An Analytic Framework for Frame-Level Dependent Bit Allocation in Hybrid Video Coding

Chao Pang, *Member, IEEE,* Oscar C. Au, *Fellow, IEEE,* Feng Zou, *Student Member, IEEE,* Jingjing Dai, *Student Member, IEEE,* Xingyu Zhang, *Student Member, IEEE,* and Wei Dai, *Student Member, IEEE*

***Abstract*—In this paper, we address the frame-level dependent bit allocation (DBA) problem in hybrid video coding. In most existing methods, the DBA solution is achieved at the expense of high, sometimes even unbearable, computational complexity because of the multipass coding involved. Motivated by this, we propose a model-based approach as an attempt to solve this problem analytically. Leveraging the predictive nature in hybrid video coding, we develop a novel interframe dependency model (IFDM), which enables a quantitative measure of the coding dependency between the current frame and its reference frame. Based on the IFDM, the buffer-constrained frame-level DBA problem is carefully formulated. Finally, the model-based DBA method called IFDM-DBA is derived, in which successive convex approximation techniques are employed to convert the original optimization problem into a series of convex optimization problems of which the optimal solutions can be obtained efficiently. Experimental results suggest that the proposed IFDM-DBA method can achieve up to a 23% bitrate reduction over the JM reference software of H.264.**

***Index Terms*—Convex optimization, dependent bit allocation (DBA), hybrid video coding, interframe dependency model (IFDM).**

I. Introduction

THE LAST few decades have witnessed an explosion in ideas and theories on multimedia technologies. Due to the huge size of raw video data, digital video compression is one of the key techniques that enables efficient interchange and distribution of visual information. Presently, the most successful video compression algorithms are based on hybrid video coding structure, which consists of a combination of the in-loop temporal motion estimation/compensation with decorrelating transform in pixel domain. Most of the existing video coding standards, such as MPEG-1/2/4 and H.261/263/264, conform to this structure.

Manuscript received June 4, 2012; revised October 8, 2012; accepted January 3, 2013. Date of publication February 1, 2013; date of current version May 31, 2013. This work was supported in part by the Hong Kong University of Science and Technology through Project FSGRF12EG01. This paper was presented in part at the IEEE International Workshop on Multimedia Signal Processing, Hangzhou, China, October 2012. This paper was recommended by Associate Editor P. Salama.

C. Pang, O. C. Au, J. Dai, X. Zhang, and W. Dai are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: pangchao87@gmail.com; eeau@ust.hk; jingjdai@gmail.com; eexyzhang@ust.hk; weidai@ust.hk).

F. Zou is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, and also with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: fengzou@ust.hk).

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.2013.2244795

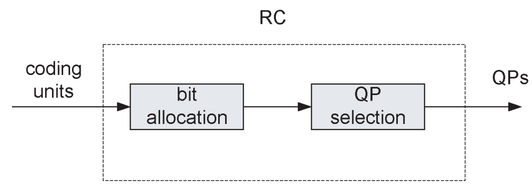

Fig. 1. Structure of RC.

In many video coding applications, because of the storage capacity and transmission bandwidth constraints, rate control (RC) is often indispensable in order to regulate the output bitstream at a given target bitrate and lead to better visual quality. As shown in Fig. 1, the RC module typically involves two steps: bit allocation and quantization parameter (QP) selection. The bit allocation aims to allocate the total available bits to the coding units [e.g., macroblock (MB), slice, frame, etc.] such that the resulted total distortion is minimized. In the step of QP selection, the QP needs to be determined in order to encode the coding unit at approximately the target number of bits assigned in the bit allocation step. Thus, many sophisticated rate–quantization (R–Q) models such as the quadratic model [1]–[3], ρ-domain model [4], [5], and statistical model [6], [7] are proposed in the literature to obtain accurate bitrate adaptation. In this paper, we focus on the bit allocation step and the coding unit for bit allocation is chosen to be one frame.

Generally, the optimal frame-level bit allocation strategy can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{R_i} \quad & \bar{D}(R_1, R_2, \ldots, R_N) \qquad i = 1, \ldots, N \\ \text{s.t.} \quad & \sum_{i=1}^{N} R_i \leq R \end{aligned} \tag{1}$$

where R is the total available bits for the N frames. R_i is the number of bits allocated to the ith frame and D_i is the corresponding compression distortion, which is usually measured by the mean squared error (MSE) between the original signal and the corresponding reconstructed signal. $\bar{D}$ is the average distortion of the N frames. In the literature, the frame-level bit allocation methods can be classified into two categories: independent bit allocation (IBA) methods and dependent bit allocation (DBA) methods. In IBA methods, the influence of the current frame on the future frames is neglected, and the rate–distortion (R–D) functions of the frames are assumed to be independent. Consequently, the $\bar{D}$ in (1) can be separately represented, and the optimization problem in (1) is simplified into

$$\begin{aligned} \min_{R_i} \quad & \sum_{i=1}^{N} D_i(R_i) \qquad i = 1, \ldots, N \\ \text{s.t.} \quad & \sum_{i=1}^{N} R_i \le R. \end{aligned} \tag{2}$$

With this simplification, the optimal solution of (2) is derived using conventional optimization methods such as Lagrangian optimization. Actually, the bit allocation methods in many practical RC algorithms [8]–[10], both one-pass and two-pass, are IBA methods. Apparently, since the IBA methods have relaxed the problem in (1) by neglecting the coding dependency between neighboring frames, they can only lead to suboptimal bit allocation solutions. Because of the relaxation, as pointed out in [11] and [12], the coding performance gap between the IBA methods and the DBA methods sometimes can be quite large.

Different from the IBA methods, the DBA methods take the interframe coding dependency into consideration. Intuitively, assuming that all the coding units in each frame are encoded with the same QP, a search tree can be established and the problem in (1) can be optimally solved through searching all the possible QP combinations of the frames. However, the computational complexity of the brute-force search method increases exponentially with the total number of frames. Based on the observation that the R–D functions for the predicted frame are usually monotonic in the quality of the reconstructed reference frame, the complexity is greatly reduced by pruning the search tree as in [11] and [13]. As stated in [13], the computational complexity is dominated by generating the necessary R–D operation points. Inspired by this, a fast approach is presented in [14] in which fewer R–D operation points are for the R–D curve reconstruction. In [12], a steepest descent algorithm is proposed to approximate the optimal DBA solution. Although these fast implementations have greatly reduced the computational complexity compared with the brute-force search method, the computational burden is still not bearable in many practical applications because of the involved multipass coding. Thus, to avoid the multipass coding, a model-based DBA method is proposed in [15]. In [15], the interframe dependency is quantitatively measured by the percentage of skipped MBs in one frame, and the optimal DBA strategy is obtained analytically. However, the method in [15] can only handle static sequences, and the skipped MB percentage actually cannot be accurately estimated before the encoding. Besides, the coding dependency between nonskipped MBs and their reference MBs also exists, which cannot be detected using this interframe dependency measure.

To overcome the drawbacks in existing DBA methods, we propose in this paper a model-based frame-level DBA method called IFDM-DBA to efficiently allocate the available bits to the frames based on a novel coding dependency model. Specifically, starting from the predictive nature in hybrid video coding, a novel dependency model is developed that enables quantitatively measuring the coding dependency for both the skipped MBs and nonskipped MBs. Then, after introducing the framewise R–D functions for intracoded and intercoded frames, the buffer-constrained DBA problem is carefully formulated, and successive convex approximation techniques are employed to convert the original optimization problem into a series of convex optimization problems of which the optimal solutions can be obtained efficiently. Note that part of this section has been reported in our previous work [16]. The differences between this paper and [16] lie in the following aspects: first, in this paper, the proposed models, including the IFDM and framewise R–D models, are analyzed and explained in detail. Also, the accuracy of the proposed models is studied carefully, and it is compared with other existing models; second, the buffer constraint is taken into account in the bit allocation process and it is incorporated in the problem formulation; third, the problem formulation and solving process is described in detail, and the needed proof and explanations are given in this paper; last but not the least, more video sequences are tested and the performance of the proposed method is compared with more other methods. All of these make the experimental results much more convincing. Moreover, the computational complexity of the proposed method is carefully studied and analyzed in this paper.

The rest of this paper is organized as follows. In Section II, a novel interframe dependency model (IFDM) which can quantitatively measure the coding dependency is introduced. In Section III, after introducing the framewise R–D models and buffer constraints, we derive the proposed optimal DBA scheme called IFDM-DBA based on the IFDM. The experimental results and further discussion are given in Section IV. Section V concludes this paper.

II. IFDM

In a generic hybrid video encoder, such as MPEG-1/2/4 and H.261/263/264 encoder, differential pulse code modulation (DPCM) in the form of motion-compensated coding is widely utilized. At the encoder side, the input frame is divided into nonoverlapped blocks and encoded block by block. For each block, motion estimation (ME) is employed to exploit the temporal redundancy between the current frame and its reference frame. Note that the reference frame is usually selected from the reconstructed frames of previous frames in order to avoid the mismatch between the encoder and decoder. During ME, the best-matched block, in terms of minimum sum of absolute differences (SAD) or sum of absolute transformed differences, is chosen to be the prediction block. Then, the residue block is calculated by subtracting the prediction block from the original block. Finally, the residue block is transformed using discrete cosine transform (DCT), and the transform coefficients are quantized and entropy coded.

As shown in Fig. 2, let x_n be the input signal of the nth frame and $\tilde{x}_n$ be the prediction signal of x_n. Then, the residue

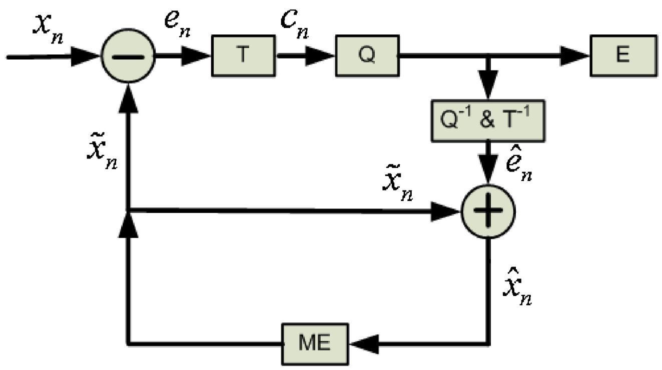

Fig. 2. Basic structure of a hybrid video encoder.

signal e_n is calculated by

$$e_n = x_n - \tilde{x}_n. \tag{3}$$

Without loss of generality, the reference frame of the nth frame is assumed to be the reconstructed frame of the immediate preceding frame. Thus, we have

$$\tilde{x}_n = \hat{x}_{n-1} \tag{4}$$

where $\hat{x}_{n-1}$ is the reconstructed signal of the $(n-1)$th frame.

Combining (3) with (4), we get

$$\begin{aligned} e_n &= x_n - \hat{x}_{n-1} \\ &= \underbrace{(x_n - x_{n-1})}_{z_n} + \underbrace{(x_{n-1} - \hat{x}_{n-1})}_{q_{n-1}} \end{aligned} \tag{5}$$

where z_n is the prediction error with the original signal of the $(n-1)$th frame for prediction. q_{n-1} is the quantization error of the $(n-1)$th frame.

Usually, the expected values of e_n, c_n, z_n, and q_{n-1} are assumed to be zero. Then, the variance of e_n denoted by $\sigma^2_{e_n}$ is

$$\begin{aligned} \sigma^2_{e_n} &= \mathbb{E}({e_n}^2) \\ &= \mathbb{E}((q_{n-1} + z_n)^2) \\ &= \mathbb{E}({q_{n-1}}^2) + \mathbb{E}({z_n}^2) \\ &= D_{n-1} + \sigma^2_{z_n} \end{aligned} \tag{6}$$

where D_{n-1} is the compression distortion, which is measured by MSE, of the $(n-1)$th frame. $\sigma^2_{z_n}$ is the variance of z_n. Note that the third equation holds by assuming that z_n and q_{n-1} are uncorrelated. Furthermore, as DCT is a unitary transform, the variance of the DCT coefficients denoted by $\sigma^2_{c_n}$ is

$$\sigma^2_{c_n} = \sigma^2_{e_n} = D_{n-1} + \sigma^2_{z_n}. \tag{7}$$

For ease of notation, we denote σ^2_n as either $\sigma^2_{c_n}$ or $\sigma^2_{e_n}$ in the rest of this paper.

However, when (7) is used within a specific video coding standard, it needs to be further modified to be more accurate. This is due to the involved dedicated compression techniques that make it difficult to estimate D_{n-1} and $\sigma^2_{z_n}$ exactly before the encoding. For example, in H.264/AVC, rate–distortion optimization (RDO) techniques are often employed at the encoder to achieve superior performance. Consequently, this would potentially influence the statistics of DCT coefficients. To elaborate, RDO is used to select the optimal encoding parameters (i.e., number of block partitions, intraprediction modes, motion vectors, etc.) for each MB in an R–D optimized sense. During RDO, the predefined R–D cost corresponding to each possible encoding parameter is calculated, and the encoding parameter leading to the minimal R–D cost is selected as the final encoding parameter. Typically, RDO contains R–D optimized mode decision and R–D optimized ME. The corresponding R–D costs, $\mathrm{RDCost}_{\mathrm{mode}}$ and $\mathrm{RDCost}_{\mathrm{ME}}$, for mode decision and ME, are defined as

$$\begin{aligned} \mathrm{RDCost}_{\mathrm{mode}} &= \mathrm{SSD} + \lambda_{\mathrm{mode}} \cdot \mathrm{Rate} \\ \mathrm{RDCost}_{\mathrm{ME}} &= \mathrm{SAD} + \lambda_{\mathrm{ME}} \cdot \mathrm{Rate} \end{aligned} \tag{8}$$

where λ_{mode} and λ_{ME} are the Lagrange multiplier which can be obtained by

$$\lambda_{\mathrm{mode}} = c \cdot Q^2, \quad c = \begin{cases} 0.68, \text{if no. of B frames} > 0 \\ 0.85, \text{otherwise} \end{cases} \tag{9}$$

$$\lambda_{\mathrm{ME}} = \sqrt{\lambda_{\mathrm{mode}}} \tag{10}$$

where Q is the quantization stepsize. Equations (8)–(10) imply that, when a larger Q is employed which implies a larger Lagrange multiplier value, the encoder favors the mode that generates less bits and pays less attention to the distortion this mode might produce. In this case, the variance of the residue signals σ^2_n tends to be larger. However, if the current coding unit is quantized with a smaller Q, the mode with less distortion will be chosen and σ^2_n will be smaller.

Because of the influence of the RDO on the statistics of DCT coefficients, one more item Q is added in (7). Moreover, α, β, and γ are introduced to make (7) more accurate. Finally, (7) is changed into

$$\sigma^2_n = \alpha \cdot D_{n-1} + \beta \cdot \tilde{\sigma}^2_n + \gamma \cdot Q_n \tag{11}$$

where σ^2_n is the variance of DCT coefficients of the nth frame. $\tilde{\sigma}^2_n$ is an estimate of $\sigma^2_{z_n}$ before the actual encoding. Q_n is the quantization stepsize (Q) for the nth frame. α, β, and γ are positive parameters. As (11) describes the influence of the reference frame on the R–D characteristics of the current frame, we denote it as the IFDM.

To apply the proposed IFDM, we need to estimate $\tilde{\sigma}^2_n$ in (11). By performing ME to the original video sequence, we estimate $\tilde{\sigma}^2_n$ as the variance of the residue. Note that the ME results usually are different with the ME results obtained during the actual encoding of the current frame. Thus, $\tilde{\sigma}^2_n$ can be viewed as an estimate of $\sigma^2_{z_n}$.

To demonstrate the accuracy of the proposed IFDM, we plot the fitting performances of several video sequences as shown in Fig. 3. For each video sequence, two randomly chosen neighboring frames are encoded. Let QP_1 and QP_2 be the QPs used to encode these two frames. In the experiments, QP_1 and QP_2 are selected to be every two QP values ranging from 10 to 46. Therefore, there are in total 361 possibilities of the QP pair (QP_1, QP_2). During each encoding process, the following coding results are recorded: the DCT coefficient variance of the second frame σ^2_2, the Q used for the second

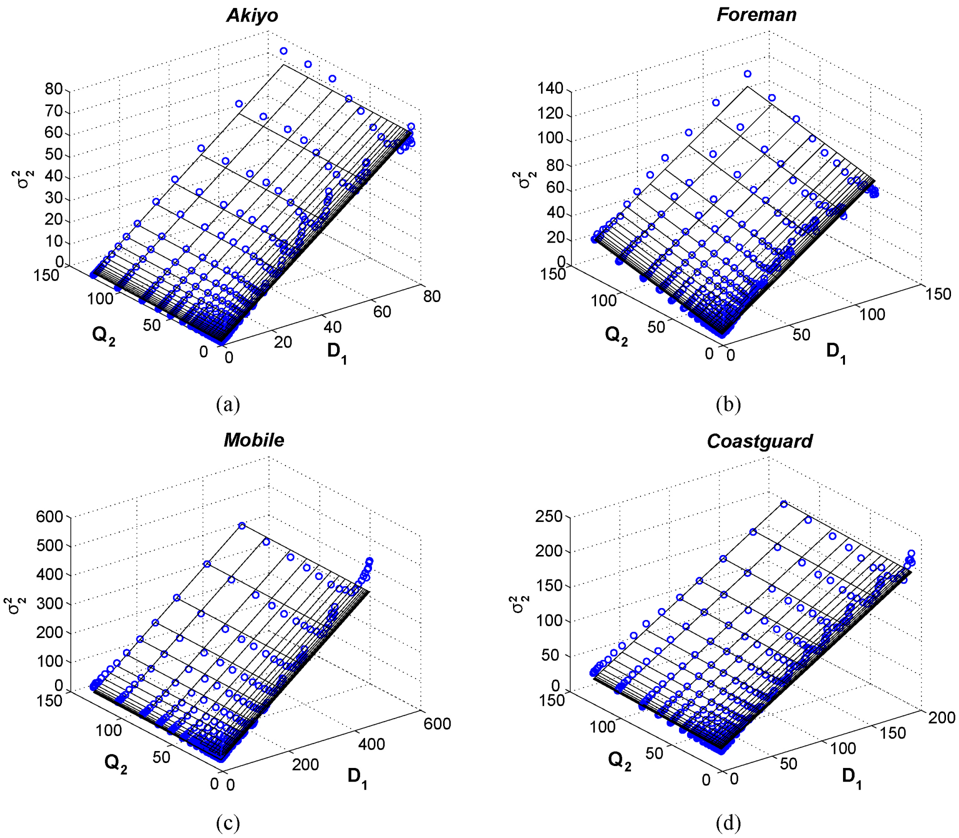

Fig. 3. Fitting results of σ_2^2 and (D_1, Q_2) using the proposed IFDM for two randomly chosen neighboring frames of different video sequences. (a) *Akiyo*. (b) *Foreman*. (c) *Mobile*. (d) *Coastguard*.

frame Q_2, and the compression distortion of the first frame D_1. As shown in Fig. 3, the proposed IFDM fits the actual data quite well.

In addition, Table I shows the estimation accuracy in terms of the $\mathbb{R}^2$ values of the proposed IFDM for some typical video sequences. $\mathbb{R}^2$ is a metric used to quantitatively measure the degree of data variation from a given model [17]. It is defined as

$$\mathbb{R}^2 = 1 - \frac{\sum_i (X_i - \hat{X}_i)^2}{\sum_i (X_i - \bar{X}_i)^2} \tag{12}$$

where X_i and $\hat{X}_i$ are the real and the estimated values of one data point i. $\bar{X}$ is the mean of all the data points. The closer the value of $\mathbb{R}^2$ is to 1, the more accurate the model is. Moreover, we compare the model accuracy of the proposed IFDM with the model introduced in [18]. From Table I, we can learn that the $\mathbb{R}^2$ values of the proposed IFDM are always higher than the $\mathbb{R}^2$ values of the model in [18], which implies that the proposed IFDM has a better estimation accuracy in estimating σ_n^2.

TABLE I

COMPARISON OF R^2 VALUES OF [18] AND THE PROPOSED IFDM

Sequence	$\mathbb{R}^2$ using the method in [18]	$\mathbb{R}^2$ using the proposed IFDM
Akiyo	0.946	0.976
Coastguard	0.939	0.974
Foreman	0.796	0.973
Mobile	0.931	0.975
City	0.932	0.966
Shuttlestart	0.839	0.972
Cactus	0.846	0.982
Traffic	0.946	0.978

III. PROPOSED IFDM-BASED FRAME-LEVEL DEPENDENT BIT ALLOCATION (IFDM-DBA) METHOD

In this section, the proposed IFDM-based frame-level DBA method called IFDM-DBA will be introduced. Before unfolding the proposed IFDM-DBA algorithm, the framewise R–D functions and buffer constraints are discussed first.

A. Framewise R–D Functions

For intracoded frames, in order to accommodate various contents of video sequences, our previously proposed frame complexity-guided R–D model [19] is employed

$$R(D) = G \cdot (\frac{a_0}{D + b_0} + c_0) \tag{13}$$

where G is the average gradient of a frame. a_0, b_0, and c_0 are model parameters. The fitting results of the R–D function in (13) for two typical video sequences are shown in Fig. 4.

For intercoded frames, since the DCT coefficients are assumed to be of zero-meaned Laplacian distribution, according

TABLE II

R^2 VALUES OF THE R–D FUNCTIONS FOR INTRAFRAMES

Sequence	$\mathbb{R}^2$ using (13)	Sequence	$\mathbb{R}^2$ using (13)
Carphone	0.995	*Silent*	0.997
News	0.990	*Foreman*	0.999
Crew	0.993	*Shuttlestart*	0.995
Bigships	0.993	*City*	0.995

TABLE III

R^2 VALUES OF DIFFERENT R–D FUNCTIONS FOR INTERFRAMES

Sequence	$\mathbb{R}^2$ using (14)	$\mathbb{R}^2$ using (15)
Akiyo	0.923	0.950
Foreman	0.943	0.979
Mobile	0.959	0.988
Paris	0.913	0.945
Crew	0.910	0.963
Shuttlestart	0.964	0.987
Bigships	0.947	0.986
City	0.978	0.993

to the classic R–D theory [20], we have

$$R(D) = a_1 \cdot \log\frac{\sigma^2}{\mathrm{D}} \qquad \text{where } \sigma^2 > D \tag{14}$$

where a_1 is model parameter and σ^2 is the variance of the DCT coefficients. In our experiments, we find that this R–D function fails to model the header bits (e.g., MB level and slice level) which need to be sent even when all the DCT coefficients are quantized to zero. Therefore, we modify the model slightly by adding an offset b_1 to compensate this effect so that

$$R(D) = a_1 \cdot \log\frac{\sigma^2}{\mathrm{D}} + b_1 \qquad \text{where } \sigma^2 > D \tag{15}$$

The fitting results of the R–D function in (15) for two typical video sequences are shown in Fig. 5.

To make a quantitative measure, the $\mathbb{R}^2$ values of the proposed R–D models for some typical video sequences are summarized in Tables II and III. As shown in the tables, the $\mathbb{R}^2$ values are very close to 1, which implies a superior fitting performance of the proposed R–D models for both intracoded and intercoded frames.

B. Buffer Constraints

In video coding, a decoder buffer is often utilized to receive the bitstream from transmission channel, and the decoder drains the compressed data, decodes it, and displays the picture to end users [18]. Let R_i be the allocated bits to the ith frame and T_0 be the initial decoding delay (in frames) of the decoder. The decoder buffer occupancy, denoted by B_n, can be calculated from the difference between the input and output bits of the buffer, that is

$$B_n = \begin{cases} n \cdot \bar{R} - \sum_{i=1}^{n-T_0} R_i, & \text{if } n \geq T_0 \\ n \cdot \bar{R}, & \text{otherwise} \end{cases} \tag{16}$$

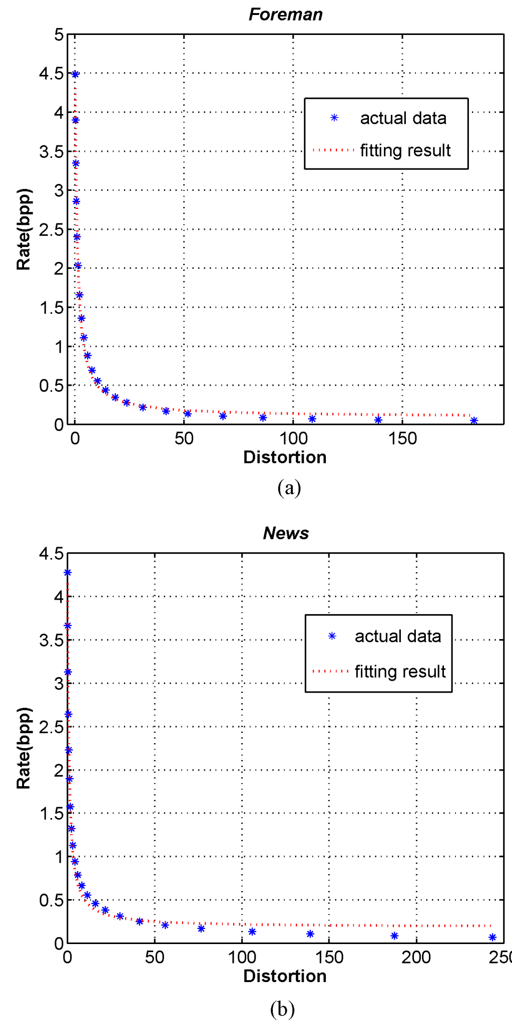

Fig. 4. Fitting results of the proposed R–D function for intracoded frames. (a) *Foreman*. (b) *News*.

where $\bar{R}$ is the average bits allocated to each frame and it can be calculated as

$$\bar{R} = \frac{B_R}{F_R} \tag{17}$$

where B_R is the target bitrate and F_R is the target framerate.

In bit allocation, one important requirement is to avoid buffer underflow or buffer overflow happening at the decoder. In other words, the buffer occupancy should be less than the buffer capacity, that is

$$0 \leq B_n \leq B \tag{18}$$

where B is the buffer capacity. The constraints in (18) are the buffer constraints to which we need to conform during the bit allocation.

C. Proposed Frame-Level DBA Method (IFDM-DBA)

Leveraging the aforementioned IFDM, framewise R–D models, and buffer constraints, in this section, we will introduce the proposed IFDM-DBA method in detail.

Assuming there are N frames in each group of pictures (GOP), with the first frame encoded as intracoded frame and

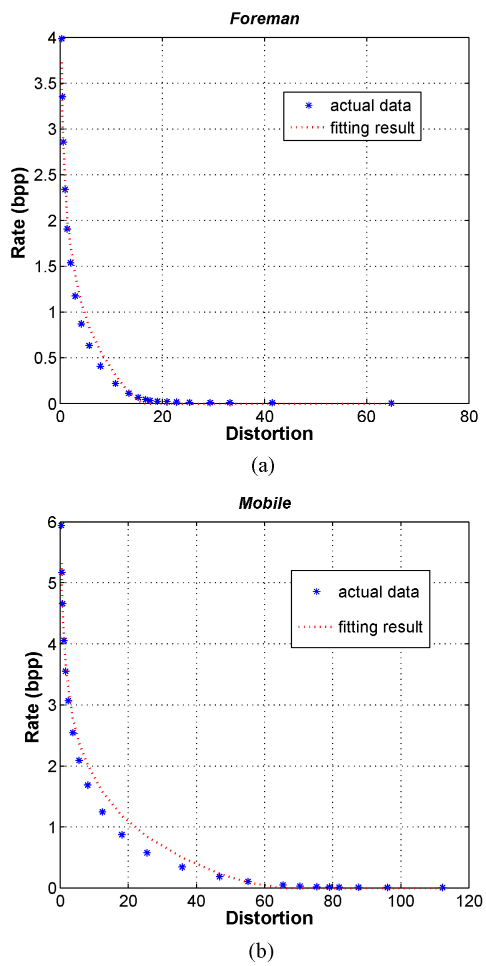

Fig. 5. Fitting results of the proposed R–D function for intercoded frames. (a) *Foreman*. (b) *Mobile*.

all the following $N-1$ frames encoded as intercoded frames. Let $\mathbf{R} = [R_1, R_2, \ldots, R_N]$ be the bit allocation strategy to the N frames and $\mathbf{D} = [D_1, D_2, \ldots, D_N]$ be the corresponding compression distortion. In this section, we focus on the following problem: under a predefined total bit budget, how to determine the frame-level bit allocation strategy $\mathbf{R}$ such that the overall compression distortion of the N frames is minimized. Mathematically, the buffer-constrained frame-level DBA problem can be formulated as

$$\begin{aligned}
\min_{\mathbf{R}} \quad & \sum_{i=1}^{N} D_i \\
\text{s.t.} \quad & \sum_{i=1}^{N} R_i \le R_{\text{GOP}} \\
& R_1 = G \cdot \left(\frac{a_0}{D_1 + b_0} + c_0\right) \\
& R_j = a_1 \cdot log\frac{\sigma_j^2}{D_j} + b_1 \qquad j = 2, 3, \ldots, N \\
& \sigma_j^2 = \alpha \cdot D_{j-1} + \beta \cdot \tilde{\sigma}_j^2 + \gamma \cdot Q_j \\
& 0 \le B_i \le B
\end{aligned} \tag{19}$$

where R_{GOP} is the total bit budget for the N frames in current GOP and it can be calculated as

$$R_{\text{GOP}} = N \cdot \bar{R} + R_{\text{rem}} \tag{20}$$

where R_{rem} is the remaining bits from the previous GOP.

As shown in the Appendix, (19) can be rewritten as

$$\begin{aligned}
\min_{\mathbf{D}} \quad & \sum_{i=1}^{N} D_i \\
\text{s.t} \quad & s + t + a_1 \cdot \sum_{j=2}^{N-1} \log\left(\alpha + \frac{\beta \cdot \tilde{\sigma}_{\text{j+1}}^2 + \gamma \cdot \text{Q}_{\text{j+1}}}{\text{D}_\text{j}}\right) \\
& + a_1 \cdot \log\frac{1}{\text{D}_\text{N}} + (N-1) \cdot b_1 \le R_{\text{GOP}} \\
& G \cdot \left(\frac{a_0}{D_1 + b_0} + c_0\right) \le s \\
& a_1 \cdot \log(\alpha \cdot \text{D}_1 + \beta \cdot \tilde{\sigma}_2^2 + \gamma \cdot \text{Q}_2) \le t \\
& 0 \le B_i \le B
\end{aligned} \tag{21}$$

where s and t are two slack variables.

Note that in order to solve the optimization problem in (21), $\tilde{\sigma}_j^2$ and Q_j $(j = 2, 3, \ldots, N)$ need to be estimated first. As discussed in Section II, we perform ME to the corresponding original frames of the test sequence, and $\tilde{\sigma}_j^2$ can be approximated by the variance of the residue. For Q_j, it is estimated from the average Q used in the previous GOP.

With both $\tilde{\sigma}_j^2$ and Q_j estimated, we simplify the notation by defining

$$\text{Diff}_\text{j} = \beta \cdot \tilde{\sigma}_\text{j}^2 + \gamma \cdot \text{Q}_\text{j} \tag{22}$$

which is now known, and Diff_j is positive. Thus, (21) becomes

$$\begin{aligned}
\min_{\mathbf{D}} \quad & \sum_{i=1}^{N} D_i \\
\text{s.t} \quad & s + t + a_1 \cdot \sum_{j=2}^{N-1} \log\left(\alpha + \frac{\text{Diff}_{\text{j+1}}}{\text{D}_\text{j}}\right) \\
& + a_1 \cdot \log\frac{1}{\text{D}_\text{N}} + (N-1) \cdot b_1 \le R_{\text{GOP}} \\
& G \cdot \left(\frac{a_0}{D_1 + b_0} + c_0\right) \le s \\
& a_1 \cdot \underbrace{\log(\alpha \cdot \text{D}_1 + \text{Diff}_2)}_{g(D_1)} \le t \\
& 0 \le B_i \le B.
\end{aligned} \tag{23}$$

However, since $g(D_1)$ is not a convex function of D_1, (23) is not a convex optimization problem. Thus, it it difficult to find the optimal solution of (23) directly. In this paper, successive convex approximation techniques are employed to solve the optimization problem in (23). Here, we briefly describe the successive convex approximation techniques, and interested readers are encouraged to read [21] and [22] for further details.

Consider the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\
s.t. \quad & f_i(\mathbf{x}) \le 0, \qquad 1 \le i \le m \\
& h_i(\mathbf{x}) = 0, \qquad 1 \le i \le p
\end{aligned} \tag{24}$$

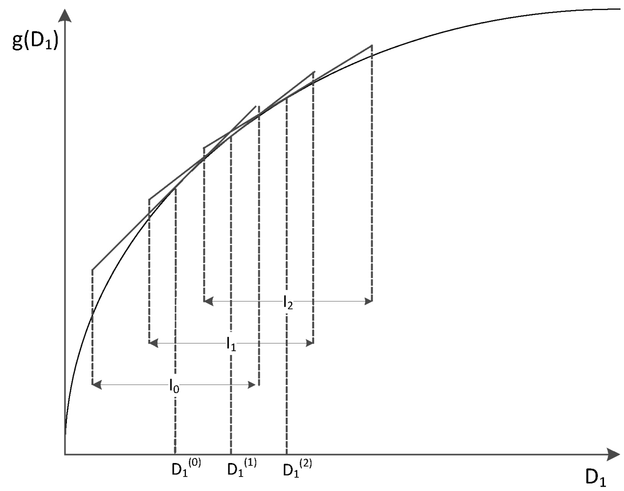

Fig. 6. Geometric understanding of *successive convex approximation*.

where $\mathbf{x}$ is the optimization variable. $f_0, f_1, ..., f_m$ are convex functions with the exception of $f_t(1 \leq t \leq m)$ which is not convex, and $h_1, h_2, ..., h_p$ are affine functions. Instead of directly solving (24) which is often very difficult, we will solve (24) iteratively by approximating $f_t(\mathbf{x})$ with $\tilde{f}_t(\mathbf{x})$ which is convex. During each iteration, (24) is changed into a convex optimization problem of which the optimal solution can be obtained efficiently using interior-point methods [21]. As pointed out in [22], this iterative approximation will converge to a point satisfying the Karush–Kuhn–Tucker (KKT) conditions of the original problem if the approximation of $f_t(\mathbf{x})$ meets the following three requirements:

1) $f_t(\mathbf{x}) \leq \tilde{f}_t(\mathbf{x})$ for all $\mathbf{x}$;
2) $f_t(\mathbf{x}_0) \leq \tilde{f}_t(\mathbf{x}_0)$ where $\mathbf{x}_0$ is the optimal solution of the approximated problem in the previous iteration;
3) $\nabla f_t(\mathbf{x}_0) = \nabla \tilde{f}_t(\mathbf{x}_0)$.

In our proposed IFDM-DBA algorithm, during the ith iteration, $g(D_1)$ is approximated with the affine function $\tilde{g}(D_1)$ defining as

$$\tilde{g}(D_1) = \underbrace{a_1 \cdot \log(\alpha \cdot \mathrm{D}_1^{\mathrm{i}-1} + \mathrm{Diff}_2)}_{\mathrm{Const}_1} + \underbrace{\frac{\alpha}{\alpha \cdot D_1^{i-1} + \mathrm{Diff}_2}}_{\mathrm{Const}_2} \cdot (D_1 - D_1^{i-1}) \quad (25)$$

where Const_1 and Const_2 are two constants that can be calculated first in each iteration. D_1^{i-1} is the optimal value of D_1 in the $i-1$th iteration. To guarantee the approximation accuracy, D_1 is restricted to be in the range of $[(1-\epsilon) \cdot D_1^{i-1}, (1+\epsilon) \cdot D_1^{i-1}]$ during the ith iteration. It is obvious that the approximation in (25) meets the above three requirements. Therefore, the iterative approximation using (25) will converge to a point satisfying the KKT conditions of the problem in (23). In this paper, we set ϵ to be 0.04 and on average 11 to 15 iterations are required to converge.

With the approximation in (25), the optimization problem in (23) is iteratively solved. During the ith iteration, (23) is converted into the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{D}} \quad & \sum_{i=1}^{N} D_i \\
\text{s.t} \quad & s + t + a_1 \cdot \sum_{j=2}^{N-1} \log(\alpha + \frac{\mathrm{Diff_{j+1}}}{\mathrm{D_i}}) \\
& + a_1 \cdot \log\frac{1}{\mathrm{D_N}} + (N-1) \cdot b_1 \leq R_{\mathrm{GOP}} \\
& G \cdot (\frac{a_0}{D_1 + b_0} + c_0) \leq s \\
& \mathrm{Const_1} + \mathrm{Const_2} \cdot (\mathrm{D_1} - \mathrm{D_1^{i-1}}) \leq \mathrm{t} \\
& 0 \leq B_i \leq B \\
& D_1 \leq (1+\epsilon) D_1^{i-1} \\
& D_1 \geq (1-\epsilon) D_1^{i-1}.
\end{aligned} \quad (26)$$

Now all the functions in the inequality constraints are convex. The objective function, being an affine function of D_i, is obviously a convex function of D_i. Therefore, according to [21], the optimization problem in (26) is a convex optimization problem and the optimal solution can be obtained with the interior-point methods [21]. For simplicity, in this paper, the scientific software CVX [23] is used instead to get the optimal solution.

The geometric understanding of solving (23) is shown in Fig. 6. Suppose the initial point of D_1 is $D_1^{(0)}$; then, $G(D_1)$ is approximated using (25) within the interval I_0 which is centered around $D_1^{(0)}$, and (23) is converted to the convex optimization problem in (26). After solving (26), assuming that the optimal solution is $D_1^{(1)}$, then the new interval I_1 can be established, and $g(D_1)$ is approximated with a new affine function of D_1 as in (25). Similarly, the optimal solution in I_1 can be obtained. We denote it by $D_1^{(2)}$ and continue a new iteration. So on and so forth, until the total distortion converges.

In the end, let the optimal bit allocation strategy be $(R_1^*, R_2^*, \ldots, R_N^*)$. In practical applications, one problem is that the generated bits of each frame cannot be the exact number of allocated bits because of the inaccuracy of R–Q model. Suppose the number of actual generated bits of the ith frame is R_i^{actual}; to fulfill the total bit budget, the final number of bits allocated to the ith frame, denoted by R_i^{final}, is adjusted as

$$\tilde{R}_i = (R_{\mathrm{GOP}} - \sum_{j=1}^{i-1} R_j^{\mathrm{actual}}) \cdot \frac{R_j^*}{\sum_{j=n}^{N} R_j^*} \quad (27)$$

$$R_i^{\mathrm{final}} = \mathrm{median}\{\mathrm{R_i^{min}}, \mathrm{R_i^{max}}, \tilde{\mathrm{R}}_\mathrm{i}\} \quad (28)$$

where the function median{a, b, c} returns the median value among a, b, and c. $R_i^{\max}$ is the maximum allocated bits for the ith frame to avoid the decoder buffer underflow, and $R_i^{\min}$ is the minimum allocated bits for the ith frame to avoid the decoder buffer overflow.

The parameters of the proposed IFDM in (11) and the R–D model in (15) are updated with the coded information during encoding. To elaborate, let $\hat{R}_t$, $\hat{D}_t$, $\hat{\sigma}_t^2$, and $\hat{Q}_t$ denote the actual generated bits, the distortion, the variance of DCT coefficients, and the employed QStep of the tth frame. Then, after the nth frame is encoded, α, β, and γ are updated as follows:

$$\Omega = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{29}$$

where $\Omega = (\alpha \quad \beta \quad \gamma)^T$, $\mathbf{y} = (\hat{\sigma}_n^2 \quad \hat{\sigma}_{n-1}^2 ... \hat{\sigma}_{n-H+1}^2)^T$, and X is a $H \times 3$ matrix defined as

$$\mathbf{X} = \begin{pmatrix} \hat{D}_{n-1} & \tilde{\sigma}_n^2 & \hat{Q}_n \\ \hat{D}_{n-2} & \tilde{\sigma}_{n-1}^2 & \hat{Q}_{n-1} \\ . & . & . \\ . & . & . \\ \hat{D}_{n-H} & \tilde{\sigma}_{n-H+1}^2 & \hat{Q}_{n-H+1} \end{pmatrix}$$

where H is the number of previous frames used for the parameter update. In this paper, H is set to be 12.

In addition, a_1 and b_1 in the R–D model (15) are updated as

$$a_1 = \frac{H\sum_{t=n-H}^{n} \hat{R}_t B_t - \sum_{t=n-H+1}^{n} \hat{R}_t \sum_{t=n-H+1}^{n} B_t}{H\sum_{t=n-H+1}^{n} B_t^2 - (\sum_{t=n-H+1}^{n} B_t)^2} \tag{30}$$

$$b_1 = \frac{\sum_{t=n-H+1}^{n} \hat{R}_t - a_1 \sum_{t=n-H+1}^{n} B_t}{H} \tag{31}$$

where $B_t = \log\frac{\hat{\sigma}_t^2}{\hat{D}_t}$.

Different from the parameter updating process in (11) and (15), the parameters in (13) are obtained by offline training. Let $(\hat{R}_n, \hat{D}_n)$ $(n = 1, 2, ..., N)$ be the empirical results of R–D pairs. Then, b_0 and c_0 are obtained by solving the following equation in the sense of least squares:

$$A \begin{bmatrix} c_0 \\ b_0 \end{bmatrix} = \begin{pmatrix} \hat{D}_1\hat{R}_1/G - \hat{D}_2\hat{R}_2/G \\ \hat{D}_2\hat{R}_2/G - \hat{D}_3\hat{R}_3/G \\ . \\ . \\ \hat{D}_{N-1}\hat{R}_{N-1}/G - \hat{D}_N\hat{R}_N/G \end{pmatrix} \tag{32}$$

where

$$A = \begin{pmatrix} \hat{D}_1 - \hat{D}_2 & \hat{R}_2/G - \hat{R}_1/G \\ \hat{D}_2 - \hat{D}_3 & \hat{R}_3/G - \hat{R}_2/G \\ . & . \\ . & . \\ \hat{D}_{N-1} - \hat{D}_N & \hat{R}_N/G - \hat{R}_{N-1}/G \end{pmatrix}. \tag{33}$$

After b_0 and c_0 are obtained, a_0 is obtained by

$$a_0 = \frac{1}{N}\sum_{n=1}^{N}(\hat{R}_n/G - c_0)(\hat{D}_n + b_0) \tag{34}$$

The values of a_0, b_0, and c_0 are stored in the lookup table which is shown in Table IV. In Table IV, the parameter values are associated with the average frame gradient G.

For better understanding, here is a summary of the proposed IFDM-DBA algorithm.

1) For each GOP, calculate the total available bits according to (20). Then, perform ME to the frames $i = 2, ..., N$ of the original video sequence and obtain the value of $\tilde{\sigma}_i^2$. In our implementation, PMVFAST [24] is used as the fast ME method. The block size for ME is chosen be 16×16 and only integer-pixel positions are checked during ME. By approximating Q_i with average Q in the previous GOP, $\mathrm{Diff_i}$ can be calculated according to (22).
2) Using the aforementioned successive convex approximation approach, (23) is iteratively solved. During each iteration, (23) is approximated into (26). The optimal solution of (26) is derived via the scientific software CVX [23]. Then, the optimal bit allocation strategy $(R_1^*, R_2^*, \ldots, R_N^*)$ can be obtained. Finally, to fulfill the total bit budget, the final number of bits allocated to each frame is adjusted according to (27) and (28).
3) During encoding, the parameters of the proposed IFDM in (11) and the R–D model in (15) are updated with the coded information according to (29)–(31).

TABLE IV

LOOKUP TABLE FOR THE PARAMETERS IN (13)

Value of G	a_0	b_0	c_0
$0 \le G < 5$	0.273	0.214	0.011
$5 \le G < 10$	0.419	0.689	0.020
$10 \le G < 15$	0.688	2.438	0.013
$15 \le G < 20$	0.859	4.330	0.010
$20 \le G < 25$	0.859	5.691	0.011
$25 \le G < 30$	0.865	7.220	0.015
$30 \le G < 35$	0.481	3.324	0.037
$35 \le G$	1.053	7.749	0.017

TABLE V

TEST SEQUENCES USED FOR THE EXPERIMENTS

Sequence	Resolution	Frame Rate	Total Frames
Akiyo	352 × 288	30	300
Coastguard	352 × 288	30	300
Foreman	352 × 288	30	300
Mobile	352 × 288	30	300
City	1280 × 720	30	300
Shuttlestart	1280 × 720	30	300
Cactus	1920 × 1080	30	300
Traffic	1920 × 1080	30	300

IV. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed IFDM-DBA algorithm, we implement it on the H.264 reference software JM 16.0. Eight video sequences listed in Table V are selected as the test sequences. Note that both standard-definition (SD) and high-definition (HD) video sequences are included. Moreover, the selected video sequences contain quite different video characteristics including slow and fast motions, smooth and complex sceneries. The GOP length is set to be 30, and the frame rate is 30 frames/s. *CABAC* is used as the entropy coding method, and the maximum search range for ME is ±32. RDO is enabled with high complexity mode. The buffer size is chosen to be the size of the target bitrate, and the initial buffer fullness is the half of the buffer size. For simplicity and fair comparison, the quadratic R–D model of the JM reference software is used. However, it should be noted that other more sophisticated R–Q models and more advanced QP selection methods can also be employed.

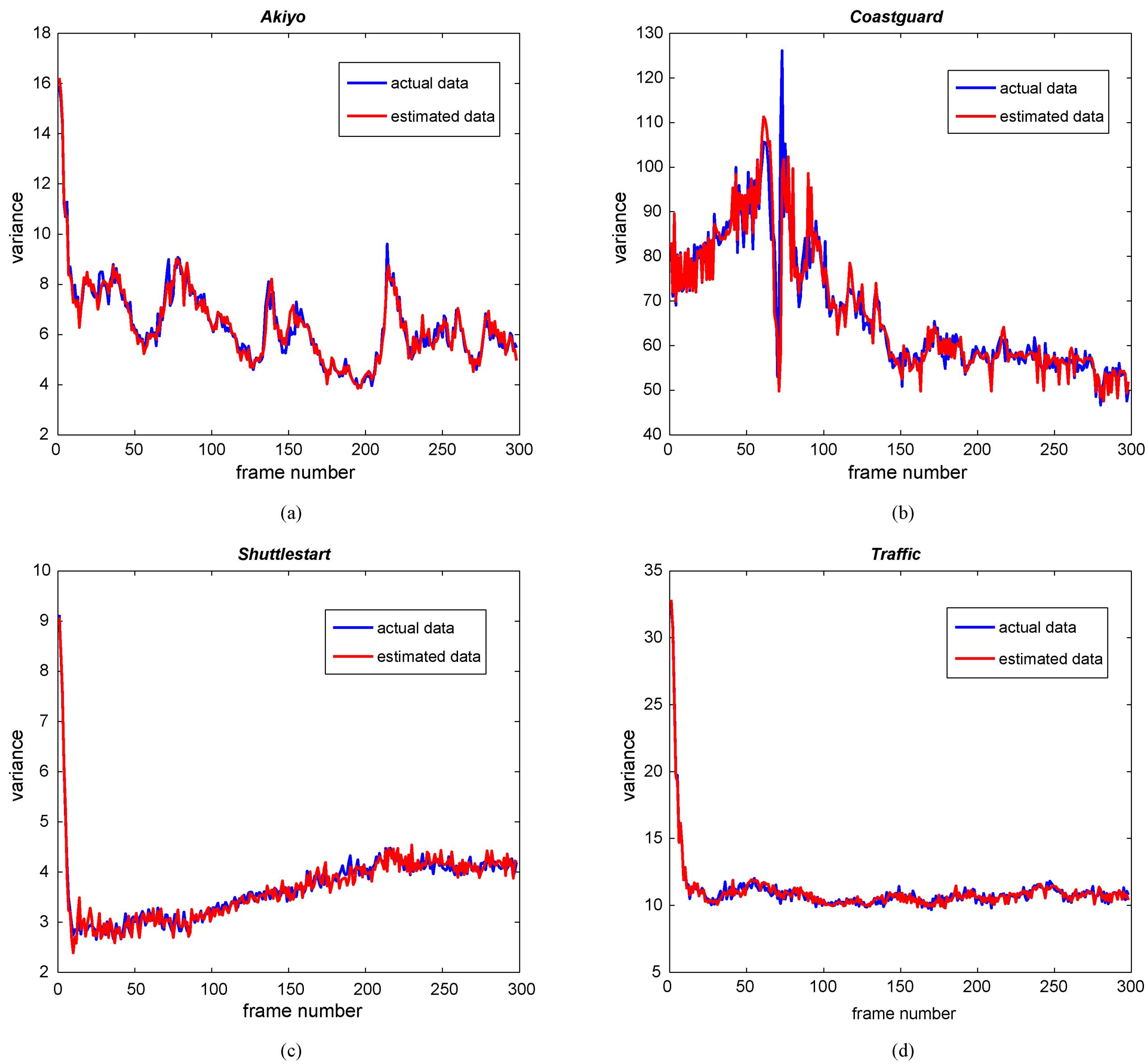

Fig. 7. Framewise estimation accuracy of the proposed IFDM for different video sequences. (a) *Akiyo*. (b) *Coastguard*. (c) *Shuttlestart*. (d) *Traffic*.

TABLE VI
ESTIMATION ACCURACY OF THE PROPOSED IFDM

Sequence	Target Bitrate (kb/s)	$\mathrm{IFDM_e}$
Akiyo	100	3.3%
Coastguard	400	2.9%
Foreman	400	3.1%
Mobile	800	3.3%
City	3000	3.2%
Shuttlestart	3000	2.9%
Cactus	8000	2.9%
Traffic	8000	2.0%

First, the estimation accuracy of the proposed IFDM is evaluated. For each video sequence, it is encoded using JM under a predefined target bitrate. Then, the actual variance and the estimated variance using IFDM of the DCT coefficients are compared. To have a quantitative measure, we define the estimation error $\mathrm{IFDM_e}$ as

$$\mathrm{IFDM_e} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{|\sigma^2_{\mathrm{i,est}} - \sigma^2_{\mathrm{i,act}}|}{\sigma^2_{\mathrm{i,act}}} \cdot 100\% \tag{35}$$

where N is the total number of frames. $\sigma^2_{i,\mathrm{est}}$ and $\sigma^2_{i,\mathrm{act}}$ are the estimated variance and the actual variance of the ith frame, respectively. The results of $\mathrm{IFDM_e}$ for the test sequences are summarized in Table VI. From Table VI, it can be seen that the average estimation error of the proposed IFDM is less than 3.0% and the maximum estimation error is less than 3.5%. Besides this overall estimation performance evaluation, we have plotted the framewise results of the actual and estimated variances of four test sequences as shown in Fig. 7.

Then, the R–D performance of the proposed IFDM-DBA algorithm is compared with two representative bit allocation methods: one is the BA method proposed in [25] and the other is the frame-level DBA algorithm proposed in [26].

TABLE VII
R--D PERFORMANCE COMPARISON WITH DIFFERENT BIT ALLOCATION METHODS

Sequence	Bandwidth (kb/s)	[25] PSNR (dB)	[26] PSNR (dB)	Proposed PSNR (dB)	over [25] BDPSNR (dB)	over [25] BDBR	over [26] BDPSNR (dB)	over [26] BDBR
Akiyo	100	39.12	39.24	39.99	0.95	−23.20%	0.79	−19.51%
	150	40.44	40.61	41.39				
	200	41.51	41.68	42.44				
	250	42.26	42.43	43.31				
Coastguard	300	30.21	30.27	30.42	0.31	−8.10%	0.23	−6.21%
	400	31.17	31.26	31.49				
	500	31.98	32.04	32.30				
	600	32.69	32.73	33.03				
Foreman	300	34.39	34.49	34.90	0.51	−11.62%	0.43	−9.92%
	500	36.43	36.53	36.98				
	700	37.92	37.98	38.36				
	900	38.97	38.98	39.38				
Mobile	500	27.81	27.87	28.21	0.36	−7.80%	0.34	−7.41%
	700	29.16	29.19	29.57				
	900	30.35	30.35	30.63				
	1100	31.35	31.35	31.60				
City	2500	34.98	35.33	35.36	0.36	−11.89%	0.10	−3.75%
	3000	35.53	35.81	35.90				
	3500	35.97	36.18	36.31				
	4000	36.36	36.48	36.67				
Shuttlestart	2500	43.29	43.45	43.64	0.37	−18.85%	0.22	−12.27%
	3000	43.62	43.79	44.01				
	3500	43.90	44.02	44.27				
	4000	44.14	44.27	44.46				
Cactus	7000	36.28	36.43	36.53	0.24	−10.62%	0.14	−6.90%
	8000	36.56	36.67	36.79				
	9000	36.81	36.88	37.06				
	10000	37.02	37.08	37.26				
Traffic	7000	38.12	38.14	38.78	0.65	−14.27%	0.60	−13.02%
	8000	38.69	38.72	39.36				
	9000	39.20	39.27	39.84				
	10 000	39.67	39.73	40.25				
Average					0.47	−13.29%	0.36	−9.88%

The experimental results are summarized in Table VII. In the experiment, Bjontegaard bitrate (BDBR) and PSNR (BDPSNR) [27] are deployed to measure the average performance over different bitrates. For BDBR, a negative number in the table indicates a rate reduction achieved by the proposed scheme at the same visual quality. As shown in the table, on average the proposed IFDM-DBA algorithm has 0.47 and 0.36 dB BDPSNR improvement over [25] and [26], respectively. Or equivalently, up to 13.29% and 9.88% bitrate savings are achieved. In addition, for the video sequence containing less motion, such as *Akiyo*, the proposed IFDM-DBA algorithm has 0.95 and 0.79 dB BDPSNR improvement over [25] and [26]. Thus, up to 23.20% and 19.51% bitrate can be saved. In addition, to compare the relative behavior of [25], [26], and IFDM-DBA, we show their instantaneous framewise PSNR for two representative sequences (*Foreman* and *City*) in Fig. 8.

Next, we compare the buffer status with different bit allocation algorithms. As explained earlier, during the bit allocation, the buffer fullness needs to be regulated such that buffer overflow or underflow will not happen. The buffer status with different bit allocation algorithms for *Foreman* and *City* is shown in Fig. 9. It can be seen that all the three methods can always make sure the buffer fullness is at a secure level.

Finally, we compare briefly the computational complexity of the three bit allocation methods. All simulations are run on the same computer with Intel Core i3 CPU @ 3.07 GHz and 8-GB RAM. The results of total running time with different joint bit allocation methods are summarized in Table VIII. As shown in the table, both [25] and [26] are simple and consume similar total running time. Compared with [25] and [26], the proposed IFDM-DBA method needs to employ CVX [23] software to solve a series of convex optimization problems using successive convex approximation techniques. In our experiments, comparing with [25] and [26], IFDM-DBA causes the overall time to increase by about 22%. During the experiment, we find that the majority of the additional encoding time is spent in using CVX to solve the convex optimization problems. The JM reference software is written in C language, and the CVX software is run in MATLAB environment. Much time is wasted on communication between the C program and the MATLAB program which is not as efficient as C program. If the convex optimization problems can be solved using C program instead, the additional computational complexity is expected to be much reduced and this will be our future work.

TABLE VIII
TOTAL RUNNING TIME COMPARISON WITH DIFFERENT BIT ALLOCATION METHODS

Sequence	Bandwidth (kb/s)	[25] (s)	[26] (s)	IFDM-DBA method (s)
Akiyo	200	905	923	1276
Coastguard	500	1008	1031	1401
Foreman	700	1112	1133	1560
Mobile	900	1253	1283	1727
City	3500	10 051	10 231	10 633
Shuttlestart	3500	8022	8134	8519
Cactus	9000	20 123	20 304	20 887
Traffic	9000	25 586	25 867	26 532

V. CONCLUSION

In this paper, the conventional frame-level DBA problem in hybrid video coding is revisited and an analytic framework is proposed. The contributions of this paper are threefold.

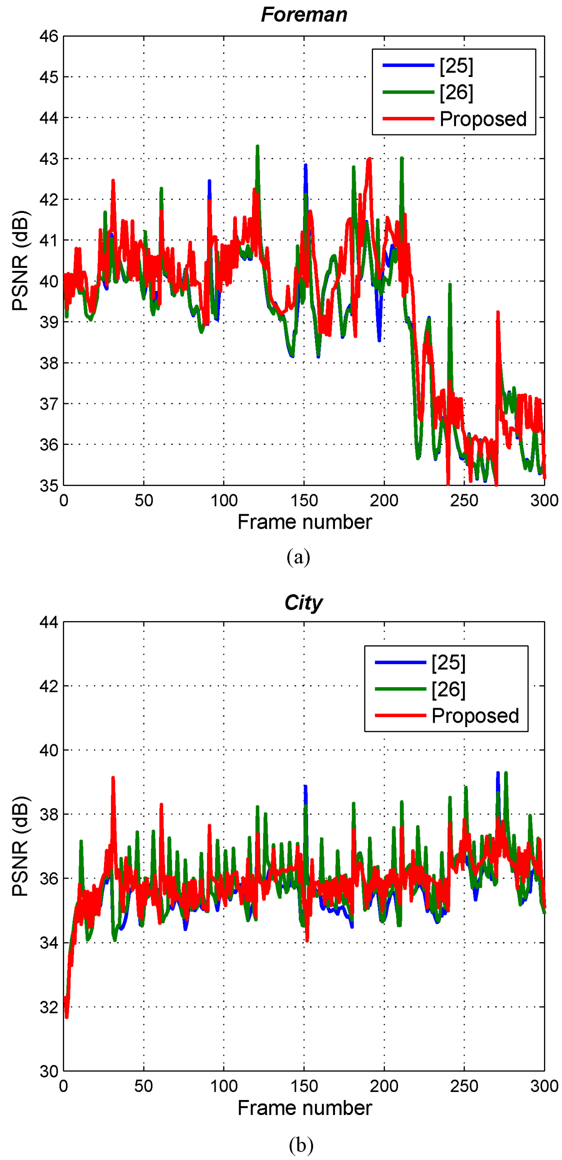

Fig. 8. Comparison of framewise PSNR with different bit allocation methods. (a) *Foreman* (900 kb/s). (b) *City* (3 Mb/s).

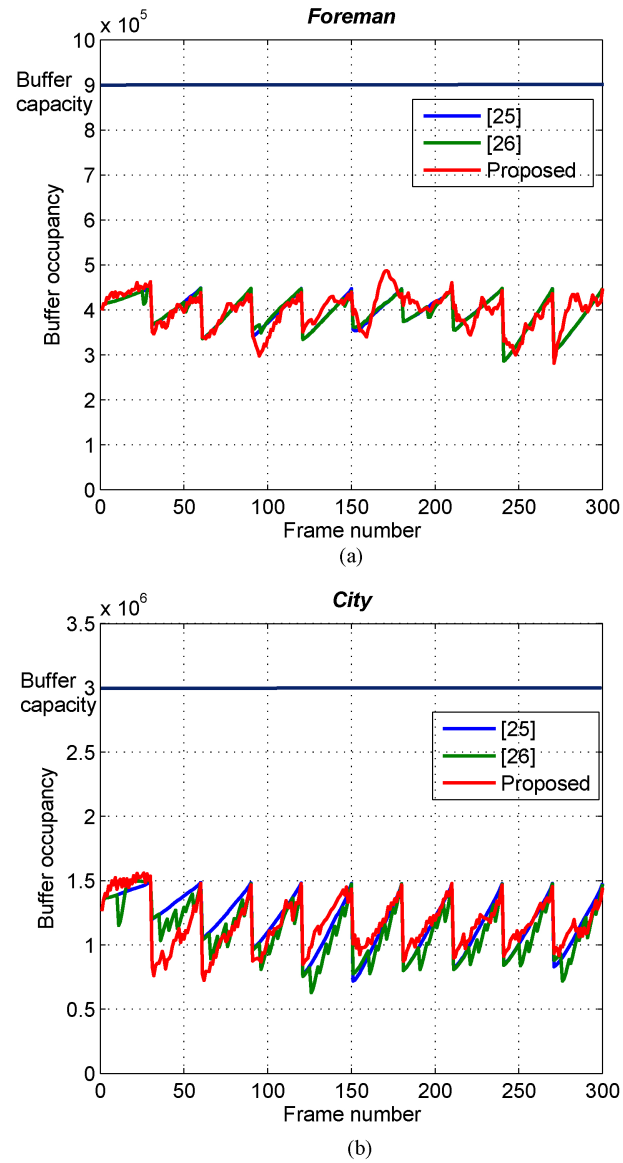

Fig. 9. Comparison of buffer status with different bit allocation methods. (a) *Foreman* (900 kb/s). (b) *City* (3 Mb/s).

First, a novel IFDM is proposed to quantitatively measure the coding dependency. Second, based on IFDM, the buffer-constrained frame-level DBA problem is carefully formulated. Third, the model-based approach called IFDM-DBA method is proposed to solve the original problem using successive convex approximation techniques. The experimental results have demonstrated the superiority of the proposed method with appealing coding performance improvement.

Although currently the proposed DBA is designed for the video coding with IPPP GOP structure, we expect the proposed method to be applicable to other GOP structures, such as IBBP, as well. Also, the situation involving multihypothesis prediction should be considered, and this will be our future work.

Appendix

By combining the constraints in (18), we can get

$$\begin{aligned}
\min_{\mathbf{R},\mathbf{D}} \quad & \textstyle\sum_{i=1}^{N} D_i \\
\text{s.t.} \quad & R_1 + a_1 \cdot \textstyle\sum_{j=2}^{N} \log \frac{\alpha \cdot D_{j-1} + \beta \cdot \tilde{\sigma}_j^2 + \gamma \cdot Q_j}{D_j} \\
& +(N-1) \cdot b_1 \leq R_{\text{GOP}} \\
& R_1 = G \cdot \left(\tfrac{a_0}{D_1 + b_0} + c_0\right) \\
& 0 \leq B_i \leq B.
\end{aligned} \tag{36}$$

Note that we have

$$\begin{aligned}
& \sum_{j=2}^{N} \log \frac{\alpha \cdot D_{j-1} + \beta \cdot \tilde{\sigma}_j^2 + \gamma \cdot Q_j}{D_j} \\
= \; & \log(\alpha \cdot D_1 + \beta \cdot \tilde{\sigma}_2^2 + \gamma \cdot Q_2) \\
+ \; & \sum_{j=2}^{N-1} \log\left(\alpha + \frac{\beta \cdot \tilde{\sigma}_{j+1}^2 + \gamma \cdot Q_{j+1}}{D_j}\right) + \log \frac{1}{D_N}.
\end{aligned} \tag{37}$$

Thus, by introducing the slack variables s and t, (18) is equivalent to the optimization problem in (20).

REFERENCES

[1] 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.

[2] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for H.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.

[3] 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. 68–78, Jan. 2007.

[4] Z. He and S. 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. 840–849, Oct. 2002.

[5] Z. He, S. Mitra, and S. K. Mitra, "Low-delay rate control for DCT video coding via ρ-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 970–982, Nov. 2002.

[6] Y. Altunbasak, N. Kamaci, 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.

[7] C. Pang, O. C. Au, J. Dai, F. Zou, W. Yang, and M. Yang, "Laplacian mixture model (LMM) based frame-layer rate control method for h.264/avc high-definition video coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2010, pp. 25–29.

[8] J. Ribas-Corbera and S.-M. Lei, "A frame-layer bit allocation for H.263+," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 10, no. 7, pp. 1154–1158, Oct. 2000.

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

[10] 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.

[11] K. Ramchandran, A. Ortegaand, and M. Vetterli, "Bit allocation for dependent quantization with application to multi-resolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 533–545, Sep. 1994.

[12] Y. Sermadevi and S.-S. Hemami, "Efficient bit allocation for dependent video coding," in *Proc. IEEE Data Compresssion Conf.*, Mar. 2004, pp. 232–241.

[13] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[14] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion charateristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 446–459, Aug. 1998.

[15] X. Li, A. Hutter, and A. Kaup, "One-pass frame level budget allocation in video coding using inter-frame dependency," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2009, pp. 1–6.

[16] C. Pang, O. C. Au, F. Zou, and J. Dai, "An analytical framework for frame-level dependent bit allocation in hybrid video coding," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2011, pp. 1–4.

[17] J. L. Devore and N. R. Farnum, *Applied Statistics for Engineers and Scientists*. New York: Duxbury, 1999.

[18] Z. He and D. O. Wu, "Linear rate control and optimum statistical multiplexing for H.264 video broadcast," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1237–1249, Nov. 2008.

[19] C. Pang, O. C. Au, J. Dai, and F. Zou, "Frame complexity guided Lagrange multiplier selection for H.264 intra-frame coding," *IEEE Signal Process. Lett.*, vol. 18, no. 12, pp. 733–736, Dec. 2011.

[20] T. Berger, Ed., *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[21] S. Boyd and L. Vandenberghe, Eds., *Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[22] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neil, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, Jul. 2007.

[23] M. Grant and S. Boyd. (2011, Apr.). CVX: Matlab software for disciplined convex programming, version 1.21 [Online]. Available: http://cvxr.com/cvx

[24] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST)-enhancing block based motion estimation," in *Proc. Vis. Commun. Image Process.*, vol. 4310. Jan. 2001, pp. 883–892.

[25] Z. Li, F. Pan, K. P. Lim, and S. Rahardja, "Adaptive rate control for H.264," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, pp. 745–748.

[26] X. Li, A. Hutter, and A. Kaup, "Efficient one-pass frame level rate control for H.264/AVC," *J. Vis. Commun. Image Representation*, vol. 20, no. 8, pp. 585–594, Nov. 2009.

[27] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T Video Coding Experts Group, Document VCEG-M33, Austin, TX, Apr. 2001.

Chao Pang (M'12) received the B.E. degree in telecommunications engineering from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2012.

He is currently a Research Associate with the Hong Kong University of Science and Technology. He was a Research Intern with Microsoft Research Asia, Beijing, China. He has published more than 25 technical journal and conference papers. He is actively involved in the standardization of high-efficiency video coding and its extensions. His current research interests include image/video coding and processing, video streaming, and wireless communication.

Oscar C. Au (F'12) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1986, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1988 and 1991, respectively.

After being a Post-Doctoral Researcher at Princeton University for one year, he joined the Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, as an Assistant Professor in 1992, where he is/has been a Professor in the Department of Electronic and Computer Engineering, the Director of Multimedia Technology Research Center, and the Director of the Computer Engineering Program. His current research contributions are on video and image coding and processing, watermarking and light-weight encryption, and speech and audio processing. His research topics include fast motion estimation for MPEG-1/2/4, H.261/3/4, and audio video standard (AVS), optimal and fast suboptimal rate control, mode decision, transcoding, denoising, deinterlacing, postprocessing, multiview coding, view interpolation, depth estimation, 3DTV, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000, HDR imaging, compressive sensing, half-tone image data hiding, GPU processing, software–hardware codesign, etc. He has published 50+ technical journal papers, 300+ conference papers, and 70+ contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His lightweight encryption and error resilience algorithms are accepted into the China AVS standard. He was the Chair of Screen Content Coding AdHoc Group in the Joint Collaborative Team on Video Coding for the upcoming high-efficiency video coding standard. He holds 17 granted U.S. patents and is applying for 70+ more on his signal processing techniques. He has performed forensic investigation and stood as an expert witness in the Hong Kong courts many times.

Dr. Au is a Board of Governor member of the Asia Pacific Signal and Information Processing Association. He is/was Associate Editors of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR—PART I: REGULAR PAPERS. He is on the Editorial Boards of the Journal of Visual Communication and Image Representation, the Journal of Signal Processing Systems, the Journal of Multimedia, and the Journal of Franklin Institute. He is/was the Chair of the IEEE Circuits and Systems (CAS) Technical Committee (TC) on Multimedia Systems and Applications, the Chair of the Signal Processing (SP) TC on Multimedia Signal Processing, and a member of the CAS TC on Video Signal Processing and Communications, CAS TC on DSP, and SP TC on Image, Video and Multidimensional Signal Processing. He has served on the Steering

Committee of the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE International Conference of Multimedia and Expo (ICME). He also served on the organizing committee of the IEEE International Symposium on Circuits and Systems in 1997, the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003, the ISO/IEC MPEG 71st Meeting in 2005, the International Conference on Image Processing in 2010, and other conferences. He was the General Chair of the Pacific-Rim Conference on Multimedia (PCM) in 2007, the ICME in 2010 and the International Packet Video Workshop in 2010. He was a recipient of the Best Paper Award at the IEEE Workshop on Signal Processing Systems 2007 and PCM 2007. He was an IEEE Distinguished Lecturer in 2009 and 2010, and has been a keynote speaker for a few times.

Feng Zou (S'10) received the B.S. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2008. He is currently pursuing the Ph.D. degree in electronic and computer engineering at the Hong Kong University of Science and Technology, Kowloon, Hong Kong, under the supervision of Prof. O. Au.

He is also a Research Intern in Mitsubishi Electric Research Laboratories, Cambridge, MA. His current research interests include intraprediction, transform and quantization designs in 2-D video compression, as well as view synthesis prediction for 3-D video compression. He is actively contributing proposals in the standardization of high-efficiency video coding and its extensions under the ITU-T/ISO/IEC Joint Collaborative Team on Video Coding and 3-D Video Coding and holds several video coding-related patents.

Jingjing Dai (M'12) received the B.E. degree in information engineering from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China in 2008, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong in 2012.

Her current research interests include image/video processing, next-generation video coding, and video codec optimization.

Xingyu Zhang (S'12) received the B.E. degree from the School of Marine Engineering, Northwestern Polytechnical University, Xi'an, China in 2009. She is currently pursuing the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong.

She has several proposals that have been adopted by the video coding standard high-efficiency video coding. Her research interests include video coding standards, interchannel prediction, scanning order, and data hiding in video coding.

Wei Dai (S'13) received the B.E. (with the highest Hons.) degree from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China in 2010. He is currently pursuing the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong.

He has published several conference papers on motion estimation, rate control, and image processing. He has submitted several proposals to the video coding standard high-efficiency video coding. His research interests include motion estimation, multiview video coding, and scalable video coding.