Constant Bit-Rate Multi-Stage Rate Control for Rate-Distortion Optimized H.264/AVC Encoders

Shuijing WU††, Student Member, Peilin LIU†, Yiqing HUANG††, Qin LIU†††, Nonmembers, and Takeshi IKENAGA†††, Member

SUMMARY H.264/AVC encoder employs rate control to adaptively adjust quantization parameter (QP) to enable coded video to be transmitted over a constant bit-rate (CBR) channel. In this topic, bit allocation is crucial since it is directly related with actual bit generation and the coding quality. Meanwhile, the rate-distortion-optimization (RDO) based mode-decision technique also affects performance a lot for the strong relation among mode, bits, and quality. This paper presents a multi-stage rate control scheme for R-D optimized H.264/AVC encoders under CBR video transmission. To enhance the precision of the complexity estimation and bit allocation, a frequency-domain parameter named mean-absolute-transform-difference (MATD) is adopted to represent frame and macroblock (MB) residual complexity. Second, the MATD ratio is utilized to enhance the accuracy of frame layer bit prediction. Then, by considering the bit usage status of whole sequence, a measurement combining forward and backward bit analysis is proposed to adjust the Lagrange multiplier $\lambda_{MODE}$ on frame layer to optimize the mode decision for all MBs within the current frame. On the next stage, bits are allocated on MB layer by proposed remaining complexity analysis. Computed QP is further adjusted according to predicted MB texture bits. Simulation results show the PSNR improvement is up to 1.13 dB by using our algorithm, and the stress of output buffer control is also largely released compared with the recommended rate control in H.264/AVC reference software JM13.2.

key words: H.264/AVC video CBR transmission, multi-stage rate control, rate-distortion optimization, Lagrange multiplier, bit allocation

1. Introduction

With the growing demand of applications for stable transmission of high quality visual information, various video coding standards are developed. As the newest international standard, H.264/AVC [1] becomes increasingly common. It achieves a much higher coding efficiency compared with that of any other existing video coding standards [2]. With so many high-level techniques adopted in H.264/AVC and its encoding, such as the rate-distortion optimized (RDO) variable block-sized motion estimation (ME) and mode decision, spatial prediction from the edges of neighboring blocks for intra coding, advanced exact-match integer block transform, multi-picture reference prediction, arithmetic entropy coding and so on [3], the source redundancies are further exploited by encoder. However, the system complexity is increased, and the constant bit rate (CBR) transmission becomes much more challenging, especially for those real-time communication applications, such as videophone and video conference. To regulate output bit-stream to meet the requirement of channel bandwidth and buffer constraints, rate control is employed in video coding system. Its basic function is to allocate a budget of bits to coding units, such as a group of pictures (GOP), frame, field, or one single macroblock (MB), adjust coding parameters according to unit complexity and bit resource. It also aims to enhance the coding quality for the CBR transmission. Actually, in order to smooth out the variable output rate, many video standards have adopted rate control as an important part, such as TM5 [4] for MPEG-2, TMN8 [5] for H.263, and VM-18 [6] for MPEG-4. For the more complicated standard H.264/AVC, the JVT-G012 [7] and [8] proposed by Li et al. forms the basis of the recommended rate control scheme, which has been widely studied and also has been implemented in the H.264/AVC reference software JM [9].

As we all know that in H.264/AVC rate control framework, the input video complexity is estimated first. Its accuracy is very important since it is one of the key factors to perform the bit allocation and quantization parameter (QP) generation. The mean-absolute-difference (MAD) is chosen to represent the coding complexity in most of previous works, such as [7]. This spatial-domain residual information sometimes is not precise enough. Hence, a frequency-domain parameter is adopted in this paper due to its slight enhancement in precision. On another aspect, some complexity estimation schemes, for example the one proposed in [10] will not be used in this paper since their high computational complexity.

Bit allocation is one of the most important parts in rate control. In many existing schemes, such as [7], [11] and [12], bits are allocated according to the target bit-rate and virtual buffer status on GOP and frame layer. Although the information from both of the buffer level and remaining bit resource values are utilized together, the calculations for certain parts are oversimplified that cannot represent the redundancy change in all pre-coded frames. So some approaches which take into account of video complexity or other optimized measurements have been proposed, such as [13] and [14]. Moreover, when rate control is enabled on MB layer, the complexity fluctuation between current frame and the previous one does not receive enough attention throughout entire consecutive execution in related works such as [15].
and [16]. The remaining complexity model utilized in [7] is too simple to predict the bits required by upcoming MB. It makes MB layer bit allocation inaccurate. On another aspect, rate control can be considered with mode decision in the universal point of view. Best mode selection for each MB under channel bandwidth constraint will affect the coding quality and actual bit-rate a lot [17]. Thus, the idea of RDO proposed in [17] and [18] should be fully considered when performing rate control.

Based on aforementioned observations, a R-D optimized rate control scheme is presented in this paper, considering complexity estimation, frame/MB layer bit allocation, and Lagrange Multiplier adjustment based global mode decision. The block diagram of encoding one frame with rate control is shown in Fig. 1. Proposed schemes have been marked in grey color. For both frame and MB layer complexity estimation, the spatial-domain representation MAD is replaced by the mean-absolute-transform-difference (MATD), which is a frequency-domain parameter with slight high precision. The frame layer bit allocation will be executed with improved complexity estimation and complexity fluctuation analysis. Then, the Lagrange multiplier ($\lambda_{MODE}$) is adjusted with coded and remaining bits information. The proposed scheme adopts the forward and backward bit usage analysis, combines them together to achieve optimization for the whole sequence. On MB layer, all the actual MATD information in previous and current frame is fully utilized to predict total target bits. After that the computed QP will also be modified according to the number of predicted MB texture bits.

The rest of this paper is organized as follows. A frequency-domain complexity estimation parameter MATD is introduced in Sect. 2. Section 3 gives out the proposed rate control schemes based on detail description of three sub-schemes. In Sect. 4, the overall multi-stage RDO based rate control algorithm is described. Experiment results and corresponding discussions are given in Sect. 5. The paper concludes with section 6.

2. Residual Complexity Estimation

As mentioned in Sect. 1, the MAD is widely adopted to represent the coding complexity, especially for the H.264/AVC. Its merit is that it can be obtained by very simple operation shown in (1) since it is only the spatial-domain signal representation. $SAD_{16 \times 16}$ and $\text{MAD}_{mb}$ is the sum-absolute-difference (SAD) and MAD values of MB, respectively. $C_m(x, y)$ and $P_m(x, y)$ are the pixel values with the position $(x, y)$ in current and previous reconstructed frames, respectively. The suffix $m$ is the pixel number in each $16 \times 16$ block (MB).

$$SAD_{16 \times 16} = \sum_{n=1}^{16 \times 16} |C_m(x, y) - P_m(x, y)|$$

$$\text{MAD}_{mb} = \frac{SAD_{16 \times 16}}{16 \times 16}$$

However, sometimes it cannot accurately reflect the trait of the residual information in video data. To have more precise estimation, a frequency-domain parameter, the mean-absolute-transform-difference (MATD) is introduced in this paper.

$$SATD_{4 \times 4} = \sum_{n=1}^{4 \times 4} |T[C_m(x, y) - P_m(x, y)]|$$

$$\text{MATD}_{mb} = \frac{\sum_{n=1}^{N_{mb}} SATD_{4 \times 4_n}}{16 \times 16}$$

The $4 \times 4$ Hadamard transform is executed for complexity prediction in (2). Similar to that in Eq. (1), the suffix $m$ here denotes the pixel number in each $4 \times 4$ sub-block. $SATD_{4 \times 4}$ is the SATD value of each $4 \times 4$ block using Hadamard transform. $\text{MATD}_{mb}$ and $\text{MATD}_{f}$ are the MATD values of each MB and frame, respectively. Notice that the “!” in $\text{MATD}_{mb}$ and $\text{MATD}_{f}$ means the calculations contain the partly Hadamard transform. Considering the computational complexity and the coherence with the transform used in RDO process, complete transform is only applied to $4 \times 4$ sub-block (the smallest block size in H.264/AVC). The sum of $SATD_{4 \times 4}$ are used to represent the SATD values for MB and frame. $N_{mb}$ is the total number of MBs in one frame. MATD is adopted to represent residual complexity instead of MAD because of its slightly better performance in the source rate model [19]. The benefit of transformed-difference is that the prediction accuracy is enhanced from both the viewpoint of objective and subjective metrics.

The performance comparison for entire proposed algorithm with the algorithm with MAD is given in Table 1. The information of test environment is described in Sect. 5. While other proposed sub-schemes are all implemented, the only difference is the residual complexity measurement. The “average PSNR Gain” denotes the improvement compared with the conventional work JVT-G012 [7] in JM13.2 [9]. From tested examples, the better performance by adopting MATD can be observed.
### Table 1 Performance Comparison for Various Algorithms with the Proposed Algorithm Based on MAD.

| Sequence       | Target (Kbps) | Average PSNR Gain (dB) | Bit Rate (Kbps) | Rate Mismatch |
|----------------|---------------|------------------------|-----------------|--------------|
|                |               | MAD | MAD | Δ Gain | MAD | MAD | MAD | MAD |
| grandmix.qcif  | 64            | 0.64 | 0.94 | -0.30 | 64.07 | 64.07 | 0.11% | 0.11% |
| panzoom.qcif   | 96            | 0.34 | 0.51 | -0.17 | 96.07 | 96.02 | 0.07% | 0.02% |
| claire.qcif    | 36            | 0.57 | 0.81 | -0.24 | 36.06 | 36.05 | 0.16% | 0.13% |
| akiyo.cif      | 36            | 0.40 | 0.38 | -0.18 | 36.06 | 36.06 | 0.16% | 0.16% |

3. Key Sub-Schemes in Proposed Rate Control

Corresponding to the three key sub-stages in rate control, some new schemes are proposed respectively in this section: 1. GOP layer bit allocation; 2. Frame layer bit allocation; 3. Lagrange multiplier adjustment; 4. MB layer bit allocation and QP adjustment. The performance of each part will be affected by the results of others since these sub-stages perform as an entirety.

#### 3.1 GOP Layer Bit Allocation

In this layer, the total remaining bits $T_r(j)$ after coding $(j - 1)^{th}$ frame ($j : 1 \sim N_P$, is $P$ frame suffix, $N_P$ is the number of total $P$ frames) will be computed for all remaining frames as in (3), where $u$ and $F_r$ are the target bandwidth and frame rate pre-determined for whole sequence. $A(j - 1)$ is the actual generated bits in the $(j - 1)^{th}$ frame. Assume the first frame in sequence is coded as $I$ frame while remaining ones are coded as $P$ frames.

$$T_r(j) = \begin{cases} \frac{u}{F_r} \times (N_P + 1) & j = 1 \\ T_r(j - 1) - A(j - 1) & j \geq 2 \end{cases}$$

(3)

After encoding each frame, the buffer occupancy will be updated by (4). $B_r(j)$ denotes buffer occupancy after coding $(j - 1)$ frames. For CBR rate control, the details of GOP bit allocation algorithm is described in [7].

$$\begin{align*}
B_r(0) &= 0 \\
B_r(j + 1) &= B_r(j) + A(j) - \frac{u}{F_r}
\end{align*}$$

(4)

#### 3.2 Frame Layer Bit Allocation

The target bits $T_f$ of $j^{th}$ frame is predicted by (5). $\beta$ is a constant with the typical value of 0.50 [9].

$$T_f = (1 - \beta) \times T_{buf,f} + \beta \times T_{rf,f}$$

(5)

$T_{buf,f}$ and $T_{rf,f}$ denotes the current $j^{th}$ frame target bit estimated from current buffer status and remaining bits, respectively. $T_{buf,f}$ is calculated by (6). $B_r(j)$ denotes the target buffer level of the $j^{th}$ frame. $\gamma$ is a constant with typical value of 0.50 [9].

$$T_{buf,f} = \left( \frac{u}{F_r} \right) - \gamma \times (B_r(j) - B_r(j))$$

(6)

In order to improve the accuracy of target bit prediction, the frame complexity fluctuation in pre-coded frames are fully considered in this paper when computing $T_{rf,f}$. The distribution of bit count is scaled by $MATD_{Ratio,j}$, as shown in (7) and (8). In (9), $MATD_P$ is the predicted MATD value of current $j^{th}$ frame, while $MATD_A$ is the actual value. Linear prediction model in [7] is remained in this paper. Current frame residual complexity is predicted from the actual information of its reference. $x_j$ and $y_j$ are model parameters that will be updated after coding each frame [7]. $N_r$ is the number of total remaining frames.

$$T_{rf,f} = \begin{cases} 0.4 \times \frac{T_r(j)}{N_r}, & MATD_{Ratio,j} < 0.5 \\
0.8 \times MATD_{Ratio,j} \times \frac{T_r(j)}{N_r} & 0.5 \leq MATD_{Ratio,j} < 1.5 \\
1.5 \times \frac{T_r(j)}{N_r}, & MATD_{Ratio,j} \geq 1.5
\end{cases}$$

(7)

$$MATD_{Ratio,j} = \frac{MATD_P}{\sum_{i=1}^{j-1} MATD_A}$$

(8)

$$MATD_P = x_j \times MATD_A + y_j$$

(9)

#### 3.3 Adaptive Lagrange Multiplier Adjustment

With the conception of rate distortion theory, mode decision is made by (10), where $\lambda_{MODE}$ is the Lagrangian cost, $D$ denotes sum of squared difference between original block and its reconstructed one, $R$ is the actual number of bits associated with a chosen mode. $\lambda_{MODE}$ is Lagrange multiplier given in [17], $QP$ is the MB quantization parameter.

$$\begin{align*}
\lambda_{MODE} &= \left\{ \begin{array}{ll}
\min\{JMODE\} & = \min\{D + \lambda_{MODE} \times R\}
\end{array} \right. \\
\lambda_{MODE} &= 0.85 \times 2^{10.12}
\end{align*}$$

(10)

The effect of $\lambda_{MODE}$ in mode selection is described in [15]. It is obvious that different values of $\lambda_{MODE}$ corresponds different rate-distortion (RD) characteristics. From the expression in (10), since $\lambda_{MODE}$ is multiplied by rate value $R$, it can be found that when the value of $\lambda_{MODE}$ becomes larger, $R$ will have more impact to the RD cost. Generally, a large value of $\lambda_{MODE}$ corresponds to higher distortion and lower bit-rate. $\lambda_{MODE}$ is the key to this tradeoff. Based on this observation, we can change the rate and distortion of upcoming picture by modifying $\lambda_{MODE}$.

The same as the residual complexity information, in one-pass coding, we can never know the exact number of
Table 2: Performance comparison for entire proposed algorithm with the algorithm without Lagrange multiplier adjustment.

| Sequence       | Target (Kbps) | Average PSNR Gain (dB) Without | Bit-Rate (Kbps) Without | Rate Mismatch Without |
|---------------|--------------|-------------------------------|------------------------|-----------------------|
|               |              | \(\Delta_{\text{adj}}\) | \(\Delta\) Gain | \(\Delta_{\text{adj}}\) | \(\Delta\) Gain | \(\Delta_{\text{adj}}\) | \(\Delta\) Gain |
| hallmonitor.cif | 36           | 0.19                          | 0.33                   | 0.14                  | 0.36          | 0.05%               | 0.14%               |
| M & D.cif      | 36           | 0.06                          | 0.17                   | 0.11                  | 0.36          | 0.14%               | 0.14%               |
| carphone.qcif  | 36           | 0.31                          | 0.39                   | 0.08                  | 0.36          | 0.11%               | 0.11%               |
| salesman.qcif  | 64           | 0.27                          | 0.36                   | 0.09                  | 0.63          | 0.05%               | 0.05%               |

bits for each frame before real coding [7]. Thus, in order to select better mode, and utilize the bit resource appropriately, it is reasonable to consider the information of already utilized bits, current predicted bits, and the remaining bits to modify \(\lambda_{\text{MODE}}\) in a universal point of view. Based on this idea, the proposed \(\lambda_{\text{MODE}}\) adjustment is given by (11). \(f_R\) is a function represents bit usage measurement determined by context-adaptive bit analysis. In order to obtain this function, firstly the bit usage among current \(P\) frame, all pre-coded \(P\) frames, and all remaining ones are analyzed by (12) and (13).

\[
\lambda_{\text{MODE}} = 0.85 \times f_R \times 2^{0.22} \tag{11}
\]

\[
f_R = \frac{w_j}{v_j} \quad \text{for } j > 1 \tag{12}
\]

\[
w_j = \frac{1}{T_j} \times \sum_{l=1}^{L_j} B_{j,l}, \quad v_j = \frac{T_j}{\sum_{l=1}^{L_j} B_{j,l}} \tag{13}
\]

The \(f_R\) computed by (12) is a coefficient for \(j^{th}\) \(P\) frame which considering both \(\omega_j\) and \(v_j\). \(B_{j,l}\) denotes the actual bits utilized for encoding \(j^{th}\) frame, \(T_P\) is the total target bits for all \(P\) frames, and \(T_j\) is the predicted target bits for \(j^{th}\) frame. In (13), \(\omega_j\) and \(v_j\) are backward and forward adjusting factors, representing pre-coded \(P\) frames bit status and the relationship between current and all remaining frames, respectively. \(\omega_j\) is the measurement for the bit usage of the pre-coded part. When \(\omega_j\) is large, it means that the coded frames may very likely have utilized too many bits, and we should increase \(\lambda_{\text{MODE}}\) to limit the bit usage upon the upcoming frame. By contrary, if it is small, the rest frames may have rich bit resource, and the corresponding \(\lambda_{\text{MODE}}\) should also be small. The operator \(v_j\) is a factor represents the complexity proportion of current frame among all not-yet-coded frames. Larger the \(v_j\) is, more complexity the current frame holds compared with other remaining ones. Of course, this frame should be allocated more bits to encode. So the change of \(v_j\) is inverse to the trend of \(\lambda_{\text{MODE}}\). The illustration of this bidirectional adjustment conception is also given in Fig. 2.

The position of current frame in whole sequence is also considered in our method. Generally, since the algorithm is based on a frame by frame flow, anterior frames often get more bits for encoding, and the bits left for remaining frames usually decrease gradually. It is not reasonable for the sequence with different feature. For example, if the latter part holds high complexity, the coding quality of whole sequence will drop significantly. Thus, \(f_R\) is scaled by following function (14), where \(a_0\) and \(a_1\) are constants with typical values of 1.2 and 0.2, respectively. These two constants are defined by the extensive experiments and corresponding analysis. The final \(f_R\) is defined by (15) and utilized to adjust corresponding Lagrange multiplier \(\lambda_{\text{MODE}}\) in Eq. (11).

\[
f_R(j) = \left( a_0 - \frac{a_1}{N_p} \times j \right) \times f_R(j) \tag{14}
\]

\[
f_R(j)_{\text{final}} = \begin{cases} 
1.0, & f_R(j) < 1.1 \\
1.0 + 0.4 \times (f_R(j) - 1.1), & 1.1 \leq f_R(j) < 1.6 \\
1.2, & f_R(j) \geq 1.6
\end{cases} \tag{15}
\]

The performance comparison for entire proposed algorithm with the algorithm without Lagrange multiplier adjustment is given in Table 2 to show the effect of the sub-scheme described in Sect. 3.3. The test environment is given in Sect. 5. From the table and the extensive experiments, it can be noticed that when Lagrange multiplier adjustment is removed from the proposed algorithm, the performance of encoder will drop, especially when the video is transmitted under small bandwidth.

3.4 MB Layer Bit Allocation and QP Adjustment

As illustrated in Fig. 3 (a), the original calculation of target bits \(mb \cdot T_{ij}\) for current \(i^{th}\) MB (\(i: 1 \sim N_{\text{mb}}\), where \(N_{\text{mb}}\) is the total number of MB in one frame) in \(j^{th}\) frame is defined by (16) in JVT-G012 [7]. It utilizes the proportion of current MB predicted complexity to total predicted complexity of all remaining MBs in current frame.
The \( mb_{\text{MADP}}p_{i,j} \) and \( mb_{\text{MADAA}}l_{i,j} \) are predicted MAD and actual MAD values of \( j^{th} \) MB in \( j^{th} \) frame, respectively. \( c_{i,j} \) and \( d_{i,j} \) are MB layer parameters in linear prediction model [7], which will be updated after coding each MB. However, the complexity prediction for remaining MBs is not accurate enough since it only utilizes their co-located MBs in previous frame.

The proposed method is shown in Fig. 3 (b). The information from all MBs in previous frame and all pre-coded MBs in current frame are utilized. First, by using pre-coded MBs in current frame and their co-located ones, ratio \( \delta \) is computed by (17). It is defined as a measure of complexity change from last frame ((\( j - 1 \))th) to current one (\( j^{th} \)). In second step, the remaining MBs’ complexity, which is represented by \( mb_{\text{MATD}}p_{i,j} \), is predicted by (18) using their co-located ones and \( \delta_{i,j} \). Here the \( mb_{\text{MATD}}p_{i,j} \) is the predicted sum of remaining MBs’ MATD in \( j^{th} \) frame. Meanwhile, the \( mb_{\text{MATDA}}p_{i,j} \) is also obtained. The linear model in [7] is utilized in the calculation expressed by (18). Finally, target bits \( mb_{T_{i,j}} \) is calculated in (19) by new estimation of complexity proportion.

By subtracting the header bits, the texture bits \( mb_{B_{i,j}} \) for encoding is further computed by (20) [15]. To avoid it to be too small, a lower bound is made by (21).

\[
\begin{align*}
mb_{T_{i,j}} &= T_j \times \frac{mb_{\text{MATDP}}^2_{i,j}}{\sum_{l=0}^{N_{mb}}(c_{i,j} \times mb_{\text{MATDA}}l_{i,j-1} + d_{i,j})^2} \\
\end{align*}
\]

\[
mb_{B_{i,j}} = mb_{T_{i,j}} - mb_{H_{i,j}}
\]

\[
mb_{B_{i,j}} = \max \left\{ mb_{B_{i,j}}, \frac{u}{F_r \times N_{mb} \times \text{MINVALUE}} \right\}
\]

Here, the \( mb_{H_{i,j}} \) is the average number of header bits generated by coded MBs in current \( j^{th} \) frame, \( \text{MINVALUE} \) is a constant with typical value of 4. Although (21) makes \( mb_{B_{i,j}} \) be enough for coding, it may also make it beyond the real bits requirement. When \( mb_{B_{i,j}} \) is smaller than the lower bound, and the actual complexity is not so high, then the computed \( QP_{i,j} \) does not lower satisfy the coding under bit constraints, and the encoder will generate overfull bits. It may lead to lack of bits for remaining coding units. So we do the adjustment based on (22) to deal with this case. \( QP_{i,j} \) is increased slightly to compensate the latent excessive usage of bit resource. The value of \( \Delta QP_{i,j} \) depends on the difference between \( mb_{B_{i,j}} \) and corresponding lower bound. In some cases, \( mb_{B_{i,j}} \) is smaller than zero and far from lower bound. Then the \( \Delta QP_{i,j} \) will set to be larger in order to control the bit generation. The \( \Delta QP_{i,j} \) is defined by Eq. (23) and (24), where LB represents the value of lower bound under current bandwidth and frame rate, \( \theta \) is the step value used to judge the gap between \( mb_{B_{i,j}} \) and LB. Bandwidth \( u \), frame rate \( F_r \), and frame size determine the value of \( \theta \) together. In our experiment, \( \theta \) is set to half of the lower bound LB.

\[
QP_{i,j} = QP_{i,j} + \Delta QP_{i,j}
\]

\[
\Delta QP_{i,j} = \begin{cases} 
2, & mb_{B_{i,j}} < LB - 5\theta \\
1, & LB - 5\theta \leq mb_{B_{i,j}} < LB - \theta \\
0, & mb_{B_{i,j}} \geq LB - \theta 
\end{cases}
\]

\[
\theta = \frac{LB}{2} = \frac{1}{2} \times \frac{u}{F_r \times N_{mb} \times \text{MINVALUE}}
\]

4. Overall Multi-Stage Rate Control Scheme

In this section, the whole proposed R-D optimized multi-stage rate control under CBR transmission is given and explained. All the aforementioned methods in Sect. 2 and Sect. 3 are combined for H.264/AVC rate control. Figure 4 shows the flowchart of the proposed scheme. The processes described in this paper are labelled with bold font. On the GOP layer, the total remaining bits will be computed and the buffer will be updated for upcoming frame. On the frame layer, MAD is replaced by \( MATD_{f} \) to represent the frame coding complexity. The complexity analysis is added to enhance the accuracy of bit allocation. For better mode decision, adaptive Lagrange multiplier \( \Lambda_{\text{MODE}} \) adjustment is added based on previous used bits and remaining bits analysis. It will control the actual bit generation also on the
frame layer. When MB layer rate control is enabled, in our scheme, a method considering complexity movement between current and previous frames is utilized to compute MB target bits. Note that the MB layer residual complexity prediction is also executed with the frequency-domain parameter \(MATD_{mb}\). The computed \(QP\) for current MB is finally adjusted according to the value of target texture bits and its difference from corresponding lower bound determined by bandwidth, frame rate, and the frame size in MB. After encoding each MBs, all the MB layer model parameters are updated. Finally, after finishing encoding all of the MBs in current frame, all frame layer model parameters are updated for next frame.

### 5. Simulation Results & Discussion

The proposed rate control scheme is tested by the recent H.264/AVC reference software JM13.2 [9]. The widely utilized recommended algorithm in JVT-G012 [7] is selected as a benchmark. Numerous experiments have been conducted to evaluate the performance for both QCIF and CIF sequences with various features in 4:2:0 format (\(M \& D\) is the abbreviation of mother and daughter). All sequences are coded with H.264/AVC baseline. With the frame rate 30 fps, 300 frames are coded for each sequence with the GOP structure of IPPP mode (first frame is intra coded \(I\) frame, and the remaining frames are \(P\) frames). CA VLC is chosen as symbol mode. The search range is 16 for QCIF and 32 for CIF, respectively. All other parameters are carefully selected for both [7] and proposed algorithm to be equivalent.

In order to show the bit generation control, we define the evaluating factor \(Rate\_Mismatch\) by (25) [20], [21]. \(AR\) and \(TR\) denote the actual bit-rate after coding and the target bit-rate, respectively.

### Table 3 Performance comparison of QCIF for proposed algorithm with JVT-G012[7] in JM13.2[9].

| Sequence  | Target(Kbps) | Average PSNR (dB) | Bit-Rate(Kbps) | Rate Mismatch |
|-----------|--------------|-------------------|----------------|---------------|
|           | JM13.2      | Proposed | Gain | JM13.2   | Proposed     | JM13.2     | Proposed     |
| salesman  | 48          | 34.77    | 35.58  | +0.81   | 48.05  | 48.04  | 0.11%  | 0.07%        |
| carphone  | 48          | 31.17    | 31.58  | +0.41   | 48.07  | 48.05  | 0.14%  | 0.10%        |
| news      | 36          | 31.88    | 32.76  | +0.88   | 36.12  | 35.86  | -0.34% | -0.40%       |
| M & D     | 48          | 34.89    | 35.51  | +0.62   | 48.10  | 48.05  | 0.21%  | 0.10%        |
| foreman   | 48          | 30.46    | 30.67  | +0.21   | 48.08  | 48.04  | 0.17%  | 0.08%        |
| claire    | 24          | 37.60    | 37.76  | +0.13   | 24.06  | 23.99  | 0.24%  | -0.04%       |
| grandma   | 64          | 38.06    | 39.00  | +0.94   | 64.09  | 64.07  | 0.15%  | 0.11%        |
| panzoom   | 48          | 34.17    | 34.76  | +0.59   | 48.08  | 48.03  | 0.17%  | 0.03%        |
| akiyo     | 24          | 35.97    | 37.72  | +1.13   | 24.06  | 23.99  | 0.24%  | -0.04%       |
| headwithglass | 24 | 29.57    | 29.97  | +0.40   | 24.13  | 24.05  | 0.53%  | 0.20%        |

### Table 4 Performance comparison of CIF for proposed algorithm with JVT-G012[7] in JM13.2[9].

| Sequence  | Target(Kbps) | Average PSNR (dB) | Bit-Rate(Kbps) | Rate Mismatch |
|-----------|--------------|-------------------|----------------|---------------|
|           | JM13.2      | Proposed | Gain | JM13.2   | Proposed     | JM13.2     | Proposed     |
| container | 192         | 34.59    | 35.36  | +0.77   | 192.06 | 192.08 | 0.03%  | 0.04%        |
| news      | 96          | 33.45    | 34.28  | +0.83   | 96.27  | 96.10  | 0.28%  | 0.11%        |
| akiyo     | 64          | 37.48    | 38.44  | +0.96   | 64.09  | 64.07  | 0.14%  | 0.10%        |
| hall_monitor | 128   | 35.39    | 35.79  | +0.40   | 128.15 | 128.06 | 0.12%  | 0.05%        |
| M & D     | 96          | 36.99    | 37.34  | +0.35   | 96.17  | 96.08  | 0.18%  | 0.08%        |
| panzoom   | 96          | 34.10    | 34.63  | +0.53   | 96.20  | 95.45  | 0.21%  | -0.58%       |
| table_jennis | 192     | 32.20    | 32.67  | +0.47   | 192.12 | 192.00 | 0.06%  | 0.00%        |
| paris     | 320         | 32.75    | 33.46  | +0.71   | 320.82 | 320.30 | 0.26%  | 0.09%        |
| highway   | 128         | 36.10    | 36.62  | +0.52   | 129.07 | 128.19 | 0.83%  | 0.15%        |

Fig. 4 Flowchart of proposed RDO based rate control scheme.
Fig. 5  PSNR comparison and Buffer status comparison of QCIF and CIF sequences.
Fig. 6  R-D curves comparison of QCIF and CIF sequences.
Reconstructed frame comparisons of QCIF and CIF sequences, with JVT-G012 [7] in JM13.2 [9] (left) and with our proposed algorithm (right).

Fig. 7 Reconstructed frame comparisons of QCIF and CIF sequences, with JVT-G012 [7] in JM13.2 [9] (left) and with our proposed algorithm (right).

\[ \text{Rate_Mismatch} = \frac{AR - TR}{TR} \times 100\% \]  \hspace{1cm} (25)

Table 3 and Table 4 show the simulation results for QCIF and CIF sequences, respectively. It can be found that our scheme outperforms JVT-G012 [7] on both coding quality (represented by peak-signal-noise-ratio (PSNR)) and bitrate control under various bandwidths. A gain up to 1.13 dB is observed for QCIF sequence while the improvement for CIF sequence is up to 0.96 dB. The rate mismatch is also reduced. Look into “news.qcif”, when transmitted under 36 Kbps, the PSNR improvement is 0.88 dB while the bit mismatch is reduced from 0.34\% to −0.40\%, which means the encoder achieves much higher coding quality and larger bit-rate saving synchronously.

Figure 5 shows the examples of PSNR and buffer status comparisons for each frame in one sequence. For most of the frames, PSNR is improved by using proposed scheme. The reason is our scheme has better solution to both frame and MB layer rate control, especially the latter one. The enhanced MB layer complexity analysis makes bit usage much more effective within each single frame. The frame layer rate control makes system level bit allocation much more reasonable. It saves bits from uncomplicated frames, and spends saved bits on frames with high residual complexity. It can be noticed that at the beginning of each sequence, the quality is almost remain the same or even has slight decrease. It is because that the conventional algorithm has less control on bit usage and always spends many bits to encode the early part of sequence (generally the “earlier part” contains from several frames to one third or half of the whole test sequence). The “earlier part” may have good quality but the rest of sequence suffers a lot due to the lack of bit budget. For this reason, the improvement for each frame will become larger during the progress of encoding (take “table_tennis.cif” for example).

By giving the “Bufferfullness” information after encoding each frame, it is clear that the proposed algorithm makes buffer become easier to control. The Bufferfullness is much closer to or lower than the target level so that the risk of buffer overflow will reduced and fewer frames will be skipped in real video codec, especially for the “early part” of the sequence. This benefits the quality a lot when the buffer size or transmission bandwidth is strictly limited in video applications. In the case of scene change (take
“table_fennis.cif” for example), the Bufferfullness can be quickly decreased to a low level.

The examples of R-D curves for QCIF and CIF sequences are given in Fig. 6. Both QCIF and CIF sequences obtain optimized R-D performance, which means under a given bandwidth encoder achieves higher coding quality. Moreover, better performances are also reflected in the improvement of subjective quality of reconstructed frames, as shown in Fig. 7. It is obvious that the visual distortion is reduced and more detail information is maintained by using our scheme.

6. Conclusion

This paper proposes a multi-stage rate control algorithm for R-D optimized H.264/AVC encoders under CBR video transmission. First, the MATD is adopted to represent the coding complexity for both frame and MB instead of originally utilized MAD. On the frame layer, the bit allocation is modified by current frame complexity estimation. The MATD ratio which represents current frame complexity proportion in all pre-coded frames is included in this process. For choosing a suitable mode in mode decision part, the Lagrange multiplier \( \lambda_{MODE} \) is adaptively adjusted also on the frame layer in our rate control scheme. On each frame, a factor combined both backward and forward bit usage measurements is utilized to adjust the \( \lambda_{MODE} \). The bit usage and frame complexity are considered in a universal point of view. So that better modes will be determined to generate much more reasonable bits throughout the entire sequence. On the MB layer, bit allocation is improved by accurate complexity analysis. All of the actual MATD values of both previous and current frame are fully utilized. Further, the computed QP will be adjusted according to the comparison of MB layer predicted texture bits and the lower bound defined by [9]. These techniques are adopted in several key stages of rate control covering both frame and MB layers. Simulation results show that the encoder gains up to 1.13 dB in PSNR by using our algorithm compared with the recommended rate control scheme which is implemented in the H.264/AVC reference software JM13.2 [9]. The target bit-rate control and subjective quality are also improved.

Acknowledgments

This work was supported by fund from Core Research for Evolutional Science and Technology (CREST), JST.

References

[1] T. Wiegand, G.J. Sullivan, and A. Luthra, “Draft ITU-T recommendation H.264 and final draft international standard of joint video specification,” (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), Doc. JVT-G050r1., Geneva, Switzerland, May 2003.
[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Widi, “Video coding with H.264/AVC: Tools, performance and complexity,” IEEE Trans. Circuits Syst. Mag., vol.4, no.1, pp.7–28, First Quarter 2004.
[3] 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, July 2003.
[4] Test Model 5, ISO/IEC JTC1/SC29/WGI1, 1993.
[5] Video Codec Test Model, ITU-T/S215 TMN 8, Portland, OR, June 1997.
[6] MPEG-4 video verification model V18.0, Coding of Moving Pictures and Audio N3908, ISO/IEC JTC1/SC29/WG11. Jan. 2001.
[7] Z.G. Li, F. Pan, K.P. Lim, G. Feng, X. Lin, and S. Rahardja, “Adaptive basic unit layer rate control for JVT,” JVT-G012, 7th Meeting, Pattaya, Thailand, March 2003.
[8] Z.G. Li, C. Zhu, N. Ling, X.K. Yang, G.N. Feng, S. Wu, and F. Pan, “A unified architecture for real-time video-coding systems,” IEEE Trans. Circuit Syst. Video Technol., vol.13, no.6, pp.472–487, June 2003.
[9] JVT, “H.264/AVC reference software (JM13.2),” http://www.hhi.fraunhofer.de/en/departments/image-processing.html.
[10] M. Jiang and N. Ling, “On enhancing H.264/AVC video rate control by PSNR-based frame complexity estimation,” IEEE Trans. Consum. Electron., vol.51, no.1, pp.281–286, Feb. 2005.
[11] M. Jiang and N. Ling, “On lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control,” IEEE Trans. Circuits Syst. Video Technol., vol.16, no.5, pp.663–669, May 2006.
[12] H.J. Lee, T.H. Chiang, and Y.Q. Zhang, “Scalable rate control for MPEG-4 video,” IEEE Trans. Circuits Syst. Video Technol., vol.10, no.6, pp.878–894, Sept. 2000.
[13] X. Jing and L.P. Chau, “Improved frame level MAD prediction and bit allocation scheme for H.264/AVC rate control,” Proc. 2007 IEEE International Symposium on Circuits and Systems (ISCAS 2007), pp.3639–3642, New Orleans, USA, May 2007.
[14] N. Kamaci, Y. Altunbasak, and R.M. Merereau, “Frame bit allocation for the H.264/AVC video coder cauchy-density-based rate and distortion models,” IEEE Trans. Circuits Syst. Video Technol., vol.15, no.8, pp.994–1006, Aug. 2005.
[15] M. Jiang and N. Ling, “An improved frame and macroblock layer bit allocation scheme for H.264 rate control,” Proc. 2005 IEEE International Symposium on Circuits and Systems (ISCAS 2005), pp.1501–1504, Kobe, Japan, May 2005.
[16] W. Yuan, S. Lin, Y. Zhang, W. Yuan, and H. Luo, “Optimum bit allocation and rate control for H.264/AVC,” IEEE Trans. Circuits Syst. Video Technol., vol.16, no.6, pp.705–715, June 2006.
[17] G.J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” IEEE Signal Process. Mag., vol.15, no.6, pp.74–90, Nov. 1998.
[18] K.P. Lim, G.J. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods,” JVT-O079, Busan, Korea, April 2005.
[19] C. Kim, H.-H. Shih, and C.-C.J. Kuo, “Feature-based intra-prediction mode decision for H.264,” Proc. IEEE Int. Conf. Image Process. (ICIP 2004), pp.769–772, Oct. 2004.
[20] M. Jiang and N. Ling, “Low-delay rate control for real-time H.264/AVC video coding,” IEEE Trans. Multimed., vol.8, no.3, pp.467–477, June 2006.
[21] X. Li, N. Oertel, A. Hutter, and A. Kaup, “Rate-distortion optimized frame level rate control for H.264/AVC,” Proc. 2008 European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, Aug 2008.
Shuijiong Wu was born in 1985. He received the B.E. degree in Information Engineering from Shanghai Jiao Tong University (SJTU), China, in 2008, and the M.E. degree in Graduate School of Information, Production and Systems (IPS), Waseda University, Japan, in 2009. He is currently a master candidate in Department of Electronic Engineering in SJTU. His research interest includes video compression algorithm and associated very large scale integration (VLSI) architecture.

Peilin Liu is a PhD Graduate (1992~1998) from the University of Tokyo majoring in Electronic Engineering and worked there as a researcher in 1999. After that, she was hired as a senior researcher for Central Research Institute of Fujitsu, Tokyo (1999~2003). She is now a professor of Department of Electronic Engineering in Shanghai Jiao Tong University, China. Her research mainly focuses on multimedia (audio/video) processing, IC design and high-performance processor development.

Yiqing Huang was born in 1982. He received the B.E. and M.E. degrees in communication and information engineering from Shanghai University, China in 2004 and 2007 respectively. In 2007, he received the M.E. degree in LSI System from Waseda University, Japan. He is currently a Ph.D. candidate in Graduate School of Information, Production and Systems, Waseda University, Japan. His research interest includes hardware friendly video coding algorithm and associated very large scale integration (VLSI) architecture.

Qin Liu was born in Nanjing, China. He received the M.S. degree in software engineering from Nanjing University (NJU), China, in 2006, and the Ph.D. degree for Low-Complexity Video Compression Algorithm and Video Encoder LSI Design from Waseda University, Japan, in 2009. He joined software institute of Nanjing University since Oct. 2009. His research interest includes video compression algorithm and its VLSI hardware architecture design.

Takeshi Ikenaga received his B.E. and M.E. degrees in electrical engineering and Ph.D degree in information & computer science from Waseda University, Tokyo, Japan, in 1988, 1990, and 2002, respectively. He joined LSI Laboratories, Nippon Telegraph and Telephone Corporation (NTT) in 1990, where he had been undertaking research on the design and test methodologies for high performance ASICs, a real-time MPEG2 encoder chip set, and a highly parallel LSI & system design for image-understanding processing. He is presently an associate professor in the system LSI field of the Graduate School of Information, Production and Systems, Waseda University. His current interests are application SoCs for image, security and network processing. Especially, he engages in the research on H.264 encoder LSI, JPEG2000 codec LSI, LDPC decoder LSI, UWB wireless communication LSI, public key encryption LSI, object recognition LSI, etc. Dr. Ikenaga is a member of IEEE, IPSJ and IIEEJ.