Received January 5, 2017, accepted February 7, 2017, date of publication March 8, 2017, date of current version August 8, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2676125

A New Rate Control Scheme For Video Coding Based On Region Of Interest

ZHEWEI ZHANG[1], TAO JING[1], (Member, IEEE), JINGNING HAN[2], YAOWU XU[2], AND FAN ZHANG[1]

[1]School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]Google Inc., Mountain View, CA 94043, USA

Corresponding author: Tao Jing (tjing@bjtu.edu.cn)

ABSTRACT The quality fluctuation of video is significant in human visual system, and thus, many rate control schemes are widely developed in the area of video communication. In recent years, researchers show more interests in region of interest (ROI)-based encoding, and it is widely applied in the latest video codecs, such as HEVC and VP9. This paper presents a new rate control scheme for ROI mode coding based on discrete fourier transform coefficient model and radial basis function neuron network. A new R-D model is proposed by classifying blocks into different depth, ROI groups, and so on. Then, rate and distortion are described based on the Laplacian distribution model using mathematical ways. A machine learning approach is induced to enhance the accuracy of the distortion estimation. By utilizing the new R-D model, a new rate control scheme is designed for ROI mode coding from the group of picture layer to coding unit layer. By comparisons with other rate control approaches, the proposed one has a better result in terms of visual quality, R-D performance, bitrate accuracy, and so on. Hence, it outperforms the conventional schemes especially for sequences with obvious ROI details.

INDEX TERMS Video coding, VP9, HEVC, region of interest, rate control, Laplacian distribution model, rate-optimization model, radial basis function neuron network, machine learning.

I. INTRODUCTION

In recent years, there is a rapid development of efficient video compression techniques considered by multimedia community. One of the state-of-art video coding standard developed by Joint Collaborative Team on Video Coding (JCTVC) is called High Efficient Video Coding (HEVC) [1], [2], whose coding efficiency performs almost two times than H.264/AVC [3]. Another state-of-art one developed by Google is called VP9 codec [4], [5]. Both of them involve more sophisticated coding features and techniques than the previous coding standards. In HEVC, more flexible hierarchical-block based coding unit (CU) is designed with quadtree partitions and higher depth levels, together with prediction unit (PU) and transform unit (TU). A coding tree block (CTB) separated PU and CU from its maximum to minimum size, and the PU in a CU block can have various sizes like $2N \times 2N$, $2N \times N$, and $N \times 2N$ for $2N \times 2N$ CU block. As a convention, CU_k denotes the CU in depth k. Hence, when CTB size is set to 64×64 with the maximum depth level of four, CU_0, CU_1, CU_2 have their resolutions as 64×64, 32×32, 16×16 respectively. Note that the coding structure in VP9 is nearly the same as HEVC however with their names distinguished [6], for example, the largest CU (LCU) in VP9 is named as super-block (SB).

With the rapid development of computer vision, people attempt to focus the region of interest (ROI) in video sequences. Previous works show that researchers intend to define the concept of ROI based on Human Visual System (HVS) [7]. Meanwhile, ROI based video encoding is adopted both in H.263 [8], [9], and the notion of object-based encoding proposed in MPEG-4 [10] focuses on ROI to cope with HVS. During the recent decade, researchers regard moving objects as ROI so that they put emphasis on moving objects and fore-ground/background detection. Hence, many state-of-art techniques are developed for ROI detection such as Gaussian Mixture Model (GMM) [11]–[13], Low-Rank modeling [14] with Robust Principle Component Analysis (RPCA) [15]–[18], and background model-based approaches [19]–[23]. ROI-based encoding is proposed in [24], where CUs belong to ROI group have smaller quantization parameters (QP) than CUs in non-ROI group, for the sake of improving visual quality for ROI blocks. Also, ROI-based encoding is widely used both in H.264 and H.265 standards [25], [26]. Since the objective of

video coding is to obtain the best reconstructed quality, rate control is an important issue for ROI-based encoding.

A. RELATED WORKS

Above the current researches acknowledgements, there are three typical models for rate control: $R-Q$ model [27], $R-\lambda$ model [28], and $R-\rho$ model [29]. In R-Q model, rates and distortion can be formulated as a function of QP separately. This model is first developed for hybrid video and it is well fitted in H.264 [24], then it is extended in ROI coding mode in [30]. Choi *et al*. [31] proposed a pixel-wise unified $R-Q$ model (URQ) implemented in HM6.0, however, no-residue information is not considered in this model. The $R-\rho$ model is a ρ-domain model based on the percentage of non-zero quantized transformed coefficient. It is implemented as a Rate-Gop method [32] in HEVC, where QP is determined by the picture order count (POC). Lee and Kim [33], Lee *et al* [34] models the parameters of ρ-domain separately for CUs with different depth and frames with various types. Also, they propose a texture and non-texture model where the bits estimation for texture and nontexture blocks are computed separately, and the scheme is also suited for ROI coding mode as well. Their experiments reveals that separately modeled ρ-domain has a better performance in HEVC, however, this is not the well-addressed solution for ROI-based mode. $R-\lambda$ model is well studied in HEVC and AVC [35], which focuses on Rate Distort Optimization (RDO). Therefore, larger λ causes larger distortion and fewer bit outputs. [36] shows that their improved $R-\lambda$ model considers both residual and nonresidual bits, and it outperforms other approaches in terms of the estimation precision. However, Wang *et al.* [37] point out the disadvantages of $R-\lambda$ is that it only considers the target bits but neglects the characteristics of video data. Therefore, we believe the performance can be improved by adding the characteristic information hierarchically based on ROI coding mode. In addition, other models based on discrete fourier transform (DCT) coefficients PDF deduce the relationship between bits output and the entropy functions [38]–[41]. These approaches achieves a better performance in accuracy at the expense of bringing computing costs.

B. OUR METHOD AND CONTRIBUTIONS

In this paper, we give a new idea of modeling the bits output by inducing a mixture Laplacian DCT (MLD) coefficient distribution to represent CU's rate and distortion by their properties (i.e., depth, ROI/non-ROI group). A frame-level target bits estimation model is formed theoretically based on ROI mode. Meanwhile, in the CU layer, a deep learning approach is designed by creating artificial neuron networks (ANN) to measure the accurate distortion, and the performance of ANN model has a better estimation on distortion than the theoretical model via experiments. Next, a new rate control scheme for based on ROI-mode is given by solving the R-D optimization problem. Moreover, the proposed rate control scheme operates from GOP level to CU level. Besides, a new GOP selection and frame bit allocation method has been proposed. Based on the proposed rate models and control scheme, more accurate rate estimation is achieved with the minimum distortion. Different from other approaches, the proposed one first cluster GOP by histogram of differences (HOD) judgements. Second, it updates a neuron network to get the precise distortion periodically. Meanwhile, the accumulated distortion is stored in a buffer to adaptively update QP by threshold so that a feedback control approach is used.

The remainder of this paper is organized as follows. In Section. II, the proposed R-D model is illustrated using mathematical ways. In Section. III, the proposed rate control method for ROI mode coding is given in GOP layer, frame layer and CU layer. Experimental results are displayed in Section. IV, which contain R-D performance, quality comparisons, etc. Finally, we conclude the paper in Section. V.

II. ROI-BASED RATE-DISTORTION MODELING

A. ROI RATE MODELING

By definition of ROI-based coding in HEVC, CUs are categorized into two basic groups: group of ROI and group of non-ROI. The two basic groups are separated further based on the coding depth of CUs. They are classified into two levels: low-textured and high-textured. The extent of 'textured' indicates the coding texture information levels of CUs. For example, CU_0 and CU_1 have the lower variances in DCT coefficients than CU_2 and CU_3. Hence, CU_0 and CU_1 are classified as CU_L and CU_2 and CU_3 are regarded as CU_H. Note that all of them belong to the basic two groups. According to the [42], the histogram of DCT coefficients are well fitted by Laplacian PDF with a confidence interval 95%. Thus, we model the source distributions of the transform residues for each category as :

$$f_c(l) = \frac{\lambda_c}{2} e^{-\lambda_c |l|}, \lambda_c \in \{\lambda_l, \lambda_h\}, \tag{1}$$

where λ_l and λ_h are model parameters for CUs in level l and h, l is a random variable for transform coefficient. λ_c is computed as [38]:

$$\lambda_c = \sqrt{2}/\sigma_c, \tag{2}$$

where σ_c is standard deviation of the residual transform coefficients for different levels of CUs in a frame. Since QP for CUs in ROI group is smaller than CUs in non-ROI group, λ_l and λ_h are different in these two groups. Depending on this property, a proposed frame-level rate model for ROI-mode coding for transform coefficients in HEVC is given by:

$$R(q) = \sum_{k \in \{r,n\}} \alpha_l N_{lk} H_{lk}(q_k, \lambda_l) + \alpha_h N_{hk} H_{hk}(q_k, \lambda_h), \tag{3}$$

where k belongs to the set of ROI and non-ROI groups. $H_c(q, \lambda_c)$ represents the entropies function for category c ($c \in \{lr, hr, ln, hn\}$). N_c is the number of pixels for CU in category c. The sum of $N_{lr}, N_{hr}, N_{ln}, N_{hn}$ and N_{SKIP} equals to the total frame's pixel size. N_{SKIP} means the total pixels number for skip blocks in CUs. α_c is a model parameter

which can be computed via linear regression scheme [33]. The entropy function is computed as:

$$H_c(q, \lambda_c) = -P_{0,c} \log_2 P_{0,c} - 2 \sum_{i=1}^{\infty} P_{i,c} \log_2 P_{i,c}, \quad (4)$$

where q is quantization step size, $P_{i,c}$ denotes the probabilities that transform coefficients are quantized to interval i in category c. It is written by:

$$P_{i,c} = \begin{cases} \int_{-(q-fq)}^{q-fq} f_c(l)dl = 1 - e^{-(1-f)q\lambda_c}, & i = 0 \\ \int_{(i-f)q}^{(i+1-f)q} f_c(l)dl = 1/2e^{-(i-f)q\lambda_c}(1 - e^{-q\lambda_c}), & i > 0. \end{cases} \quad (5)$$

In this expression, f is a rounding offset. According to [33], f is set empirically to 1/6 for intercoded CU and 1/3 for intracoded CU. After some mathematical manipulations, (4) is simplified as :

$$H_c(q, \lambda_c) \simeq a_c(e^{-(1-f)q\lambda_c}), \quad (6)$$

where a_c is a model parameter, the proof of this is done explicitly in [34]. Finally, combining with (6) and (9), we get $R(q)$ as:

$$R(q) = \sum_{k\in\{r,n\}} \alpha_l N_{lk}(e^{-(1-f)q\hat{\lambda}_l}) + \alpha_h N_{hk}(e^{-(1-f)q\hat{\lambda}_h}). \quad (7)$$

Notice that $\hat{\lambda}_l$ and $\hat{\lambda}_h$ are the estimated model parameters in different CU categories. The whole bits output in a frame not only contains the bits from transform coefficients, but also includes bits for header, motion vectors, CUs coding mode, etc. For simplify reasons, we consider the major contributions among these bits output. Suppose R_{ext} as output bits for a frame excluded from the transform coefficients, then we have:

$$R_{ext} = \sum_{k\in\{r,n\}} N_{lk}\bar{R}_l(mv) + N_{hk}\bar{R}_h(mv) + N_{Ik}\bar{R}_h(\mathrm{I}_k), \quad (8)$$

where N_{lk} and N_{hk} are the pixel number of inter blocks for CU in different group levels, N_{Ik} denotes the pixel number of intra blocks. $\bar{R}(mv)$ means the average motion vector bits per block in CU levels and $\bar{R}(I)$ is the average bits for intra blocks of CU. Therefore, the rate model for a frame is proposed as:

$$R = R(q) + R_{ext}, \quad (9)$$

where $R(q)$ is written in (3).

As shown in Fig. 1, the histogram of DCT coefficient in each CU category is well fitted into Laplacian PDFs in various CU depth levels. The DCT coefficient follows Laplacian PDF with a confidence interval 95% or more, which proves the similar results provided in [42].

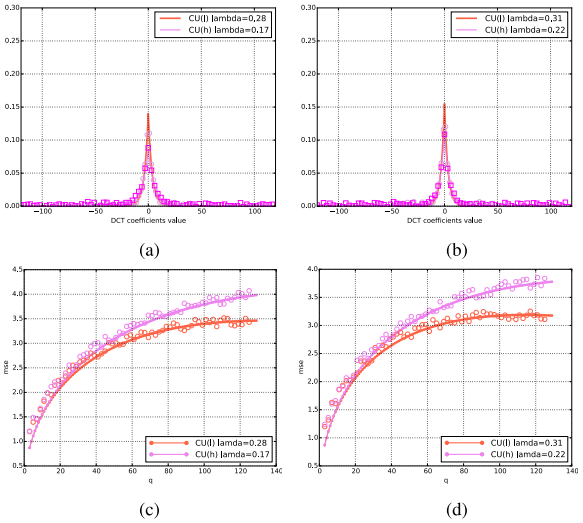


FIGURE 1. CUs DCT coefficients distribution and distortion analysis for *BasketballPass* and *RaceHorses*. (a) *BasketballPass*. (b) *RaceHorses*. (c) *BasketballPass*. (d) *RaceHorses*.

B. ROI DISTORTION MODELING

1) THEORETIC MODELING

The distortion model is designed for the purpose of measuring the relationship between rate and distortion. It is critical to find appropriate R-Q and D-Q models for the sake of making balance between rate output and distortion. However, to find an accurate D-Q model is hard since transform distribution model has errors itself. Also, the quad-tree coding structure in HEVC brings in the diversity properties of CU, thus, increasing the prediction difficulty of D-Q model. According to the discussions above, the distortion $D_j(q)$ can be computed approximately by an expression below:

$$\begin{aligned} D_j(q) &= P_{0,c} + 2\sum_{i=1}^{\infty} \int_{(i-f)q}^{(i+f)q} f_c(l)|l - iq|dl \\ &= P_{0,c} + 2\sum_{i=1}^{\infty} (\int_{(i-f)q}^{iq} f_c(l)(iq - l)dl \\ &\quad + \int_{iq}^{(i+f)q} f_c(l)(l - iq)dl). \end{aligned} \quad (10)$$

Note that j is the depth of CU and its range is default from 0 to 3 in HEVC, and i denote the quantization step index. Substituting (1) into (10), we get:

$$\begin{aligned} D_j(q, \lambda_c) &= P_{0,c} + 2\sum_{i=1}^{\infty} \int_{iq}^{(i+f)q} \frac{1}{2}\lambda_c e^{-\lambda_c l}(l - iq)dl \\ &\quad + 2\sum_{i=1}^{\infty} \int_{(i-f)q}^{iq} \frac{1}{2}\lambda_c e^{-\lambda_c l}(iq - l)dl \\ &= \sum_{i=1}^{\infty} -[fq + \frac{1}{\lambda_c}]e^{-\lambda_c(i+f)q} \\ &\quad + \frac{2}{\lambda_c}e^{-\lambda_c iq} + (fq - \frac{1}{\lambda_c})e^{-\lambda_c(i-f)q} + P_{0,c} \\ &= \{-[fq + \frac{1}{\lambda_c}]e^{-\lambda_c fq} + \frac{2}{\lambda_c} + (fq - \frac{1}{\lambda_c})e^{\lambda_c fq}\} \\ &\quad \times e^{-\lambda_c q}/(1 - e^{-\lambda_c q}) + 1 - e^{-(1-f)q\lambda_c}. \end{aligned} \quad (11)$$

Note that the purpose of deducing this expression is to convert integral to basic calculation, so that it is easy to be implemented in computer. The total distortion of a frame can be estimated as :

$$D = \sum_{k\in\{r,n\}} N_{lk} D_{lk}(q_k, \lambda_l) + N_{hk} D_{hk}(q_k, \lambda_h). \tag{12}$$

Similarly, N is the block pixel number and D_c is the distortion function in level c, where $c \in \{lr, hr, ln, hn\}$.

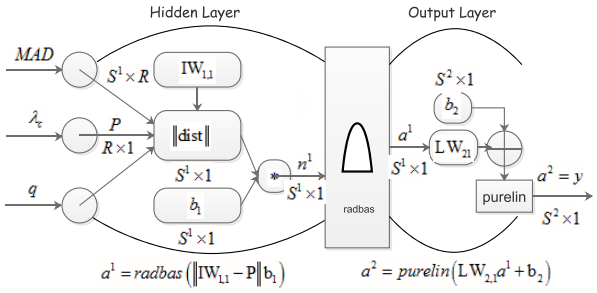


FIGURE 2. Basic structure of RBF network.

2) ANN MODELING

The discussion in *Theoretic Modeling* reveals that only approximate models can be found as a prediction for the distortion. CU's diversity and the ROI categories also enhance the challenge for encoders to make trade-off between distortion and rate output. Hence, using Artificial Neuron Network (ANN) can improve the current model and help encoders make decisions on target bits allocation. In the previous subsection, it can be seen that the D-Q model is not a linear model. Thus, an optimal choice is to induce Radial Basis Function Neuron Network (RBFNN) for non-linear model prediction. The structure of RBFNN is displayed in Fig. 2. For illustrative purpose, matrices and their dimensions are marked in this figure. The input matrix $\mathbf{P}$ and output matrix $\mathbf{T}$ have the following structure:

$$\mathbf{P} = \begin{pmatrix} \mathrm{MAD}_1 & \cdots & \mathrm{MAD}_R \\ \lambda_{c1} & \cdots & \lambda_{cR} \\ q_1 & \cdots & q_R \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} D_1, \cdots, D_R \end{pmatrix}. \tag{13}$$

Here, $\mathrm{MAD}_i, i = 1, 2, \cdots, R$ is the mean of absolute difference between the reconstruct CU and the origin one at level c. R is the number of training samples which is shown in the figure. Distortion D is treated as the output value in this network. $\mathbf{IW}_{1,1}$ and $\mathbf{LW}_{2,1}$ are weight matrices with the shape of $S^1 \times R$ and $S^2 \times S^1$ respectively, where S^1 equals to R and $S^2 = 1$ since we only have one predict value D in each samples. Threshold $\mathbf{b}_1 = [b_{11}, \cdots, b_{1S^1}]^{\mathrm{T}}$ and $b_2 = b_{21}$. For convenient reasons, any b_{1j} $(b_{1j} \in b_1)$ is set to $\frac{0.8326}{\mathrm{spread}}$, where *spread* is the spread speed of radial basis and it is usually set as 0.5. The output data a^1 from the hidden layer is computed by RBF as:

$$a^1 = \exp(-\|\mathbf{IW}_{1,1} - \mathbf{P}\|^2 b_i). \tag{14}$$

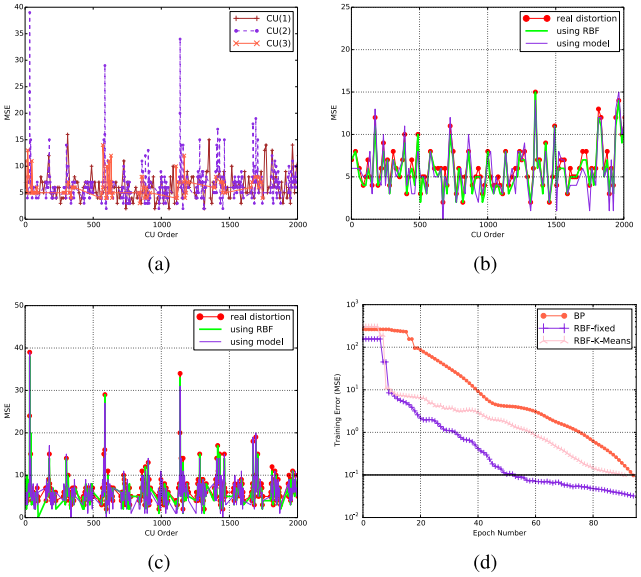


FIGURE 3. Performance comparisons between Theoretic Modeling and ANN Modeling. (a) MSE distributions for different CU levels. (b) Distortion prediction for low level of CU. (c) Distortion prediction for high level of CU. (d) Training performance of different neural networks.

Also, the matrix $\mathbf{T}$ from output layer can be expressed as:

$$\begin{aligned} \mathbf{T} &= [\mathbf{LW}_{2,1}\ \mathbf{b}_2] \bullet [\mathbf{A}; \mathbf{I}]; \\ \mathbf{T}_i &= \mathbf{LW}_{2,1} a^1 + b_2, \end{aligned} \tag{15}$$

where $\mathbf{A} = [a_1, a_2, \cdots, a_n]$, $\mathbf{b}_2 = b_{21}$, and $\mathbf{I} = [1, \cdots, 1]_{1\times n}$. By solving equation in (15), $\mathbf{LW}_{2,1}$ can be computed as:

$$\mathbf{LW}_{2,1} = (\mathbf{T} - \mathbf{b}_2\mathbf{I})\mathbf{A}^{\dagger}, \tag{16}$$

where the $\mathbf{A}^{\dagger} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$. As for the training sample parameters, λ_c and q are easy to get. MAD_i is computed as an average value to forbid abrupt change:

$$MAD_i = \frac{1}{N_f} \sum_{n=1}^{N_f} MAD_{in}. \tag{17}$$

N_f denotes the most recent frames related to CU_i in a co-located position which is set default to 3. (17) is derived from the fact that *MAD* distribution is similar in corresponding positions of the previous frames.

The performances of the two approaches of ROI distortion modeling are shown in Fig. 3. We implement this test based on HM13.0 software. Information of CUs is extracted from the *BlowingBubbles* sequences at the 8th frame. Distortion values for different CU depth (1-3) are displayed in Fig. 3(a). Observe that CUs with higher depth (i.e., size 8X8) have smaller distortion values than lower depth. In Fig. 3(b) and Fig. 3(c), modeling accuracy is evaluated by making comparisons between the two methods. For ANN approach, training samples are set to 400 CUs at the same level of the previous frame. The inputs of ANN are acquired by (2) and (17). MSE is measured as the training goal between the ANN outputs and real values, which is set to 0.1 as the target. Fig. 3(b)

$$\frac{\partial D(q_j,\lambda_c)}{\partial q_j} = \{\underbrace{[(1+f(f+1)q_j\lambda_c)e^{-\lambda_c(f+1)q_j} - 2e^{-\lambda_c q_j} + (1+f(f-1)\lambda_c q_j)e^{\lambda_c(f-1)q_j}]}_{A}(1-e^{-\lambda_c q_j}) - (\lambda_c e^{-\lambda_c q_j})$$
$$\times \underbrace{[-(fq_j+\frac{1}{\lambda_c})e^{-\lambda_c(f+1)q_j} + \frac{2}{\lambda_c}e^{-\lambda_c q_j} + (fq_j-\frac{1}{\lambda_c})e^{\lambda_c(f-1)q_j}]}_{B}\}/(1-e^{-\lambda_c q_j})^2 + \underbrace{(1-f)q_j\lambda_c e^{-(1-f)q_j\lambda_c}}_{C} \quad (18)$$

reveals the difference between modeling outputs and real distortion for CU_l (CU_h in Fig. 3(b)). After training process, RBF approach has better performance in accuracy than the Theoretic Modeling, however, with higher spatial and time complexity for training. Besides, learning efficiency is compared between BP, RBF-fixed (proposed in this paper), and RBF-k-means neuron networks. The RBF-K-Means method uses the K-Means Clustering algorithm to update the center of each hidden unit, until convergence. Learning efficiency performance in Fig.3(b) presents that the proposed RBF-fixed ANN has a better convergence speed than others, which means its learning efficiency is higher.

C. ROI R-D OPTIMIZATION

A popular problem in video rate control is to optimize the total distortion with a constrained rate available, which is named as the R-D Optimization problem. Our goal is to minimize the distortion function of a frame series in GOP. This problem is simply depicted as :

$$\min \sum_{j=0}^{\mathrm{GS}-1} D_j(q_j,\lambda_j)$$
$$s.t. \sum_{j=0}^{\mathrm{GS}-1} R_j(q_j,\lambda_j) \le R_T \quad q_j,\lambda_j > 0. \quad (19)$$

Note that GS is the number of frames in a GOP, and its value depends on the video source. A brief discussion of dynamically determining GOP is done later in this paper. D and R are rate and distort function based on q and λ respectively. In general, ROI-mode encoding allocates q_r and q_n to CUs in ROI group and non-ROI group correspondingly, so that QP is varied in a single frame and $q_r - q_n = \Delta$, where Δ is a constant value. According to the features of ROI-mode encoding, (19) can be rewritten as:

$$\min \sum_{j=0}^{\mathrm{GS}-1}\sum_{k\in\{r,n\}} N_{jlk}D(q_{jk},\lambda_{jl}) + N_{jhk}D(q_{jk},\lambda_{jh})$$
$$s.t. \sum_{j=0}^{\mathrm{GS}-1}\sum_{k\in\{r,n\}} N_{jlk}R(q_{jk},\lambda_{jl}) + N_{jhk}R(q_{jk},\lambda_{jh}) \le R_T \quad q_{jk},\lambda_{jk} > 0. \quad (20)$$

It is easy to check that this problem is a convex optimization problem by rules in [43]. However, this problem is very hard to solve since N_{lk} and N_{hk} is changing at each frames and the object function contains sub-functions with variable q and λ_c. Fortunately, λ_c can be acquired or updated based on (2) or previous encoded frames. We attempt to approximately simplify this problem. To begin, the Lagrange function $L(q,\mu)$ is written below:

$$L(q,\mu) = \inf_q \sum_{j=0}^{\mathrm{GS}-1} D_j(q_j,\lambda_j) + \mu(\sum_{j=0}^{\mathrm{GS}-1} R_j(q_j,\lambda_j) - R_T). \quad (21)$$

By computing partial derivation with respect to q_{jk}, we have:

$$\frac{\partial L(q,\mu)}{\partial q_{jk}} = \sum_{\substack{c\in\{l,h\}\\ k\in\{r,n\}}} N_{jck}(\frac{\partial D(q_{jk},\lambda_{jc})}{\partial q_{jk}} - \mu\frac{\partial R(q_{jk},\lambda_{jc})}{\partial q_{jk}}). \quad (22)$$

Using (11) and (6), $\frac{\partial D(q_j,\lambda_c)}{\partial q_j}$ can be expressed in (18), as shown at the top of the this page. Since λ_c is quite small, $(\lambda_c e^{-\lambda_c q_j}) \times B$ and C approximately equals to zero. Thus, we have the following result:

$$\frac{\partial D(q_j,\lambda_c)}{\partial q_j} - \mu\frac{\partial R(q_j,\lambda_c)}{\partial q_j} = \frac{A}{1-e^{-\lambda_c q_j}} - \mu(1-f)\lambda_c e^{-(1-f)q_j\lambda_c}a_c = 0. \quad (23)$$

By solving this equation, q_j can be acquired. However, the Lagrange multiple parameter μ is unavailable. A theoretic way to compute μ is using the Karush-Tuhn-Tucker (KKT) condition of the problem in (20):

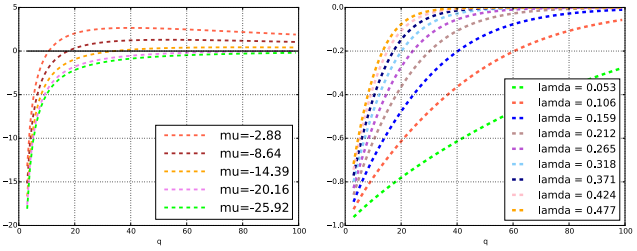
$$\begin{cases} \mu(\sum_{j=0}^{\mathrm{GS}-1}\sum_{\substack{c\in\{l,h\}\\ k\in\{r,n\}}} a_{jc}e^{-(1-f)\lambda_{jc}q_{jk}} \times N_{jck} - R_T) = 0.[\mathrm{I}] \\ q_{jk} > 0, \lambda_{jc} > 0, a_{jc} > 0, j = 0,\cdots,\mathrm{GS}-1.[\mathrm{II}] \\ (23).[\mathrm{III}] \end{cases} \quad (24)$$

Substituting the condition I of (24) into (23), a new expression is get:

$$\sum_{j=0}^{\mathrm{GS}-1}\sum_{\substack{c\in\{l,h\}\\ k\in\{r,n\}}} N_{jck}(\frac{A(\lambda_{jc},q_{jk})}{1-e^{-\lambda_{jc}q_{jk}}}) - \mu(1-f)\tilde{\lambda}_{jc}R_T = 0. \quad (25)$$

Here, we rewritten A in (18) as $A(\lambda_{jc},q_{jk})$ since A is a function of λ_{jc} and q_{jk}. $\tilde{\lambda}_{ck}$ is treated as average weight factor, and it is computed by this formula:

$$\tilde{\lambda}_{ck} = \frac{\sum_{j=0}^{\mathrm{GS}-1}\sum_{\substack{c\in\{l,h\}\\ k\in\{r,n\}}} N_{jck}\lambda_{jc}}{\mathrm{GS}\times\tilde{N}}, \quad (26)$$



(a) (b)

FIGURE 4. (a) Solutions of q in (23). (b) Boundary for $fun(q, \lambda)$ in (31).

where $\bar{N}$ equals to frame width $\times$ frame height. Rearranging (25), μ is computed as:

$$\mu = \frac{\sum_{j=0}^{\mathrm{GS}-1} \sum_{\substack{c\in\{l,h\}\\ k\in\{r,n\}}} N_{jck}\left(\frac{A(\lambda_{jc},q_{jk})}{1-e^{-\lambda_{jc}q_{jk}}}\right)}{(1-f)\tilde{\lambda}_{jc}R_T}. \tag{27}$$

By substituting (27) into (23), q_{jk} can be gained by solving the equation. However, it is a transcendental equation which is hard to solve, and an approximate way is proposed in the following discussion. By expanding (23), we have:

$$[f(1+f)q_j\lambda_c+1]e^{-\lambda_c(f+1)q_j} + [1+f(f-1)q_j\lambda_c]e^{\lambda_c(f-1)q_j} - 2e^{-\lambda_c q_j} = (1-e^{-\lambda_c q_j})\mu(1-f)a_c\lambda_c e^{-(1-f)q_j\lambda_c}. \tag{28}$$

Multiplying $e^{\lambda_c q_j}$ on both sides, the equation becomes:

$$[f(1+f)q_j\lambda_c+1]e^{-\lambda_c f q_j} + [1+f(f-1)q_j\lambda_c]e^{\lambda_c f q_j} - 2 = \underbrace{(e^{\lambda_c q_j}-1)}_{X}\mu(1-f)a_c\lambda_c e^{-(1-f)q_j\lambda_c}. \tag{29}$$

Observe that X can be simplified as $e^{\lambda_c q_j}$ since $e^{\lambda_c q_j} \gg 1$. After some manipulation steps, we get:

$$\underbrace{[1+f(f+1)q_j\lambda_c]e^{-2\lambda_c q} - 2e^{-\lambda_c q_j}}_{fun(q,\lambda)} + [1+f(f-1)q_j\lambda_c] = \mu(1-f)a_c\lambda_c. \tag{30}$$

Now, we find that $fun(q, \lambda)$ is within a range of (-1,0). The proof is simple. In fact, (30) has the following feature:

$$[1+f(f+1)q_j\lambda_c]e^{-2\lambda_c q_j} - 2e^{-\lambda_c q_j} \le \underbrace{e^{-\lambda_c q_j}(e^{-\lambda_c q_j}-2)}_{Y}. \tag{31}$$

Observe that the derivative of Y is positive, so that Y is monotonic increasing. Taking q_j from 0 to ∞, the boundary of Y is [-1,0]. Fig. 4(a) displays the solutions of quantization steps q from the given equation (23) with varying μ. The intersection points between curves and horizontal zero line are treated as solutions. Fig. 4(b) validates the proof that $fun(q, \lambda)$ is within a range of -1 to 0, and it is seen clearly through changing value of λ_c. According to this, (31) can be approximately written as:

$$q_j = \frac{\mu(1-f)a_c\lambda_c - O}{f(f-1)\lambda_c}, \tag{32}$$

where O is a constant offset whose range is [0,1]. In practical, it is set as 0.8 nearly to the mean of $fun(q, \lambda) + 1$. By using the similar calculation in (32), q_j can be estimated although with some turbulence, then it is adjusted by a control strategy in the next section.

III. ROI-BASED RATE CONTROL SCHEME

A. GOP LEVEL RATE CONTROL

In ROI-based coding procedure, an intuitive approach is to cluster similar topics of sequences into one GOP. In other words, frames with the same ROI elements are allocated in the same GOP so that GOP size is changeable, which is different from fixed GOP size in HEVC. However, the coding structure of GOP must be the same, and it is initialized in the config file of the encoder. Observe that an important thing is to cluster GOP size before encoding each frames, and a way is to distinguish frames by computing Histogram Of Difference (*HOD*) referred from [8]:

$$HOD(f_n, f_m) = \sum_{i=0}^{level} \frac{|hist(f_n, i) - hist(f_m, i)|^2}{N_{pixel}}, \tag{33}$$

where f_n, f_m are frame index. *level* denotes the histogram level, and it is set to 255 as default. *hist* is the histogram value, and N_{pixel} equals to the total number of pixels in a frame. We state a cluster method based on *HOD* below:

Algorithm 1 GOP-Level Cluster and Rate Control Algorithm

1: Set the minimum GOP structure unit size S_{GOP}, $cnt = 1$, $H_{prev} = 0$ and threshold th.
2: **for** each frames f_i **in** encode-frame buffer **do**
3: Compute $H_{cur} = HOD(f_i, f_{i-1})$ using (33).
4: **if** $i > 2$ **and** $\frac{H_{cur}-H_{prev}}{H_{cur}} > th$ **then**
5: $GOP[cnt] = S_{GOP} \times \lceil i/S_{GOP} \rceil$.
6: Allocate R_T based on (35).
7: Skip $f_{i+1} \cdots GOP[cnt]$ **if** $GOP[cnt] \ge f_{i+1}$.
8: cnt++, and $H_{prev} = H_{cur}$.
9: **end if**
10: **end for**

Observe that vector $GOP[cnt]$ stores the cluster size of each GOP, and the size must equal to $n \times S_{GOP}$, where n is a positive integer. Clustering is implemented by computing the gradient of *HOD*. When detecting abrupt changing, a new GOP size is added to $GOP[cnt]$ and target bits are allocated for this GOP depending on its size and texture complexity. For texture complexity, define γ as the percentage of ROI simply as:

$$\gamma = \frac{N_{ROI}}{N_{pixel}}, \tag{34}$$

where N_{ROI} is the number of pixels belonging to ROI in a frame, and the ROI-percentage for each sequences is plotted in Fig. 5. Suppose R_u is the target bit rate set by users, then R_T for $GOP[cnt]$ is computed as:

$$R_T = \frac{R_u}{fps} S_{GOP}(\bar{\gamma} + \delta). \tag{35}$$

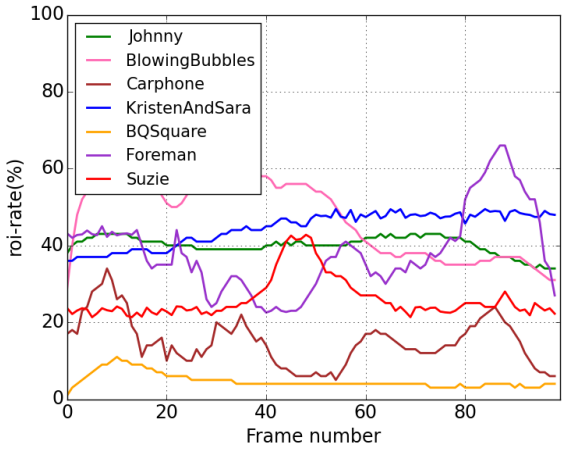


FIGURE 5. ROI-percentage for different video sequence.

TABLE 1. Basic configurations of HM13.0 and VP9.

Configurations (H-M13.0)	Values	Configurations (VP9)	Values
Profile	Main	Profile	Main
CTU Size/Depth	64/4	SB Size/Depth	64/4
Maximum GOP Size	20	Maximum GOP Size	20
TransformSkip	1	TransformSkip	1
NumTileMinus1	1	TileNumbers	4
RateControl	1	end-usage	vbr
SAO Enable	1	Two Pass	2
IntraPeriod	15	kf-max-dist	15

δ represents a constant offset (0.75 as default), and $\bar{\gamma}+\delta$ varies from 0.75 to 1.75. We aim to increase target bits distribution for larger $\bar{\gamma}$ and reduce it for smaller $\bar{\gamma}$, so that R_T depends on both GOP size and ROI texture complexity. This allocation strategy is reasonable since it puts more weight on GOP with more ROI elements to meet human visual quality demands.

B. FRAME LEVEL RATE CONTROL

Once determining GOP sizes and GOP target bits, it is ought to consider to what amount of bits we should set for each frame. To handle this, a key point is how to update parameters such as a_c, λ_c ($c \in \{l, h\}$), and μ in our ROI rate model. Like other works, they are updated via history results in order to forbid high fluctuation during encoding. At the beginning frame of each GOP, λ_c is initialized from (2) as an average value:

$$\lambda_c = \frac{1}{B_c}\sum_{i=1}^{B_c}\frac{\sqrt{2}}{\sigma_{ci}}. \tag{36}$$

B_c is the number of CUs in category c. When encoding frames in GOP, λ_c is updated by the following formula:

$$\lambda_{c,n} = \frac{1}{N_{re}}\sum_{i=1}^{N_{re}}\omega_i\lambda_{c,n-i}, \tag{37}$$

where N_{re} denotes recent frames encoded before the current one, n is the frame index and ω_i is a weight factor that

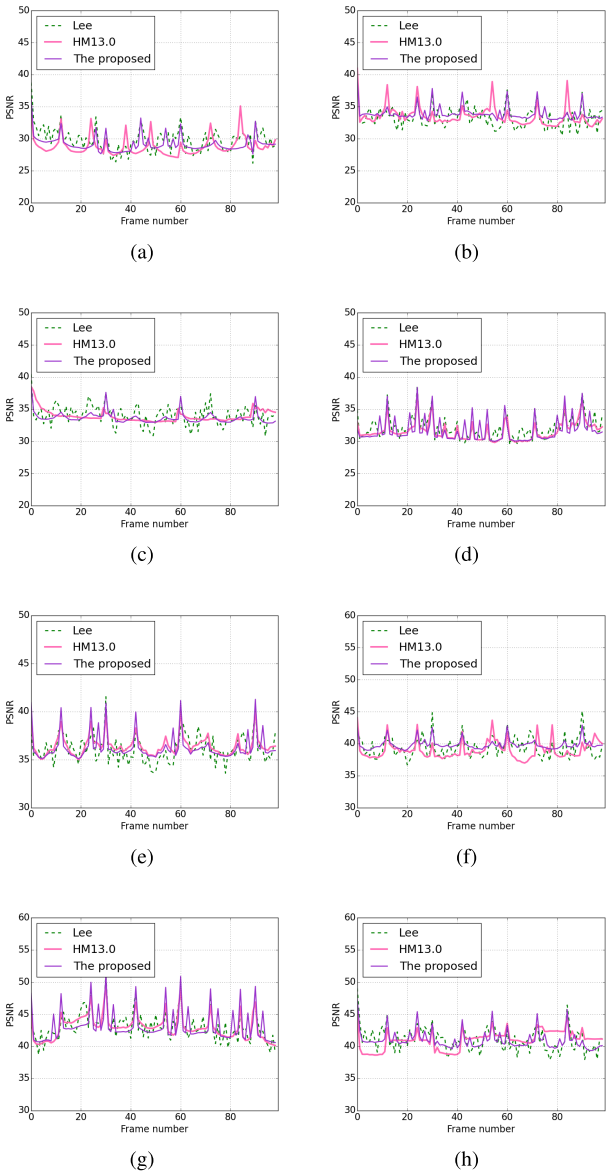


FIGURE 6. PSNR curves per frames for different rate control schemes. **(a)** ***PartyScene.*** **(b)** ***RaceHorse.*** **(c)** ***BQSquare.*** **(d)** ***BlowingBubbles.*** **(e)** ***City.*** **(f)** ***Ice.*** **(g)** ***Stephan.*** **(h)** ***Suzie.***

depends frame categories (i.e., I,P,B). a_c is initialized as empirically as a random value between 0.75 and 1.75. During the encoding procedure, it is updated by linear regression referred from [33]:

$$a_c = \frac{\sum_{i=0}^{N-1} R_{c,n-i}\hat{R}_{c,n-i} - \frac{1}{N}(\sum_{i=0}^{N-1} R_{c,n-i})(\sum_{i=0}^{N-1}\hat{R}_{c,n-i})}{\sum_{i=0}^{N-1}\hat{R}^2_{c,n-i} - \frac{1}{N}(\sum_{i=0}^{N-1}\hat{R}_{c,n-i})^2}. \tag{38}$$

Note that N equals to the frame number counted at the beginning of current GOP, meanwhile, $R_{c,n-i}$ and $\hat{R}_{c,n-i}$ are real bits output and estimate bits without R_{ext} (8) for this frame respectively. Given an initial value of a_c, it is updated and

TABLE 2. R-D performance In ROI encoding mode.

Sequences	Proposed vs HM13.0		Proposed vs Lee		Proposed vs VP9	
	BD-PSNR(dB)	BD-BR	BD-PSNR(dB)	BD-BR	BD-PSNR(dB)	BD-BR
Traffic	0.44	-4.52%	0.33	-2.17%	0.51	-4.96%
BQTerrace	0.38	-4.01%	0.41	-5.01%	0.47	-5.82%
KristenAndSara	0.28	-2.26%	0.31	-3.07%	0.33	-3.52%
Jonny	0.26	-2.13%	0.29	-2.92%	0.32	-3.34%
PeopleOnStreet	0.39	-3.42%	0.46	-4.93%	0.35	-2.88%
ChinaSpeed	0.03	-0.70%	-0.03	1.12%	-0.01	0.07%
Average	**0.30**	**-2.83%**	**0.30**	**-2.83%**	**0.33**	**-3.41%**
RaceHorses	0.10	-3.02%	0.13	-3.28%	0.07	-2.21%
BlowingBubbles	0.04	-1.86%	0.09	-2.77%	0.01	-1.13%
PartyScene	0.18	-3.99%	0.15	-3.04%	0.22	-4.21%
BQSquare	0.16	-3.79%	0.21	-4.02%	0.14	-2.89%
Average	**0.12**	**-3.17%**	**0.15**	**-3.30%**	**0.11**	**-2.61%**
Stephan	0.15	-3.57%	0.13	-3.01%	0.17	-4.03%
Carphone	0.12	-3.26%	0.06	-1.42%	0.09	-2.67%
Mobile	-0.02	0.97%	0.01	-0.39%	-0.01	0.36%
Suzie	0.26	-4.34%	0.36	-5.83%	0.29	-3.99%
Average	**0.13**	**-2.55%**	**0.14**	**-2.66%**	**0.14**	**-2.58%**

regressed when encoding each frame as iterations. A tough problem is how to update μ instantaneously. Suppose we are encoding frame j now, according to equation (27), the remaining λ_c and q is unavailable. To solve this 'egg and chicken' problem, we state a strategy for updating μ by:

$$\mu_{cnt} = \frac{N-i}{N}\bar{\mu}_{cnt-1} + \frac{i}{N}\frac{\sum_{j=0}^{i-1}\sum_{\substack{c\in\{l,h\}\\k\in\{r,n\}}} N_{jck}\left(\frac{A(\lambda_{jc},q_{jk})}{1-e^{-\lambda_{jc}q_{jk}}}\right)}{(1-f)\tilde{\lambda}_{jc}\sum_{j=0}^{i-1}R_j}, \tag{39}$$

where N represent recent frame size, and $\bar{\mu}_{cnt-1}$ means the previous GOP 's μ computed by (27). $\sum_{j=0}^{i-1}R_j$ represents the accumulated output bits from the start frame of current GOP to the current frame i. In fact, there is little difference of μ between previous and current GOP. Hence, it is reasonable to use a decay weight to estimate current value of μ. In summary, a frame-level rate control strategy for ROI mode is stated in Algorithm. 2. Note that in step 7, function *clip* is defined through an empirical expression:

$$clip(x_{cur}) = \begin{cases} \min(x_{cur}, x_{prev}\times e^{\frac{1}{2}}); & x_{cur} > x_{prev}; \\ \max(x_{cur}, x_{prev}\times e^{-\frac{1}{2}}) & x_{cur} > x_{prev}. \end{cases} \tag{40}$$

Also, ζ is refreshing rate (an integer value) to forbid over-learning of RBFNN, and it is empirically set to 3 in practical.

C. CU LEVEL RATE CONTROL

After setting the frame-level QP (renamed as QP_{cur}), the encoder encodes each CU in order, or by using tiles in slice layer. In ROI encoding mode, CUs are classified in four categories (i.e., CU_{lr}, CU_{hr}, CU_{hn}, CU_{ln}). For those CUs belonging to ROI group, QP is lowered down to reduce distortion, and it generates more bits output. As for CUs belonging to non-ROI group, QP is lifted up to compensate bits outputs. Hence, we make small adjustments in QP for each dependent CU to meet the accuracy demands of rate control. Note that CU's distortion value can be predicted by

Algorithm 2 Frame-Level Rate Control Algorithm

1: **for** each frames f_i **in** GOP vector $GOP[cnt]$ **do**
2: Acquire N_{ick}, and compute λ_c using (37).
3: Compute a_c and μ using (38) and (39) respectively.
4: $clip(x)$, where $x = \{a_c, \lambda_c\}$.
5: Compute q_k for frame i using (32).
6: Allocate estimate target bits $\hat{R}_i$ for frame i using (7) and (8).
7: Set QP for frame i by mapping $q_k, k \in \{r, n\}$.
8: Encode each CU by implementing CU-level rate control (See Algorithm. 3).
9: Acquire the real output bits R_i.
10: **if** $mod(i, \zeta) = 0$ **then**
11: Restart training RBFNN and config its parameters.
12: **end if**
13: **end for**

ANN model by training RBFNN at the beginning of each GOP, the prediction error of CU's reference is proposed as an important factor to alternate this CU's QP. Suppose we are encoding an inter-mode CU_c now, and the co-located CUs in CU_c's reference lists are named as $\{CU_{c1}, \cdots, CU_{c\mathsf{rf}}\}$, where rf is the size of reference list. The average prediction error can be measured by:

$$\bar{e} = \frac{1}{\mathsf{rf}}\sum_{i=1}^{\mathsf{rf}}\hat{D}(i, CU_{ci}) - D(i, CU_{ci}). \tag{41}$$

$\bar{e}$ reflects the extent of overestimation or underestimation for the current encoding CU. If $\bar{e} > 0$, we deduce that distortion prediction is overestimation, so as the opposite condition that $\bar{e} < 0$. QP is altered within a range of [QP-Δ, QP+Δ] depending on $\bar{e}$. According to [9], adjusting factor Δ is set to 2 as maximum without bringing fluctuations. Therefore, we have the following control approach for CU-level: Observe that Algorithm. 3 makes adjustments of each CU based on its category c. The purpose is to let the total rate output of CUs meet the estimate target bit $\hat{R}_i$ gained by solving the

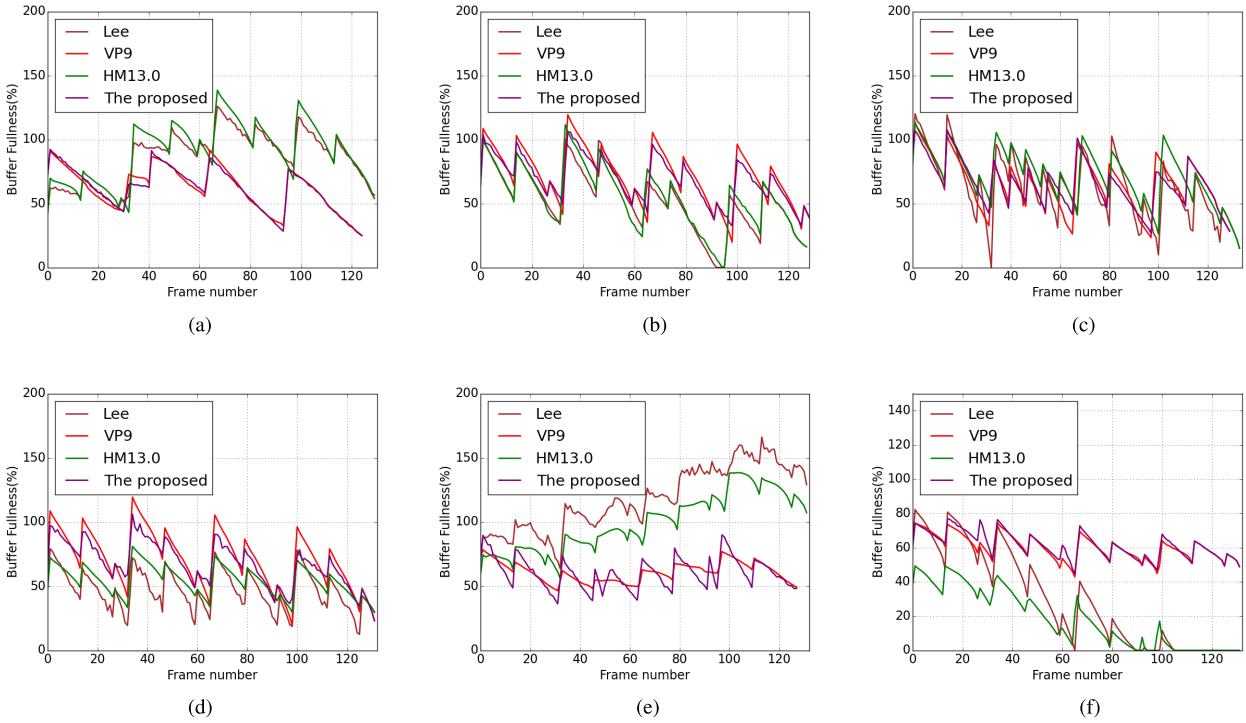


FIGURE 7. Buffer fullness analysis for different rate control schemes at low-delay. (a) *BlowingBubbles*. (b) *PartyScene*. (c) *RaceHorses*. (d) *Foreman*. (e) *Suzie*. (f) *Ice*.

R-D optimization problem in (19). A threshold ξ is set to judge wether the estimated error reaches a bound. In other words, if $|\bar{e}| > \xi$ and $\bar{e} > 0$, then increasing QP will raise $D(i, CU_{ci})$ near to $\hat{D}(i, CU_{ci})$ so that $|\bar{e}|$ is reduced. When $|\bar{e}| > \xi$ and $\bar{e} < 0$, the same result follows by decreasing QP.

Algorithm 3 Frame-Level Rate Control Algorithm

1: **for** each CU_c in frame f_i ($c \in \{l, h\}$) **do**
2: Acquire CU_c's reference list $\mathcal{L}$, compute $\bar{e}$ using (41).
3: **if** $|\bar{e}| > \xi$, adjust QP $= \bar{e} > 0?\text{QP} + \Delta : \text{QP} - \Delta$.
4: Encode the current CU_c with the assigned QP.
5: Acquire $\hat{D}(i, CU_{ci})$ for the current CU indexed by i from the encoder, and push it into $\mathcal{L}$.
6: **end for**

IV. EXPERIMENTAL RESULTS

A. EXPERIMENT SETUP

In order to verify the effectiveness of the proposed rate control scheme in ROI encoding mode, we implement the method in HM13.0 and VP9. Test sequences of various resolution with different signal characteristics are included. The GOP structure is set as IPPP, and sequences are encoded at various target bitrates. For comparison purpose, four other rate control methods are taken into consideration. They are: HM13.0, Lee *et al* [34], VP9. Their R-D models contains $R-\lambda$, $R-\rho$, and $R-q$ models with the variants. As for ROI coding mode, each frame is assigned with an ROI palette $R(i, j)$, where i, j is within the boundary of height and width of the frame. $R(i, j) = 1$ indicates the current pixel located at row i column j is ROI pixel, otherwise $R(i, j) = 0$. QP assigned with ROI blocks is initiated with a constant gap Δ to non-ROI blocks, and the coding configurations are initialized in Table. 1.

B. QUALITY COMPARISONS

Video quality is evaluated by objective PSNR since it is the most widely accepted measurement [44]. The PSNR between the reconstructed and the original video signal for the set of pixels in $\boldsymbol{A}$ is defined as :

$$\text{PSNR} = 10 \lg \frac{255^2}{\text{MSE}} \text{dB}, \tag{42}$$

where MSE denotes the mean squared error of pixel set $\boldsymbol{A}$. In this sub-experiment, we compare the proposed method, Lee's method, and the original rate control scheme of ROI encoding mode in HM13.0. The GOP size is dynamically determined both in the proposed and Lee's method by their algorithms. Specifically in *Lee et al*, ROI and non-ROI components are treated as texture and non-texture parts. Four HEVC and four AVC test sequences are included which can be downloaded on the official website [45]. The PSNR curve per frames is displayed on Fig.6, one can see that the proposed rate control scheme achieves an average of 0.5-1.5dB improvement in PSNR results. The improvements can be seen obviously when γ changes frequently or fast scene changes such as *BlowingBubbles* and *Stephan*.

TABLE 3. Bitrate accuracy and encoding time analysis.

Sequences	Proposed				HM13.0				VP9			
	BRAC	BR_T	BR_A	Enc.T	BRAC	BR_T	BR_A	Enc.T	BRA	BR_T	BR_A	Enc.T
BasketballDrive	98.33%	4.10M	4.17M	32096.3s	97.26%	4.10M	4.21M	31985.8s	98.21%	4.10M	4.02M	30050.7s
BQTerrance	98.05%	3.81M	3.88M	29960.3s	97.73%	3.81M	3.89M	28542.9s	98.21%	3.81M	3.87M	28603.3s
ParkScene	98.31%	1.62M	1.593M	16950.0s	98.14%	1.62M	1.588M	15997.1s	98.02%	1.62M	1.587M	15473.7s
RaceHorses	99.01%	1.27M	1.25M	5360.4s	98.39%	1.27M	1.29M	4450.9s	97.46%	1.27M	1.30M	4330.7s
BlowingBubbles	97.98%	474K	464.42K	1330s	97.20%	474K	460.73K	1103.9s	97.30%	474K	486.80K	1077.9s
Johnny	98.42%	350K	355.53K	1296.3s	97.92%	350K	357.28K	1089.9s	98.83%	350K	345.9K	1120.1s
City	99.51%	400K	401.59K	477.3s	99.02%	400K	403.92K	360.3s	99.72%	400K	398.88K	396.0s
Stephan	98.97%	350K	353.61K	411.6s	98.94%	350K	346.28K	303.3s	98.13%	350K	343.46K	293.7s
Ice	98.40%	300K	295.21K	301.7s	97.93%	300K	293.79K	286.6s	96.99%	300K	309.03K	255.0s
Carphone	96.45%	80K	77.16K	56.3s	97.10%	80K	82.32K	73.0s	93.84%	80K	84.93K	37.1s
Container	99.38%	80K	80.49K	44.2s	97.12%	80K	82.30K	33.6s	97.61%	80K	78.09K	38.1s

C. R-D PERFORMANCE

The R-D performances are evaluated by utilizing five data points at different target bit rates according to [37]. where BD-PSNR and bjØntegaard delta-bit rate (BD-BR) are included. In this sub-experiment, we compare the proposed rate-control method with the others on three sequences categories. The first one contains high resolutions (i.e., 1920×1080), the second one has medium resolutions (i.e., 832×480), and the last one includes traditional test sequences of H.264 whose size are CIF and QCIF. The results are summarized in Table. 2. In the first group, our method shows an average improvement of 0.30, 0.33-dB gains in BD-PSNR and 2.83%, 3.41% BD-BR reductions respectively. In the second group, the averafe improvements of BD-PSNR gains are 0.12, 0.15, 0.11-dB and the reduction of BD-BR are 3.17%, 3.30%, 2.61% correspondingly. In the third one, our approach shows similar results as the second, specifically, 0.13, 0.14-db improvement in BD-PSNR and 2.55%, 2.66%, 2.58% savings in BD-BR. In general, the proposed rate control scheme has a remarkable R-D performance according to BD-BR and BD-PSNR evaluation.

D. BUFFERFULLNESS ANALYSIS

Buffer occupancy analysis is important for rate control and bit assignments problems which is relevant to practical applications [46]. The buffer size is set as:

$$B_{uffer} = D_{elay} \times T_{arget}, \tag{43}$$

where D_{elay} denotes the delay time for the real-time video bitstream and T_{arget} is the channel bandwidth. In variable-bitrate (VBR) situation, the buffer fullness is represented by:

$$\begin{cases} B_0 = B_{init} \\ B_{i+1} = \min(B_i + RT_i - b_i, B_{max}), \end{cases} \tag{44}$$

where b_i is the bits consumed for frame i. R is the given bit rate, and T_i is the time it takes to display frame i. According to (43), the buffer occupancy is determined by the target bits for a large extent. Fig. 7 shows buffer fullness occupancy results for six test sequences. The data is presented as the percentage value, usually, B_{max} is set as $1.5 \sim 2$ times of R. For comparison purpose, we run the schemes for HM13.0, VP9 without ROI coding mode and Lee *et al*, the proposed one with ROI coding mode to plot the buffer fullness curve. One can see that in ROI coding mode, the template buffer size fluctuate since the ROI rate γ varies at each frame. However, in general, the proposed method can seldom have overflow underflow cases, which indicates that it has a better control performance. Meanwhile, it provides less variation in ROI coding mode and has a satisfactory performance in buffer occupancy.

E. BITRATE ACCURACY AND ENCODING TIME

Bitrate accuracy can reflects how precise that the actual bit rate is when comparing with the target bit rate. Then, we have the bitrate accuracy (BRAC) defined as:

$$BRAC = (1 - \frac{|BR_{Actual} - BR_{Target}|}{BR_{Target}}) \times 100\% \tag{45}$$

In this sub-experiment, we measure the bitrate accuracy together with the encoding time based on HM13.0 and VP9 encoders. An Intel 2-Core 2.7GHZ CPU is used for the encoder to encode test sequences in ROI coding mode, and the proposed method is implemented on HM13.0 compared with VP9. Table. 3 shows the results of BPAC, the target bit rate BR_T, the actual bit rate BR_A and the elapsed encoding time (sec). The target bit rate is sorted in the descend order. Obviously, the encoding time depends on the number of frames processed and the bitrate output (video quality). One can see that the proposed method gets a better bitrate accuracy in ROI coding mode at the expense of consuming longer encoding time. This is because artificial neural networks needs more training time to forecast an accurate R-D model. However, it achieves a better performance within a tolerant boundary of the extra computational time.

V. CONCLUSION

In this paper, a new rate control scheme is proposed for ROI mode coding based on DCT coefficient model. CUs are categorized by their depth levels and whether they belong to ROI or non-ROI group. The proposed R-D model takes considerations of various sraristical characteristics of transformed coefficient residues for CUs by multiple Laplacian PDFs. For the sake of improving the estimation of distortion, a machine learning approach is adopted by using historical results and parameters as the training sequence and

the distortion is predicted as the output of neuron network. A new rate control scheme is designed from GOP level to CU level. Besides, we keep a feedback control in alternating QP for each CU via accumulated errors. Experimental results show that the proposed rate control method for ROI-based coding improves an average PSNR of 0.5-1.0dB than other approaches. Also, the proposed method shows a BD-BR or BD-PSNR improvements compared with other rate control methods. In addition, it still maintains a stable buffer status levels. The future work will contain the computing time optimization of the algorithm, meanwhile, more features of frame and block characteristics will be included in R-D estimation for accuracy enhancement.

ACKNOWLEDGMENT

The authors would like to thank to the anonymous reviewers for their precious and very important comments to improve this paper. They are really thankful for their hard working and good suggestions.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] M. Wien, "High efficiency video coding: Coding tools and specification," *Artif. Intell.*, vol. 131, no. 1, pp. 309–401, 2014.

[3] 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.

[4] D. Mukherjee *et al.*, "The latest open-source video codec vp9—An overview and preliminary results," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 390–393.

[5] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," in *Proc. Picture Coding Symp.*, Dec. 2013, pp. 394–397.

[6] D. Mukherjee *et al.*, "A technical overview of vp9 2014;the latest open-source video codec," *Motion Imag. J. SMPTE*, vol. 124, no. 1, pp. 44–54, Jan. 2015.

[7] O. Faugeras, "Digital color image processing within the framework of a human visual model," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 4, pp. 380–393, Aug. 1979.

[8] H. Song and C. C. J. Kuo, "A region-based H.263+ codec and its rate control for low VBR video," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 489–500, Jun. 2004.

[9] L. Tong and K. R. Rao, "Region of interest based H.263 compatible codec and its rate control for low bit rate video conferencing," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2005, pp. 249–252.

[10] R. Schafer, "MPEG-4: A multimedia compression standard for interactive applications and services," *Electron. Commun. Eng. J.*, vol. 10, no. 6, pp. 253–262, Dec. 1998.

[11] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, vol. 2. Jun. 1999, p. 252.

[12] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[13] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. Aug. 2004, pp. 28–31.

[14] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *Proc. SPIE Independent Compon. Anal., Compressive Sampling, Wavelets, Neural Net, Biosyst. Nanoeng.*, vol. 8750, p. 87500V, May 2013. [Online]. Available: http://dx.doi.org/10.1117/12.2017684, doi: 10.1117/12.2017684.

[15] E. Candés, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?: Recovering low-rank matrices from sparse errors," in *Proc. Sensor Array Multichannel Signal Process. Workshop (SAM)*, Oct. 2010, pp. 201–204.

[16] Z. Gao, L.-F. Cheong, and Y.-X. Wang, "Block-sparse RPCA for salient motion detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1975–1987, Oct. 2014.

[17] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Oct. 2012, pp. 1225–1228.

[18] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3047–3064, May 2012.

[19] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[20] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[21] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[22] P. Chiranjeevi and S. Sengupta, "Detection of moving objects using multichannel kernel fuzzy correlogram based background subtraction," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 870–881, Jun. 2014.

[23] D. K. Panda and S. Meher, "Detection of moving objects using fuzzy color difference histogram based background subtraction," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 45–49, Jan. 2016.

[24] Y. Shi, S. Yue, B. Yin, and Y. Huo, "A novel ROI-based rate control scheme for H.264," in *Proc. 9th Int. Conf. Young Comput. Sci. (ICYCS)*, Nov. 2008, pp. 77–81.

[25] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "Roi-based rate control using tiles for an hevc encoded video stream over a lossy network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 1389–1393.

[26] H. Li, Z. Wang, H. Cui, and K. Tang, "An improved ROI-based rate control algorithm for H. 264/AVC," in *Proc. 8th Int. Conf. Signal Process.*, vol. 2. Sep. 2006, pp. 252–257.

[27] 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.

[28] W. Lim, I. V. Bajić, and D. Sim, "QP initialization and adaptive MAD prediction for rate control in HEVC-based multi-view video coding," in *Proc. IEEE 11th. IVMSP Workshop*, Jun. 2013, pp. 1–4.

[29] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, Jun. 2002.

[30] L. Yang, L. Zhang, S. Ma, and D. Zhao, "A ROI quality adjustable rate control scheme for low bitrate video coding," in *Proc. Picture Coding Symp.*, May 2009, pp. 1–4.

[31] H. Choi, J. Yoo, J. Nam, D. Sim, and I. V. Bajić, "Pixel-wise unified rate-quantization model for multi-level rate control," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1112–1123, Dec. 2013.

[32] S. Wang, S. Ma, S. Wang, D. Zhao, and W. Gao, "Rate-GOP based rate control for High Efficiency Video Coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1101–1111, Dec. 2013.

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

[34] B. Lee, M. Kim, and T. Q. Nguyen, "A frame-level rate control scheme based on texture and nontexture rate models for High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 465–479, Mar. 2014.

[35] L. Tian, Y. Zhou, and X. Cao, "A new rate-complexity-QP algorithm (RCQA) for HEVC intra-picture rate control," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2014, pp. 375–380.

[36] B. Li, H. Li, L. Li, and J. Zhang, "λ domain rate control algorithm for High Efficiency Video Coding," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3841–3854, Sep. 2014.

[37] M. Wang, K. N. Ngan, and H. Li, "Low-delay rate control for consistent quality using distortion-based lagrange multiplier," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 2943–2955, Jul. 2016.

[38] W. Gao, S. Kwong, H. Yuan, and X. Wang, "DCT coefficient distribution modeling and quality dependency analysis based frame-level bit allocation for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 139–153, Jan. 2016.
[39] N. Kamaci, Y. Altunbasak, 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.
[40] S. Sanz-Rodríguez, Ó. del-Ama-Esteban, M. de-Frutos-López, and F. Díaz-de-María, "Cauchy-density-based basic unit layer rate controller for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 8, pp. 1139–1143, Aug. 2010.
[41] 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.
[42] E. Y. Lam, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.
[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2006, pp. 683–701. [Online]. Available: http://dx.doi.org/10.1007/s10208-009-9045-5
[44] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
[45] *The HEVC/AVC Official Test Sequences*. [Online]. Available: ftp://hevc@ftp.tnt.uni-hannover.de/testsequences
[46] P. Westerink, R. Rajagopalan, and C. Gonzales, "Two-pass MPEG-2 variable-bit-rate encoding," *IBM J. Res. Develop.*, vol. 43, no. 4, pp. 471–488, Jul. 1999.

ZHEWEI ZHANG received the B.S. degree from Beijing Jiaotong University, Beijing, China, where he is currently pursuing the Ph.D. degree with the School of Electronic Information Engineering. His current research interests include video compression, image process, machine learning, and pattern recognition and analysis.

TAO JING received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 1994 and 1999, respectively. He is currently a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, China. His research interests include capacity analysis, spectrum prediction and resource management in cognitive radio networks, RFID in intelligent transporting system, smart phone application, and multimedia.

JINGNING HAN received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2008 and 2012, respectively. He is currently with the WebM Codec Team, Google Inc., Mountain View, CA, USA, where he is involved in video compression, processing, and related technologies. His research interests include video coding and computer architecture. He was a recipient of the Outstanding Teaching Assistant Awards from the Department of Electrical and Computer Engineering, University of California at Santa Barbara, in 2010 and 2011, the Dissertation Fellowship in 2012, and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2012.

YAOWU XU received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, NewYork, NY, USA, in 2008 and 2012, respectively. He is currently with the WebM Codec Team, Google Inc., Mountain View, CA, USA, where he is involved in video compression, processing, and related technologies. He is an Expert in video compression and processing, digital image processing, embedded audio/video architecture, mobile audio/video architecture, hardware-software integration, real-time multimedia embedded systems.

FAN ZHANG received the B.E. and M.S. degrees in communication and information system from Beijing Jiaotong University, Beijing, China, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Shu Hua Wireless Network and Information Perception Center,. His research interests are cognitive radio networks, energy harvesting, and mobile social networks.