*BlowingBubbles*, 416 × 240, 42th frame). (b) HM10.0 rate control (30.19 dB) (*BlowingBubbles*, 416 × 240, 42th frame). (c) Chen's method [13] (29.62 dB) (*BlowingBubbles*, 416 × 240, 42th frame). (d) JCTVC-H0213 [24] (29.98 dB) (*BlowingBubbles*, 416 × 240, 42th frame). (e) Proposed rate control (28.88 dB) (*PartyScene*, 832 × 480, 20th frame). (f) HM10.0 rate control (28.55 dB) (*PartyScene*, 832 × 480, 20th frame). (g) Chen's method [13] (26.54 dB). (*PartyScene*, 832 × 480, 20th frame). (h) JCTVC-H-213 [24] (26.99 dB) (*PartyScene*, 832 × 480, 20th frame).

TABLE VII
PERFORMANCE COMPARISON WHEN RDOQ IS USED

Test sequences	Fps	Target rates (Mbps)	Proposed			HM10.0			Chen [13]			JCT-VC H0123 [24]		
			Actual rates	PSNR (dB)		Actual rates	PSNR (dB)		Actual rates	PSNR (dB)		Actual rates	PSNR (dB)	
				Avg.	Std. dev.		Avg.	Std. dev.		Avg.	Std. dev.		Avg.	Std. dev.
BlowingBubble (416×240)	50	0.75	0.74	33.14	1.15	0.75	33.03	1.51	0.77	33.12	1.40	0.77	33.08	1.32
		1.50	1.50	35.93	1.37	1.50	35.82	1.41	1.54	33.95	1.17	1.54	35.98	1.12
RaceHorses (832×480)	30	1.00	1.00	31.04	1.80	1.00	31.04	2.02	1.03	30.74	2.58	1.03	30.93	1.83
		3.00	3.00	35.05	2.16	3.00	35.04	2.19	3.08	34.84	2.27	3.08	34.85	2.41
PartyScene (832×480)	50	2.00	2.01	28.80	1.02	2.00	28.63	1.73	2.03	28.04	3.32	2.05	28.66	1.18
		4.00	4.02	31.68	1.02	4.00	31.49	2.68	4.11	31.29	3.07	4.11	31.62	1.05
ParkScene (1920×1080)	24	8.00	8.00	38.80	0.37	8.00	38.69	0.57	8.17	38.67	1.07	8.13	38.60	1.38
		12.00	12.00	40.03	0.41	12.00	39.88	0.55	12.23	39.91	1.30	12.11	39.97	1.08
Average				34.31	1.16		34.20	1.58		33.82	2.02		34.21	1.42

in maximum against the conventional methods. Hence, the proposed rate control scheme is more effective than the other three methods for hierarchical quadtree HEVC coding structure.

We also investigate PSNR variations during encoding. Since high fluctuation of frame PSNR values may cause perceptual annoying to viewers, the fluctuation of PSNR values is one of the important factors for rate control applications. Fig. 9 shows the PSNR values over frames during encoding for the proposed scheme and the other three methods. Compared to the three methods, the proposed scheme yields more stable visual quality with substantially smaller PSNR fluctuations as depicted in Fig. 9. As shown in Fig. 9(b), the proposed rate control scheme shows the 5 dB-higher PSNR value in maximum than the other three methods in the frame period between the 20th and the 30th frame of the *PartyScene* sequence. The frames in the period contain various texture characteristics with static background and highly moving objects (children and bubbles).

In order to investigate the influence of initial QP values for rate control, we use the same initial QP values in (19) for all rate control algorithms for fair comparison. The original HM 10. 0 used to the HM 10.0 rate control method according to (19), which are larger values than the initial QP values of the original HM 10.0. As a result, the PSNR values of the following frames during encoding slightly increase because the bit amounts allocated to the remaining frames become relatively increased. Nevertheless, the proposed rate control method still shows better performance in PSNR and PSNR standard deviations. It should be noticed in Fig. 10 that the PSNR values of the first frame for the original HM 10.0 rate method with the same initial QP values are different from those of the other three rate control methods. This is because the original HM 10.0 rate method modifies the Lagrange multiplier in the RDO process.

Fig. 11 shows two pairs of the reconstructed frames by the proposed rate control scheme, the HM 10.0 rate control, Chen's method and JCTVC-H0213 for the *BlowingBubbles* (416 × 240, encoded at 0.6 Mb/s) and *PartyScene* (832 × 480, encoded at 3 Mb/s) sequences. As compared in terms of PSNR values in Fig. 9, the visual qualities of the reconstructed frames by the proposed rate control scheme are conspicuously better than those by the other three methods. In Fig. 11, the conventional methods show reduced PSNR values due to a limited texture and nontexture rate estimation capability. On the other hand, the proposed rate control scheme effectively works especially for the test sequences with various texture types where the quadtree coding structure of HEVC is very advantageous.

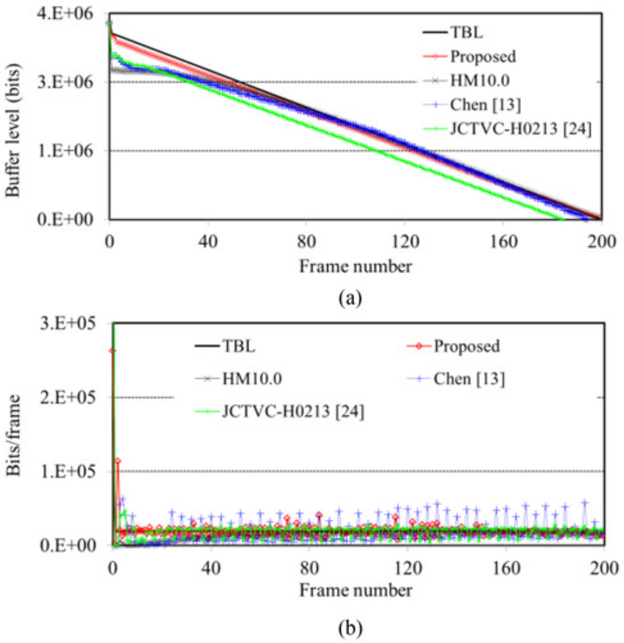


Fig. 12. Coded number of bits for each frame for *PartyScene* sequence. (a) Frame by frame buffer levels for *PartyScene* sequence encoded at 1Mb/s. (b) Number of coding bits during encoding.

Fig. 12 shows 1) buffer status levels and 2) encoded bits per frame for the proposed rate control scheme, the HM 10.0 rate control scheme, Chen's method and JCTVC-H0213 for the *PartyScene* sequence at a target bitrate of 1 Mb/s. Due to the high accuracy of buffer control and bitrate estimations, the proposed rate control scheme outperforms the other three methods. The buffer status level of the proposed rate control scheme is much closer to the target buffer level (TBL) than those of the HM 10.0 rate control scheme, Chen's method and

TABLE VIII
COMPLEXITY ANALYSIS (200 FRAMES ARE TESTED)

Sequences	Enc. Time (sec.)	Proc. Time (sec.)	Proc. Time (%)
BlowingBubbles	270.6	2.1	0.78%
RaceHorses	1141.4	21.1	1.85%
PartyScene	880.1	7.7	0.87%
Video3	1301.9	7.4	0.57%
ParkScene	3130.6	31.9	1.02%

JCTVC-H0213. As shown in Fig. 12(a) and (b), the HM 10.0 rate control scheme and Chen's method show the overflow risk between 0th frame and 40th frame due to excessive bit consumption in beginning stage of encoding. On the other hand, the proposed rate control scheme shows more stable and accurate generation of output bits than those of the other three methods.

In HEVC, the rate-distortion optimized quantization (RDOQ) [29] is adopted as an encoding scheme. We investigate rate control performances when RDOQ is used. When RDOQ scheme is used, we also set the rounding offset value to 1/6 for intercoded CU and 1/3 for intracoded CU in (10), which are the same as the RDOQ OFF case. Table VII shows the performance comparisons when RDOQ is used. As shown in Table VII, the proposed method outperforms the conventional methods in PSNR and PSNR standard deviation. The proposed rate control scheme yields 0.11, 0.49, and 0.1 dB higher PSNR values than the HM 10.0 rate control, Chen's methods and JCTVC-H0213 in average, respectively. In addition, proposed rate control scheme yields the standard deviation values of PSNR with 0.42, 0.86 and 0.26 points less than the other three methods.

In order to investigate the additional complexity for the proposed rate control scheme, we performed experiments under the simulation environments of Intel Core i-7 CPU @3.40 GHz with 8.0 GB memory, and 64 bit Windows 7 operating system. Table VIII shows the total encoding times and additional processing times for the proposed rate control scheme. As shown in Table VIII, additional processing times for the proposed scheme are negligible, which range from 0.57% to 1.85% since all processes for rate control are performed in a frame level.

V. CONCLUSION

In this paper, a frame-level rate control scheme is proposed based on novel rate models for texture and nontexture data for HEVC codecs. Motivated from the fact that signal characteristics with respect to the CU depth levels are considerably different, we categorized the CU depth levels into three types of low-, medium-, and high-texture categories. The proposed texture rate model takes well into account the different statistical characteristics of transform coefficient residues by using multiple Laplacian PDFs for different CU categories in various depth levels of HEVC, producing considerably accurate rate estimation performances. In addition, a nontexture rate model is proposed in which nontexture bits can precisely be estimated based on the linear relation between the total nontexture data and the dominant nontexture data in each CU category. The experimental results show that the proposed rate control scheme shows the average PSNR with 0.44 dB higher and the average PSNR standard deviation of 0.32 point lower, compared to the existing methods. In particular, the proposed rate control scheme outperforms the conventional rate control schemes for the sequences with abrupt changes in signal characteristics. In addition, the proposed rate control scheme is capable of maintaining stable buffer status levels, compared to the conventional methods.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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, Jul. 2003.

[2] B. Bross, W.-J. Han, G. J. Sullivan, J. R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Test Specification Draft 6*, JCTVC-H1004, 8th JCTVC Meeting, San Jose, CA, USA, Feb. 2012.

[3] B. Lee and M. Kim, "Modeling rates and distortions based on a mixture of laplacian distributions for interpredicted residues in quadtree coding of HEVC," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 571–574, Oct. 2011.

[4] M. ISO/IEC JTC1/SC29/WG11, *MPEG-2 Video Test Model 5*, Apr. 1993.

[5] ITU-T/SG15, *Video Codec Test Model*, TMN8, Portland, OR, USA, Jun. 1997.

[6] 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.

[7] Z. G. Li, F. Pan, K. P. Lim, G. Feng, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for JVT," JVT-G012-rl, Pattaya, Thailand, Mar. 2003.

[8] Y. Liu, Z. G. Li, and Y. C. Soh, "A novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 1152–1162, Jan. 2007.

[9] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for .264/AVC video coding and its application to rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1533–1544, Dec. 2005.

[10] J. Dong and N. Ling, "A context-adaptive prediction scheme for parameter estimation in H.264/AVC macroblock layer rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1108–1117, Aug. 2009.

[11] M. Jiang and N. Ling, "On enhancing H.264/AVC video rate control by PSNR-based frame complexity estimation," *IEEE Trans. Consum. Electron.*, vol. 51, no. 1, pp. 281–286, Feb. 2005.

[12] M. Jiang and N. Ling, "Low-delay rate control for real-time H.264/AVC video coding," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 467–477, Jun. 2006.

[13] J.-Y Chen, C.-W Chiu. C.-L. Li, and M.-J. Chen, "Burst-aware dynamic rate control for H.264/AVC video streaming," *IEEE Trans. Broadcast.*, vol. 57, no. 1, pp. 89–93, Mar. 2011.

[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] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Rate-distortion modeling for H.264/AVC encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 530–543, May 2007.

[16] N. Kamaci, Y. Altunbasac, and R. M. 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.

[17] S. S. Rodriguez, O. D. Esteban, M. Lopez, and F. Maria, "Cauchy-density-based basic unit rate controller for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 8, pp. 1139–1143, Aug. 2010.

[18] Z. He and S. K. Mitra, "Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 840–848, Oct. 2002.

[19] S. Milani, L. Celetto, and G. A. Mian, "An accurate low-complexity rate control algorithm based on (ρ, E_q)-domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 257–262, Feb. 2008.

[20] Q. Xu, X. Lu, Y. Liu, and C. Gomila, "A fine rate control algorithm with adaptive rounding offsets (ARO)," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 19, no. 10, pp. 1424–1435, Oct. 2009.

[21] M. Jiang and N. Ling, "On Lagrange multiplier quantizer adjustment for H.264 frame-layer video rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 663–669, May 2006.

[22] S. Zhou, J. Li, J. Fei, and Y. Zhang, "Improvement on rate-distortion performance of H.264 rate control in low bitrate," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 8, pp. 996–1006, Aug. 2007.

[23] X. Jing, L.-P. Chau, and W.-C. Siu, "Frame complexity-based rate-quantization model for H.264/AVC intraframe rate control," *IEEE Signal Process. Lett.*, vol. 15, no. 3, pp. 373–376, Mar. 2008.

[24] H. Choi, J. Nam, J. Yoo, D. Sim, and I. V. Bajic, "Rate control based on unified RQ model for HEVC," JCTVC-H0213, 8-th JCTVC meeting, San Jose, CA, USA, Feb. 2012.

[25] B. Li, H. Li, L. Li, and J. Zhang, *Rate Control by R-Model for HEVC*, JCTVC-K0103, 11-th JCTVC Meeting, Shanghai, China, Oct. 2012.

[26] W. W. Hines, D. C. Montgomery, D. M. Goldman, and C. M Borror, *Probability and Statistics in Engineering*, 4th ed. New York, NY, USA: Wiley, 2003.

[27] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distribution for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[28] S. R. Smoot, L. A. Rowe, and E. F. Roberts, "Laplacian model for AC DCT terms in image and video coding," in *Proc. 9th Workshop Image and Multi. Sig. Process.*, Belize City, Belize, pp. 178–179, Mar. 1996.

[29] M. Karczewicz, Y. Ye, and I. Chong, *Rate Distortion Optimized Quantization*, VCEG-AH21, 34th Video Coding Experts Group (VCEG) Meeting, Antalya, Turkey, Jan. 2008.

[30] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, "*HM Reference Software 10.0* [Online]," Available: https://hevc.hhi.fraunhofer.de/svn/svn HEVCSoftware/tags/HM-10.0.

[31] JVT of ISO/IEC MPEG and ITU-T VCEG, "*JM Reference Software* [Online],". Available: http://iphome.hhi.de/suehring/tml/.

[32] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, ITU-T SG16 Q.6 Document, VCEG-M33, Apr. 2001.

Bumshik Lee (M'07) received the B.S. degree in electrical engineering from Korea University, Seoul, Korea, in 2000, and the M.S. and Ph.D. degrees in information and communications engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006 and 2012, respectively.

He is a Post-Doctoral Scholar with the University of California, San Diego, CA, USA. In 2012, he was a Post-Doctoral Researcher with KAIST. His research interests include rate-distortion modeling and rate control for High Efficiency Video Coding, visual quality assessment, perceptual-based video coding, transform coding for ultra high-definition video, and pattern recognition.

Munchurl Kim (M'99) received the B.E. degree in electronics from Kyungpook National University, Daegu, Korea, in 1989, and the M.E. and Ph.D. degrees in electrical and computer engineering from University of Florida, Gainesville, FL, USA, in 1992 and 1996, respectively.

He was with Electronics and Telecommunications Research Institute, Daejeon, Korea, as a Senior Research Staff Member, where he led the Realistic Broadcasting Media Research Team. In 2001, he joined the School of Engineering, Information and Communications University, Daejeon, as an Assistant Professor. In 2009, he joined the Department of Electrical Engineering (EE), Korea Advanced Institute of Science and Technology, Daejeon, as an Associate Professor, where he is currently a Professor. He has been engaged in scalable video coding and High Efficiency Video Coding in JCT-VC standardization activities of ITU-T VCEG and ISO/IEC MPEG. His research interests include 2-D/3-D perceptual video coding, visual quality assessments on 3-D/UHD video, machine learning, and pattern recognition.

Truong Q. Nguyen (F'05) is a Professor with the Electrical and Computer Engineering Department, University of California, San Diego, CA, USA. He is a co-author (with Prof. Gilbert Strang) of a popular textbook, *Wavelets and Filter Banks* (Wellesley-Cambridge, 1997), and the author of several MATLAB-based toolboxes on image compression, electrocardiogram compression, and filter bank design. He has over 400 publications. His research interests include 3-D video processing and communications and their efficient implementation.

Prof. Nguyen received the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (Image and Multidimensional Processing Area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks in 1992. He received the NSF Career Award in 1995 and is currently the Series Editor (Digital Signal Processing) for Academic Press. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1994 to 1996, the IEEE SIGNAL PROCESSING LETTERS from 2001 to 2003, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1996 to 1997, and from 2001 to 2004, and for the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2004 to 2005.