On Lagrange Multiplier and Quantizer Adjustment for H.264 Frame-Layer Video Rate Control

Minqiang Jiang and Nam Ling, *Senior Member, IEEE*

***Abstract*—H.264/AVC encoder employs a complex mode-decision technique based on rate-distortion optimization. It calculates rate-distortion cost (RDcost) for all possible modes to choose the best one having the minimum RDcost. This paper presents a frame-layer rate control for H.264/AVC that computes the Lagrange multiplier (λ_{MODE}) for mode decision by using a quantization parameter (QP) which may be different from that used for encoding. At the same time, we also compare actual bits produced by previous macroblocks (MBs) with the total bits allocated to these MBs to further modify λ_{MODE}. The objective of these measures aims to produce bits as close to the frame target bits for rate control as possible. This is very important in the case of low-bit-rate tight buffer applications. In order to obtain an accurate QP for a frame, we employ a complexity-based bit-allocation scheme and a QP adjustment method. Simulation results comparing with the H.264 Joint Video Team (JVT) rate control method show that the H.264 encoder, using the proposed algorithm, achieves a visual quality improvement of about 0.56 dB, performs better for buffer overflow and underflow, and achieves a smaller PSNR deviation.**

***Index Terms*—AVC, bit allocation, H.264, Joint Video Team (JVT), Lagrange multiplier, rate-distortion optimization (RDO), rate control, video coding.**

I. Introduction

THE H.264/AVC video coding standard [1] achieves much higher coding efficiency than that of the previous standards such as MPEG-2, H.263, H.263+, H.263++, and MPEG-4. This is mainly due to the fact that the H.264 encoder employs more complicated approaches in the coding procedure. One important approach is variable-size block motion estimation (ME) and mode decision. In order to improve prediction accuracy, H.264 uses tree-structured hierarchical macroblock (MB) partitions. For example, an intercoded 16×16 pixel MB can be broken into MB partitions of sizes 16×8, 8×16, or 8×8. The 8×8 partitions can be further broken into sub-MB partitions of sizes 8×4, 4×8, and 4×4. When conducting rate-constrained mode decision for an MB in an interframe, besides the multiple intermodes, H.264 also supports skip mode and intramodes. An intramode has two block types: 4×4 and 16×16. Rate-distortion-optimized motion estimation is first performed for each intermode, and, then, given these motion vectors, the overall rate-distortion cost (RDcost) for each intermode and intramode is computed [3]. The mode with the minimal cost is selected as the best mode. The mode decision is made by minimizing

$$J_{\mathrm{MODE}}(s, c, \mathrm{MODE} \mid \lambda_{\mathrm{MODE}}) = \mathrm{SSD}(s, c, \mathrm{MODE}) + \lambda_{\mathrm{MODE}} \cdot R(s, c, \mathrm{MODE}). \quad (1)$$

In the above equation, SSD denotes the sum of square differences between the original block s and its reconstruction c, MODE indicates a mode out of a set of potential MB modes: $\{\mathrm{SKIP}, 16\times 16, 16\times 8, 8\times 16, 8\times 8, 8\times 4, 4\times 8, 4\times 4, \mathrm{Intra}4\times 4, \mathrm{Intra}16 \times 16\}$, $R(s, c, \mathrm{MODE})$ is the number of bits associated with choosing MODE, including bits for the MB header, the motion, and all integer transform/quantization coefficients, and λ_{MODE} is the Lagrange multiplier, which is given by [3]

$$\lambda_{\mathrm{MODE}} = 0.85 \times 2^{(\mathrm{QP}-12)/3} \quad (2)$$

where QP is the MB quantization parameter. In this method, a QP should first be given for each MB, and then the optimal MB mode can be decided. Note that the total number of bits for every frame in a video sequence is not controlled in the above argument. When frame-layer rate control is enabled, the number of bits actually used for a frame is controlled by frame QP only because each MB uses the same frame QP for encoding and hence the same Lagrange multiplier was used for performing rate-distortion optimization (RDO). No sound mechanism exists to minimize the mismatch between the number of bits actually used for a frame and its target bits allocated, except for the frame QP generated. This paper proposes a scheme to control the mode decision of each MB using an adaptive Lagrange multiplier to meet the rate allocation for each frame without changing the frame QP. Meanwhile, frame skipping can be improved with a smaller mismatch in low-bit-rate applications. However, it must be pointed out that the most important factor affecting the performance is the frame QP. In general, a frame-layer rate-control scheme first allocates a target number of bits to each frame and then computes the QP to meet the frame target allocation. To achieve accurate actual bits for each frame, we first generate a good estimation of frame target bits, with the bit allocation method to obtain a QP as described in the next paragraphs. In order to ensure that we minimize the mismatch between actual bits generated and the target bits, especially in low-bit-rate applications, this paper proposes an adaptive Lagrange multiplier scheme and a QP adjustment method.

For frame bit allocation, the problem is how to compute a bit budget for each frame. Current studies can roughly be classified into two categories: one is predominantly based on buffer status and the other uses a scene-content complexity-based approach. Many rate control algorithms, such as TMN8 [11], linear rate control algorithm [6], MPEG-4 Q2 [8], [13], [14], and adaptive rate control for the Joint Video Team (JVT) [2], belong to the first category. For example, in [6] and [11], the target number of bits for each frame is determined by $u/F_r - \Delta$, where u denotes the channel bandwidth given to the sequence in bits per second (bps), F_r denotes the predefined frame rate in frames per second (fps) (hence, u/F_r is bits/frame flowing out), and Δ is a small value that provides feedback from the buffer status. In [13] and

Manuscript received September 23, 2004; revised May 12, 2005. This paper was recommended by Associate Editor H. Sun.

The authors are with the Department of Computer Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: mjiang@scu.edu; nling@scu.edu).

Digital Object Identifier 10.1109/TCSVT.2006.873159

[14], the initial target bit number for each new P-frame is scaled based on the current buffer level to maintain a buffer occupancy of about 50% of the buffer size after encoding each frame. Nevertheless, since the scene-content complexity of a video sequence changes along time, the quality of encoded video therefore varies dramatically if only buffer-based rate control is used. It is clear that such bit allocation schemes are unable to achieve optimal performance because they do not match the nonstationary characteristics of video signals. To achieve high and consistent quality of encoded video, some approaches allocate bits among video frames by a multipass encoding strategy [15], [16], but they are more appropriate for offline encoding due to the time-consuming multipass procedure. Other approaches perform target bit allocation by considering frame-content complexity [4], [5], [12]. Frames with higher complexity can have more bits, and frames with lower complexity may have fewer bits. For example, Ribas-Corbera and Lei [12] proposed an optimized method to assign target bits to each frame according to frame energy, i.e., the variance of residue. Our method in this paper is based on frame bit allocation considering both buffer status and frame complexity [4], [5]. We estimate frame complexity by using the relative mean absolute difference (MAD) statistics of the current frame versus that of the previously encoded frames.

Once the target number of bits for a frame is determined, the next step is to determine the frame QP to meet the target bit allocation. The solution is heavily dependent upon the rate-distortion (R-D) models used. The MPEG-4 Q2 rate control scheme is one of the most popular schemes based on a quadratic rate-quantization (R-Q) model [17]. Recently, Li *et al.* proposed an adaptive rate-control scheme [2], [7] that forms the basis of the H.264 rate control recommendations [10], where this quadratic R-Q model was also adopted to compute the corresponding QP. Since any model is actually an approximate model and cannot always match ideally with real application environment, the estimation of model parameters usually has some deviations or errors and hence does the QP obtained by frame rate control. We therefore propose a QP adjustment scheme in this paper, including a method by monitoring the cumulative difference between the actual bits and the target bits, in order to obtain a QP as accurately as possible.

This paper is organized as follows. Section II gives the basic principle of our adaptive Lagrange multiplier method used in our proposed rate control. Details of our proposed frame-layer bit allocation and QP adjustment schemes are described in Section III. Simulation results are given in Section IV, followed by the conclusions in Section V.

II. Adaptive Lagrange Multiplier

For simplicity, we first simplify (1) as

$$J = D + \lambda_{\mathrm{MODE}} \cdot R. \tag{3}$$

For an MB, assuming modes 1 and 2 with RDcost $J_1 = D_1 + \lambda_{\mathrm{MODE}} \cdot R_1$ and $J_2 = D_2 + \lambda_{\mathrm{MODE}} \cdot R_2$, respectively, where $D_1 < D_2, R_1 > R_2$ and $J_1 < J_2$, we can easily obtain

$$\lambda_{\mathrm{MODE}} < \frac{D_2 - D_1}{R_1 - R_2}. \tag{4}$$

If we increase λ_{MODE}, then (4) may no longer be satisfied so that the better mode changes from mode 1 to mode 2 (i.e., $J_2 < J_1$), i.e., a larger λ_{MODE} corresponds to higher D and lower R; the converse is true for lower λ_{MODE}. This implies that changes in the λ_{MODE} influence the rate and distortion of the resulting coded pictures. This is the basis for our proposed adaptive Lagrange multiplier method. Computation of the adaptive Lagrange multiplier includes two steps as discussed in the following subsections.

A. Using Computed Quantization Parameter QP for Computing λ_{MODE}

We observed that many existing rate control schemes, including that of JVT [2], determine the final frame QP by clipping the computed QP (denoted as $\mathrm{QP}_{\mathrm{computed}}$) obtained from the frame-layer rate control calculation, to no more than $\pm\Delta\mathrm{QP}$ of the previous frame's (which is denoted as $\mathrm{QP}_{\mathrm{pf}}$), to maintain the smoothness of visual quality among successive frames. The current frame QP is given by

$$\mathrm{QP}_{\mathrm{cf}} = \min\{\mathrm{QP}_{\mathrm{pf}} + \Delta\mathrm{QP}, \max\{\mathrm{QP}_{\mathrm{pf}} - \Delta\mathrm{QP}, \mathrm{QP}_{\mathrm{computed}}\}\} \tag{5}$$

where $\mathrm{QP}_{\mathrm{cf}}$ denotes the QP of the current frame for encoding and $\Delta\mathrm{QP}$ has a typical value of 2 or 3. Considering λ_{MODE} in (2), increasing QP by 1, λ_{MODE} will increase by $2^{1/3} = 1.26$ times, increasing QP by 2, λ_{MODE} will increase by 1.59 times, and so on. If the computed QP is larger than $\mathrm{QP}_{\mathrm{pf}} + \Delta\mathrm{QP}$, the $\mathrm{QP}_{\mathrm{cf}}$ used for encoding is then $\mathrm{QP}_{\mathrm{pf}} + \Delta\mathrm{QP}$, which is smaller than $\mathrm{QP}_{\mathrm{computed}}$. In such a case, the actual total number of bits produced for the frame is very likely to be larger than the frame's target. The worst case is that overflow occurs after encoding the current frame especially, in a low-bit-rate situation. Even if overflow does not occur, buffer occupancy may be at a high level, which will cause the next P-frame to be allocated with fewer target bits [cf. (15) in Section III] and result in visual quality degradation. For such a case, we propose a larger Lagrange multiplier λ_{MODE} in mode decision to favor a mode which produces fewer bits. On the contrary, a smaller computed QP may be clipped to $\mathrm{QP}_{\mathrm{pf}} - \Delta\mathrm{QP}$ as $\mathrm{QP}_{\mathrm{cf}}$ for encoding and RDO. In this case, the actual total number of bits produced for the frame is likely to be less than the frame's target, and underflow may even occur. We therefore propose to use $\mathrm{QP}_{\mathrm{computed}}$ instead of $\mathrm{QP}_{\mathrm{cf}}$ for computing λ_{MODE} in (2). Now λ_{MODE} is adjusted to be

$$\lambda_{\mathrm{MODE}} = 0.85 \times 2^{(\mathrm{QP}_{\mathrm{computed}} - 12)/3}. \tag{6}$$

B. Further Adjusting λ_{MODE} by Considering Actual Encoding Bits

Since the encoder is supposed to produce total bits by all MBs in a frame as close to the frame target bit allocation (denoted as T_f) as possible, it is reasonable to further adjust λ_{MODE} based on actual encoding results at that point. We propose to further modify λ_{MODE} for MB i, denoted as λ_i and given as

$$\lambda_i = 0.85 \times \alpha_i \times 2^{(\mathrm{QP}_{\mathrm{computed}} - 12)/3} \tag{7}$$

TABLE I
PERFORMANCE COMPARISON (AVERAGE MISMATCH RATIO, FRAME SKIP, AND UNDERFLOW) FOR OUR PROPOSED ALGORITHM WITH [2]

Sequence	Frame Rate (fps)	Average mismatch ratio (%)		No. of skipped frames		No. of underflows	
		[2]	Our Adaptive λ	[2]	Our Adaptive λ	[2]	Our Adaptive λ
Carphone QCIF (24 kbps)	10	19.26	14.43	6	4	3	0
Carphone CIF (32kbps)	10	25.89	17.16	22	15	0	0
Foreman QCIF (24 kbps)	10	25.35	21.11	16	7	0	0
Foreman CIF (32 kbps)	10	11.50	3.56	12	5	5	2
Stephan QCIF (128 kbps)	10	13.03	10.68	0	0	0	0
Akiyo QCIF (48 kbps)	30	14.84	3.95	0	0	0	0
Highway CIF (24kbps)	10	18.88	12.02	6	4	0	0
Salesman QCIF (64 kbps)	30	27.40	19.33	0	0	0	0

where α_i is computed on a macroblock-basis using the following formulas:

$$\alpha_i = B_{i-1}/T_{i-1}, \quad \text{for } i > 1 \tag{8}$$

$$B_{i-1} = \sum_{l=1}^{i-1} B_{\mathrm{MB},l} \tag{9}$$

$$T_{i-1} = \sum_{l=1}^{i-1} T_{\mathrm{MB},l} \tag{10}$$

$$T_{\mathrm{MB},l} = \frac{\mathrm{MAD}_{\mathrm{MB},l}}{\mathrm{MAD}_{\mathrm{frame}}} \times T_f \tag{11}$$

where $B_{\mathrm{MB},l}$ is the actual bits and $T_{\mathrm{MB},l}$ is the target bit of the lth MB in the frame, $\mathrm{MAD}_{\mathrm{MB},l}$ is the MAD of the lth MB, and $\mathrm{MAD}_{\mathrm{frame}}$ is the average MAD of the current frame. B_{i-1} denotes the cumulative bits produced thus far, and T_{i-1} denotes the total target bits for the previous $i-1$ macroblocks. When α_i becomes greater than 1.0, we know that we have generated too many bits so far. We need to increase λ_i so as to reduce the number of bits generated. When α_i becomes less than 1.0, we have encoded too few bits and need to lower λ_i so as to generate more bits. Note that α_i naturally reflects the need for an adaptive λ_{MODE} used in RDO for the ith MB. We therefore propose to use α_i as a multiplicative factor in (7) and hence an adaptive λ_{MODE}.

In order to show improvement of the mismatch ratio between the target bits and actual achieved bits per frame, we define

$$\mathrm{mismatch\_ratio}(j) = |((A_j - T_j)/T_j) * 100| \tag{12}$$

where A_j is the actual number of bits of frame j and T_j is the number of target bits T_f of frame j. The buffer size is set to be $0.5 \times u$ (u is channel bandwidth) [14], to show improvement of frame skipping in low-rate applications. Experimental conditions are set as in Section IV. Effects of our adaptive λ on mismatch and frame skip are reported in Table I. As expected, Table I shows that using the adaptive Lagarange multiplier λ for RDO can effectively reduce both mismatches and the number of frames skipped, especially for high motion sequences.

III. OUR FRAME-LAYER RATE CONTROL

A. Frame-Layer Bit Allocation

Without the loss of generality, we assume that the first frame in a group of pictures (GOP) is an intra-coded I-frame and the remaining frames are all predicted P-type frames. The frame target bits T_f for the current frame are a weighted combination of T_r and T_{buf} and are given by

$$T_f = \beta * T_r + (1-\beta) * T_{\mathrm{buf}} \tag{13}$$

where β is a weighting factor, and its typical value is 0.70. T_{buf} and T_r represent the frame target bits estimated from buffer status and those from the bits remaining for encoding the sequence factoring into the consideration of frame complexity, respectively, and will be defined later. The target buffer level was first defined in [2] as

$$B_t = B_t' - \frac{\left(B_c^1 - B_s/8\right)}{N_p - 1} \tag{14}$$

where B_c^1 is the buffer fullness after the first P-frame is coded (the QP of the first P-frame is predefined as that of the I-frame), B_s is the buffer size, N_p is the total number of P-frames in a GOP, B_t is the target buffer level of the current P-frame, and B_t' is the target buffer level of the previous P-frame. The initial buffer fullness of a GOP is set as $B_s/8$. T_{buf} of the current frame in (13) is defined by [7]

$$T_{\mathrm{buf}} = (u/F_r) - \gamma \cdot (B_c - B_t) \tag{15}$$

taking into account the actual buffer occupancy B_c and the target buffer level B_t before encoding the current frame where u is the channel bandwidth given to the sequence, F_r is the predefined frame rate, hence u/F_r is bits/frame flowing out, and γ is a constant with a typical value of 0.75. We extend our earlier work in [4] and [5] and propose that T_r in (13) be calculated by (16), shown at the bottom of the following page, where R_r is the number of bits remaining for encoding the sequence, and N_r is the number of P-frames remaining to be coded. The thresholds TH_1 and TH_2 came from experiments, whose typical values are set as 1.1 and 2.0, respectively. $\mathrm{MAD_{ratio}}$ in (16) is defined as the frame complexity measure, which is the ratio of the predicted MAD of the current frame (denoted as $\mathrm{MAD_{frame}}$) to the average MAD of all previously encoded P-frames in the GOP [4], [5]. The basic idea is that T_r is set smaller for a frame with lower frame complexity and is set larger for a frame with higher complexity. The objective of this improvement is to save bits from those frames that are less complex and allocate more bits to frames that are more complex. The use of relative measurement of complexity (ratio) and the remaining average bits (R_r/N_r) to calculate T_r [see (16)] enables us to control bit rates close to a constant value.

B. QP Adjustment

From the above subsection, we obtain a frame target bits T_f. Let T_{header} denote the bits used for motion and header information of the previous frame, and $T_{\mathrm{texture}} = T_f - T_{\mathrm{header}}$ denote the number of bits for texture (residual) coding. The quantization parameter QP corresponding to T_{texture} can be computed based on a quadratic R-Q model [14]

$$T_{\mathrm{texture}} = X_2 \frac{\mathrm{MAD}}{\mathrm{QP}^2} + X_1 \frac{\mathrm{MAD}}{\mathrm{QP}} \tag{17}$$

where X_1 and X_2 are the first and second-order coefficients. X_1 and X_2 are updated by the previous coded frames. The detailed algorithm can be found in [14], and it is thus not elaborated on in this paper. To maintain the smoothness of visual quality among successive frames, the computed QP is limited to change within a range (e.g., a maximum of 25% from the previous frame's quantization parameter $\mathrm{QP_{pf}}$, as in [14]). In the H.264 rate-control scheme [2], [9], the QP for encoding the current frame, which is denoted as $\mathrm{QP_{cf}}$, is decided by

$$\mathrm{QP_{cf}} = \min\{\mathrm{QP_{pf}} + \Delta\mathrm{QP1}, \max\{\mathrm{QP_{pf}} - \Delta\mathrm{QP2}, \mathrm{QP_{computed}}\}\} \tag{18}$$

where ΔQP1 and ΔQP2 are constants with the same value of 2. In our experiments, we take the values of 3 and 2 for ΔQP1 and ΔQP2, respectively. This kind of adjustment is, however, not sufficient in some cases. We consider two special cases and propose QP adjustments in the following discussions.

1) T_{texture} Imposed by a Lower Bound: Since T_{texture} may be too small, a lower bound is usually used [2], [9], which is given by

$$T_{\mathrm{texture}} = \max\{T_{\mathrm{texture}}, u/(\mathrm{MINVALUE} \times F_r)\} \tag{19}$$

where MINVALUE is a constant and its typical value is 4. However, the computed $\mathrm{QP_{cf}}$ by using the lower bound is very likely to result in generating more bits than the current frame target bits T. It is necessary to further adjust $\mathrm{QP_{cf}}$ so as to achieve its target bits more accurately. We adjust $\mathrm{QP_{cf}}$ simply by adding 1, i.e.,

$$\mathrm{QP_{cf}} = \mathrm{QP_{cf}} + 1. \tag{20}$$

2) Buffer in the Danger of Overflow or Underflow: When buffer occupancy is higher than 50% of buffer size and if the cumulative difference between the actual bits and target bits becomes larger and larger, this means that the actual bits after encoding cannot be decreased sufficiently to maintain a stable buffer status. The buffer is then in danger of an overflow. Similarly, if the cumulative difference between the actual bits and target bits becomes smaller and smaller (i.e., larger on the negative side) while the buffer level is lower than 30% of the buffer size, the buffer may be in the danger of an underflow. We define

$$\mathrm{AT\_ratio}_i = \begin{cases} A_i/T_i & \text{if } A_i \geq T_i \\ -T_i/A_i & \text{if } A_i < T_i \end{cases} \tag{21}$$

$$H_\Sigma^1 \quad \text{or} \quad H_\Sigma^2 = \sum_{i<n} \mathrm{AT\_ratio}_i \tag{22}$$

where A_i and T_i are the actual bits and the target bits of the ith frame in the past. H_Σ^1 is computed when the current buffer level remains above 50%; otherwise, it is set to be zero. H_Σ^2 is computed when the current buffer level remains below 30%; otherwise, it is set to be zero. The $\mathrm{QP_{cf}}$ for the current frame is then further adjusted by

$$\mathrm{QP_{cf}} = \begin{cases} \mathrm{QP_{cf}} + \Delta\mathrm{QP3}, & \text{if } H_\Sigma^1 > H_{\mathrm{th_1}} \\ \mathrm{QP_{cf}} - \Delta\mathrm{QP3}, & \text{if } H_\Sigma^2 < H_{\mathrm{th_2}} \end{cases} \tag{23}$$

where ΔQP3 is a constant with a typical value of 1, H_{th1} is the threshold of H_Σ^1 with a typical value of 8, and H_{th2} is the threshold of H_Σ^2 with a typical value of -6, in our experiments.

C. Performing R-D Optimization

After we have obtained the frame target bits T_f, $\mathrm{QP_{computed}}$ and the final $\mathrm{QP_{cf}}$, we apply the method in Section II to calculate the adaptive λ_{MODE}. The Lagrange multiplier λ_{MODE} for mode decision is then computed by (7). If the sum of absolute differences (SAD) is adopted as the distortion criterion instead of using SSD, the lambda for motion estimation is given by [3]

$$\lambda_{\mathrm{MOTION}} = \sqrt{\lambda_{\mathrm{MODE}}}. \tag{24}$$

We have thus derived, with the use of α_i and $\mathrm{QP_{computed}}$, a better adaptive λ_{MODE} and $\lambda_{\mathrm{MOTION}}$.

IV. Experimental Results

We have implemented our proposed rate-control scheme on the software of [9] test model software. As a reference for com-

$$T_r = \begin{cases} 0.8 \cdot \mathrm{MAD_{ratio}} \cdot R_r/N_r & \mathrm{MAD_{ratio}} < \mathrm{TH_1} \\ (\mathrm{TH_1} + 0.3 \cdot (\mathrm{MAD_{ratio}} - \mathrm{TH_1})) \cdot R_r/N_r & \mathrm{TH_1} \leq \mathrm{MAD_{ratio}} < \mathrm{TH_2} \\ (\mathrm{TH_1} + 0.3 \cdot (\mathrm{TH_2} - \mathrm{TH_1})) \cdot R_r/N_r & \mathrm{MAD_{ratio}} \geq \mathrm{TH_2} \end{cases} \tag{16}$$

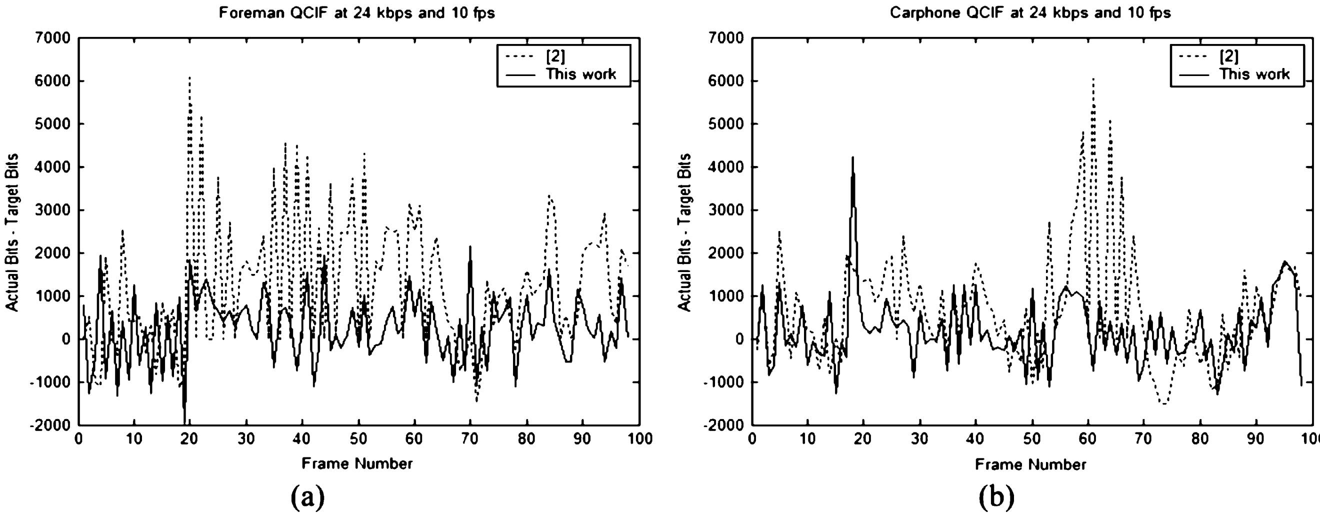

Fig. 1. Difference between the actual bits and target bits per frame for (a) "Foreman" and (b) "Carphone."

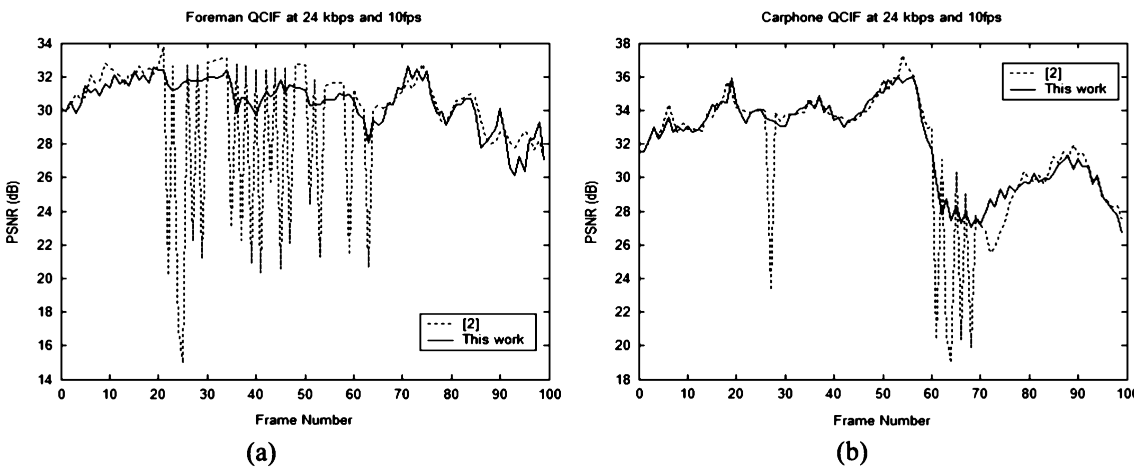

Fig. 2. PSNR per frame for sequences (a) "Foreman" and (b) "Carphone."

parisons, the rate-control algorithm [2] was selected. We also implemented the frame skip scheme described in [2] for both algorithms. When a frame was skipped, the respective previously encoded frame was used in the PSNR computation. Results of eight QCIF and CIF test sequences with various target bit rates and different frame rates are tabulated in this paper. The first frame was an intra-coded I-frame and the remaining frames in a GOP were P-type frames. For all tests, context adaptive binary arithmetic coding (CABAC) and RDO were enabled, and a motion estimation (ME) search window was set as 16. All other parameters such as Hadamard and the number of reference frames were carefully selected to be equivalent. For cases where frame rate equals 10 frames/s, a total of 100 frames were tested for each video sequence. For a case where frame rate equals 30 frames/s, a total of 300 frames were tested for each video sequence.

Among the sequences tested, two are described in Figs. 1–3. The difference between the actual bits and the target bits per frame is plotted in Fig. 1(a) for the sequence "Foreman," with an average mismatch of 1476 bits when [2] was used and only 228 bits when our scheme was used. Fig. 1(b) shows the same for the sequence "Carphone" with an average mismatch of 701 bits when [2] was used and only 166 bits when our scheme was used. It is therefore clear that the number of actual bits generated by using our method is closer to the target bits. Fig. 2 shows the PSNR value for each reconstructed video frame. In Fig. 2(a), for the sequence "Foreman," a total of 16 frames were forced to skip when [2] was used, whereas no frame was skipped when our scheme was used. Fig. 2(b) shows the case for the sequence "Carphone." It is clear that our method significantly improves the average PSNR when compared to that of [2]. Similar results have been obtained for all other sequences that we have tested. Fig. 3 shows buffer-level variations. When the buffer level exceeds 80% of the buffer size, frames are forced to skip to keep the buffer from an overflow. It was found that the actual buffer level using our scheme is kept closer to its target buffer level. This implies that our scheme produces a steadier buffer status. It is also noted that underflow occurs when [2] is used, as in Fig. 3(b). Although underflow does not affect motion continuity, it wastes channel bandwidth.

More results are reported in Table II. We achieved a PSNR gain of up to 1.09 dB with an average PSNR gain of 0.56 dB

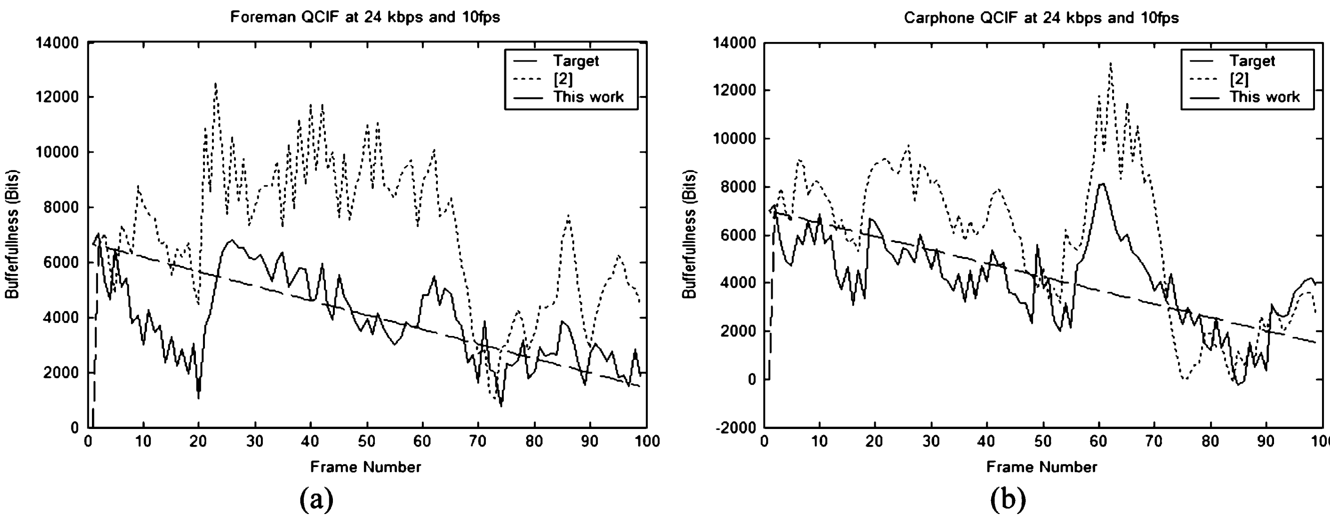

Fig. 3. Buffer level per frame for sequences (a) "Foreman" and (b) "Carphone."

TABLE II
SIMULATION RESULTS OF OUR PROPOSED ALGORITHM IN TERMS OF PSNR (dB) AND BIT RATES (kb/s) (WHEN $\mathrm{fps} = 10$, A TOTAL OF 100 FRAMES ARE CODED; WHEN $\mathrm{fps} = 30$, A TOTAL OF 300 FRAMES ARE CODED)

Sequence	Frame	Average PSNR (dB)			PSNR Std. Deviation		Bit Rates (kbps)	
		[2]	Ours	Gain	[2]	Ours	[2]	Ours
Carphone QCIF (24 kbps)	10	31.58	31.94	+0.36	3.80	2.60	24.11	24.14
Carphone CIF (32 kbps)	10	28.65	29.48	+0.83	4.57	2.89	32.55	32.32
Foreman QCIF (24 kbps)	10	29.48	30.57	+1.09	4.07	1.41	24.30	23.96
Foreman CIF (32 kbps)	10	26.96	27.59	+0.63	3.43	1.79	31.74	32.02
Stefan QCIF (128 kbps)	10	31.43	31.78	+0.35	1.92	1.79	128.44	128.20
Akiyo QCIF (48 kbps)	30	41.58	42.06	+0.48	1.46	1.63	47.93	48.01
Highway CIF (24 kbps)	10	34.20	34.40	+0.20	1.52	0.36	24.30	23.99
Salesman QCIF (64 kbps)	30	37.60	38.12	+0.52	1.98	1.85	64.06	63.96

over the entire test set. It was found that our method improves PSNR deviation well in most cases. Smaller PSNR fluctuation (standard deviation) implies a more stable visual quality and is highly desired in video streaming. Table II also shows that both the scheme in [2] and our scheme can achieve accurate target bit rates.

V. CONCLUSION

We have presented an improved λ and QP determination scheme for frame-layer rate control for H.264 video encoding. First, we consider the case where the computed QP is clipped to a smaller or a larger QP, meaning that the encoder will produce more or less bits than the frame target. Our proposed method first modifies the Lagrange multiplier (λ_{MODE}) for mode decision by using a computed QP which may be different from that used for encoding. Second, we consider the actual bits produced by previous MBs and the total target bits allocated to them. We then further modify λ_{MODE} using α_i. The objective of these measures is to produce bits as close to the frame target as possible. In addition, to obtain a better QP, we propose a complexity-based bit allocation scheme and a QP adjustment method. Our QP adjustment is applied to a frame due to a lower bound being imposed on texture bits. Furthermore, we adjust QP by monitoring the cumulative difference between the actual bits and the target bits. Our accurate QP improves RDO. As demonstrated in our experiments, our method reduces the gap between actual bits and target bits. In comparison, our proposed algorithm produces smoother visual quality variations with less skipped frames and results in similar or higher visual qualities, with an average PSNR gain of 0.56 dB over the entire test set. Our method also keeps buffer occupancy steadier.

REFERENCES

[1] T. Weigand, G. J. Sullivan, and A. Luthra, "Draft ITU-T Recommendation H.264 and Final Draft International Standard 14496–10 Advanced Video Coding," in *Joint Video Team of ISO/IEC JTCI/SC29?WG11 and ITU-T SG16/Q.6, Doc. JVT-G050r*, Geneva, Switzerland, May 2003.

[2] Z. G. Li, F. Pan, K. P. Lim, G. Feng, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for JVT," presented at the 7th Meeting JVT-G012-r1, Pattaya II, Thailand, Mar. 2003.

[3] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullian, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[4] M. Jiang, X. Yi, and N. Ling, "Improved frame-layer rate control for H.264 using MAD ratio," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, May 2004, vol. III, pp. 813–816.

[5] M. Jiang, X. Yi, and N. Ling, "Frame-layer bit allocation scheme for constant video quality," presented at the Proc. IEEE Int. Conf. Multimedia Expo., Taipei, Taiwan, R.O.C., Jun. 2004.

[6] Z. He, Y. K. Kim, and S. K. Mitra, "Low-delay rate control and smoothing for video coding via p-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 928–940, Aug. 2001.

[7] 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. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 472–487, Jun. 2003.

[8] *MPEG-4 Video Verification Model V18.0, Coding of Moving Pictures and Audio N3908*, ISO/IEC, JTC1/SC29/WG11, Jan. 2001.

[9] [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/2003_03_Pattaya/JVT-G012r1_software.zip

[10] G. Sullivan, T. Wiegand, and K. P. Lim, "Joint model reference encoding methods and decoding concealment methods," in *Section 2.6: Rate Control* (in JVT-I049), San Diego, CA, Sep. 2003.

[11] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172–185, Feb. 1999.

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

[13] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 186–199, Feb. 1999.

[14] H.-J. Lee, T. 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, Sep. 2000.

[15] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multi-resolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 533–545, Sep. 1994.

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

[17] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate-distortion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 246–250, Feb. 1997.