Lagrangian Multiplier Based Joint Three-Layer Rate Control for H.264/AVC

Miaohui Wang and Bo Yan, *Member, IEEE*

***Abstract*—Lagrangian multiplier (LM) based mode decision is one of the most important technologies in standard H.264/AVC encoder. Based on LM theory, this paper presents a joint three-layer (JTL) model for H.264/AVC rate control. At macroblock (MB) level, we dynamically revise LM for each of MBs by its estimated complexities, which is able to select a better coding mode than the current scheme with the constant LM adopted in H.264/AVC. At frame level, a more flexible and effective quantization parameter (QP) adjustment scheme is designed for I-frame to avoid buffer overflow or underflow. In addition, we also present a new target bits allocation scheme in group of picture (GOP) level. Experimental results show that our JTL model can not only significantly improve the video quality with the average PSNR gain up to 0.97 dB, but also provide a more stable buffer occupancy with respect to other existing rate control methods.**

***Index Terms*—H.264/AVC, Lagrangian multiplier, rate control.**

I. Introduction

H.264/AVC jointly developed by ISO and ITU is the state-of-the-art video coding standard. It achieves significant improvement in video coding performance, and meanwhile, it incorporates some advanced technologies, such as the Lagrangian function based rate distortion optimization (RDO), multiple reference pictures, variable block sizes and a quarter pixel precision for motion compensation etc. Unfortunately, some of these novel technologies also impose challenges to the rate control of the encoder and one of them is Lagrangian encoder control. In order to implement RDO, it requires quantization parameter (QP) before mode decision. QP is computed by rate-quantization (R-Q) model, which needs the mean absolute value (MAD) of the current frame. However MAD is only available after RDO. This constitutes the famous "chicken and eggs" dilemma [1].

In H.264/AVC, Lagrangian multiplier (LM) at macroblock (MB) level is a constant. Although it has achieved satisfying results, the constant LM is not the best choice for every MB with different complexities in one frame. Literature [2] shows that dynamically adjusted LM at MB level can achieve better performance than the constant one. Recently, many rate control algorithms have been proposed for H.264/AVC [3]–[7]. However most of them don't consider the existing LM problem. Jiang *et al.* proposed an LM adjustment algorithm in [2], which uses the actual bits of the encoded MBs in the same frame to update the LM of current MB in real time. However, there are two problems with Jiang's method. One is that it doesn't consider the complexity of current MB while revising its LM. The other is that the total target bits of MBs in the current frame is seldom equal to the predefined frame target bits after coding the current frame. In addition, in the existing rate control algorithms of H.264/AVC, the target bits allocated to the current group of picture (GOP) usually suffer from the bandwidth occupation of the previous GOP, and it significantly influences the current target bits allocation.

In order to solve the above problems, this paper presents a joint three-layer (JTL) rate control model for H.264/AVC, which covers MB, frame and GOP levels. Firstly, we propose to adjust the LM of current MB according to the MAD of the collocated MB in previous frame at MB level. Then at frame level, we propose a flexible adaptive QP adjustment scheme for I-frame. Finally, in GOP level, we introduce a new GOP target bits allocation scheme to smoothen the target bits allocation bottleneck between neighboring GOPs.

The rest of this paper is organized as follows. In Section II, we present our JTL rate control model in MB, frame and GOP levels respectively. Then we evaluate JTL model by simulations and present the results in Section III. Finally, in Section IV we draw the conclusion.

II. Joint Three-Layer (JTL) Rate Control

A. Adaptive LM Adjustment

In H.264/AVC, RDO determines the option of the best block mode. During video encoding process, RDO mode decision is implemented for each inter-mode and intra-mode { Skip, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4, Intra 4×4, Intra 16×16, direct } [8]. The mode with the minimal rate distortion (RD) cost is the optimum selection. RD cost at frame level is defined as:

$$J(MODE|QP,\lambda_{MODE}) = SSD(MODE|QP) + \lambda_{MODE}\times R(MODE|QP) \quad (1)$$

where $SSD(MODE|QP)$ is the sum of squared difference between the original block and its reconstructed block in one frame. $R(MODE|QP)$ represents the frame bit rate associated with the selected block mode. QP is the quantization parameter. λ_{MODE} is the LM given by [8]:

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

Manuscript received February 22, 2009; revised April 14, 2009. First published May 02, 2009; current version published June 05, 2009. This work was supported in part by the NSFC under Grant 60703034, and in part by Shanghai Pujiang Program under Grant 07PJ14017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. James E. Fowler.

The authors are with the School of Computer Science, Fudan University, Shanghai, China (e-mail: byan@fudan.edu.cn).

Digital Object Identifier 10.1109/LSP.2009.2022147

Similarly, the RD cost process at MB level is analyzed as

$$J_{MB[i]} = SSD_{MB[i]} + \lambda_{MODE} \times R_{MB[i]} \tag{3}$$

where $SSD_{MB[i]}$ is the sum of squared difference between the ith MB and its reconstruction. $R_{MB[i]}$ is the bit rate associated with the ith MB.

Equations (2) and (3) contain the following relation [2]: a smaller λ_{MODE} corresponds to smaller QP, lower $SSD_{MB[i]}$ and higher $R_{MB[i]}$. The reverse is also true. This relation implies that we can adjust LM to influence the output bit rate and distortion, which is the basis of our JTL model.

We employ the MAD of MBs to measure the complexity of the current MB. The MAD of ith MB in frame $n-1$ is given by

$$MAD_{MB}[i, n-1] = \frac{1}{M \times N} \times \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |I_{cur}(x, y) - I_{ref}(x - \Delta x, y - \Delta y)| \tag{4}$$

where $I_{cur}(x, y)$ and $I_{ref}(x - \Delta x, y - \Delta y)$ indicate the luminance value of the current MB at the location of (x, y) in current frame and the location $(x - \Delta x, y - \Delta y)$ in its reference frame respectively. Δx and Δy are the motion vectors. M and N denote the horizontal and vertical dimensions of the current MB respectively.

In H.264/AVC, λ_{MODE} is a constant at MB level. As we analyzed above, the constant λ_{MODE} means that MBs in one frame have the same distortion and bit rate level. Since the complexities of MBs in one frame are different, the fixed λ_{MODE} can hardly satisfy the optimum bits allocation. Ideally, LM at MB level should fluctuate along with the different complexities of MBs. In other words, if the texture of MB is simple, LM should be increased and it can save bits for other complex MBs in the current frame. Otherwise, LM should be reduced, which indicates that the current complex MB can get more bits for coding than the constant LM. Hence we employ adjustment factor $\alpha_{i,n}$ to further revise LM:

$$\lambda_{MODE}[i, n] = \alpha_{i,n} \times \lambda_{MODE} \tag{5}$$

where $\lambda_{MODE}[i, n]$ is the λ_{MODE} for ith MB in frame n.

After testing various video sequences, we find that the MAD value of MB in current frame is similar to that of the same spatial location in the previous frame. Two of them are shown in Fig. 1. Because of the complexity similarity between the ith MB in frame n and the corresponding MB in frame $n-1$, we employ $MAD_{MB}[i, n-1]$ in frame $n-1$ to define the LM adjustment factor $\alpha_{i,n}$ in frame n. We also find that exponential relationship below can better revise LM in comparison to others:

$$\alpha_{i,n} = \left(\frac{MAD_{Frame}[n-1]}{MAD_{MB}[i, n-1]} \right)^{\phi} \tag{6}$$

where $MAD_{MB}[i, n-1]$ is the MAD of ith MB in frame $n-1$. $MAD_{Frame}[n-1]$ is the mean value of the $MAD_{MB}[i, n-1](i = 0, 1 \ldots)$. In our experiment, ϕ is set as 1/4.

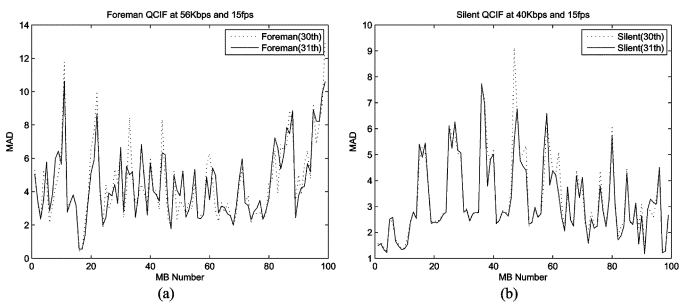

Fig. 1. MAD versus MB number for sequences. (a)"Foreman"; (b)"Silent."

B. QP Adjustment

I-frame, absolutely, is the crucial ingredient in video sequence. Low quality I-frames, which are used as the reference frame for the following P frame, will certainly degrade the whole video sequence quality. On the contrary, high quality I-frame usually produces more bits than P or B frames in the same GOP and may corrupt the video quality due to buffer overflow and frame skip.

Let QP_I present the QP of current I-frame. QP_I derived from JVT-G012 [9] can't work well in two special cases. When the video is smooth, QP_I is always too big for actual coding environment and it wastes bandwidth (buffer underflow). If the video context changes quickly, QP_I may become so small for coding that it leads to frame skip (buffer overflow). Various test sequences show that QP_I in JVT-G012 [9] is not flexible to deal with these two special video cases. Based on above analysis, we consider the buffer occupancy ratio and the number of skipped frame ratio in previous GOP to estimate and further revise QP_I of the current I-frame. If either the buffer occupancy ratio or the number of skipped frame ratio is bigger than the predefined threshold, it implies that the current QP_I is too small for current I-frame. On the contrary, if both of them are smaller than the predetermined threshold, it indicates that the current QP_I is too big for current I-frame. So we revise QP_I as follows:

$$QP_I = \begin{cases} QP_I - 1, & \frac{N_{SP}}{Np} \le th_1 \text{ and } BR \le th_1 \\ QP_I + 1, & th_1 < \frac{N_{SP}}{Np} \le th_2 \text{ or } BR \ge th_2 \\ QP_I + 2, & th_2 < \frac{N_{SP}}{Np} \le th_3 \\ QP_I + 4, & \frac{N_{SP}}{Np} > th_3 \end{cases} \tag{7}$$

where N_{SP} and N_P indicate the number of skipped and total P-frame in the previous GOP respectively. The initial values of them are zero for the first GOP. BR is the buffer occupancy ratio before encoding the current GOP. $th\{1, 2, 3\}$ are thresholds which are used to measure the current buffer occupancy level and frame skip level in our experiment, whose typical values are set as 0.10, 0.30 and 0.55 respectively.

In addition, the QP choice for P-frame is computed by quadratic R-Q model in [10].

C. Target Bits Allocation for GOP

In H.264/AVC, the total target bits allocated to GOP level are given by [11]:

$$T_r(n_{i,0}) = \frac{u(n_{i,1})}{F_r} \times N_{gop} + T_r(n_{i-1,N_{gop}}) \tag{8}$$

where N_{gop} is the total number of frames in one GOP. $T_r(n_{i-1,N_{gop}})$ denotes the occupancy of virtual buffer after coding the $(i-1)$th GOP. $u(n_{i,1})$ is the available channel bandwidth of the 1st frame in ith GOP. F_r is the predefined frame rate. We denote the total target bits allocated to the current GOP by $T_r(n_{i,0})$.

$T_r(n_{i,j})$ is updated frame by frame as follows:

$$T_r(n_{i,j}) = T_r(n_{i,j-1}) - A(n_{i,j-1}) \tag{9}$$

where $A(n_{i,j-1})$ is the number of bits generated by the $(j-1)$th frame in the ith GOP. $T_r(n_{i,j})$ is the remaining bits after coding the $(j-1)$th frame.

In the first several GOPs of the coded sequences, the coded stream is hard to converge to the predefined target bits and each GOP usually produces more bits than its target bits. Therefore the bandwidth of the current GOP is very likely to be occupied by previous GOP, which is denoted by negative remaining bits in JM12.2 [11]. In order to smoothen this bottleneck, we propose to disperse the number of remaining bits of the previous GOP to the following GOPs at the beginning of coding the current GOP.

In our model, if the number of remaining bits is positive, JVT-G012 method is used to allocate bits in GOP level. Otherwise, the target bits are allocated as follows.

Different from (8), the bits allocated to the ith GOP are given by:

$$T_r(n_{i,0}) = \frac{u(n_{i,1})}{F_r} \times N_{gop} + \beta \times T_r(n_{i-1,N_{gop}}) \tag{10}$$

where

$$\beta = a \times N_{gop} + b. \tag{11}$$

Various experiments show that linear relation between β and N_{gop} can effectively mitigate the allocation bottleneck. In our experiment, a and b are adjustment factors, whose typical values are set as 0.003 and 0.02 respectively.

Then the number of remaining bits $T_r(n_{i,j})$ $(j = 0, 1, \ldots)$ is updated frame by frame according to (9). At the end of the ith GOP, the number of remaining bits $T_r(n_{i,N_{gop}})$ can be updated by

$$T_r(n_{i,N_{gop}}) = T_r(n_{i,N_{gop}}) + (1-\beta) \times T_r(n_{i-1,N_{gop}}). \tag{12}$$

In order to improve the coding performance, scene detection is required for rate control. The difference of histogram (DOH) is used for scene detection, which is defined by

$$DOH_n = \frac{1}{M \times N} \sum_{i=0}^{255} \left| h_n^Y(i) - h_{n-1}^Y(i) \right| \tag{13}$$

where h_n^Y and h_{n-1}^Y are luminance histograms for frame n and $n-1$ respectively. M and N are the width and height of the current frame respectively. The scene change threshold (Th) is set as 0.5 which can work well. If $DOH_n \geq Th$, the current frame will be encoded in a new GOP with an I-frame mode [9] and the following frames will be encoded by the JTL method.

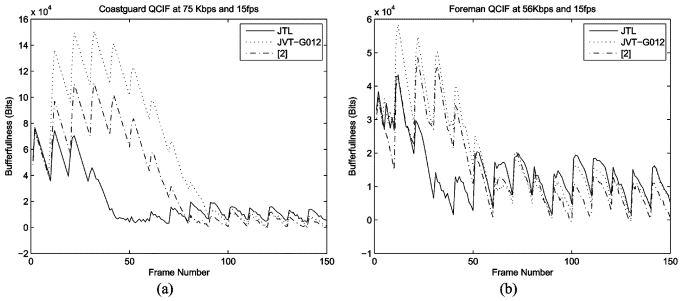

Fig. 2. Buffer fullness comparison versus frame number for sequences (a)"Coastguard"; (b)"Foreman."

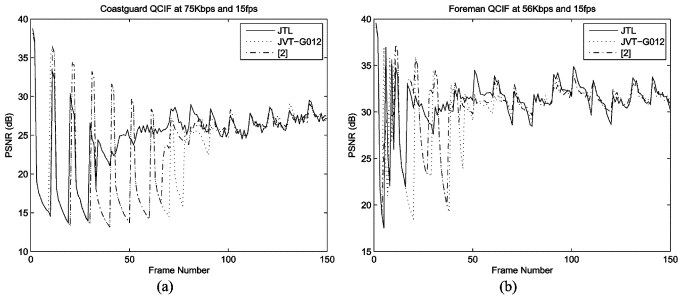

Fig. 3. PSNR comparison versus frame number for sequences (a)"Coastguard"; (b)"Foreman."

III. Simulation Results

H.264/AVC reference software JM12.2 [11] is implemented to evaluate the performance of our proposed JTL model. The simulation was conducted with the first 150 frames of several standard QCIF and CIF test sequences, including "Akiyo," "Silent," "Foreman," "Bus," "Coastguard," and "Hall-monitor." "Foreman-Silent" was created by cascading two corresponding sequences. The scene change was set at frame number 58.

The rate control algorithm JVT-G012 [9] and [2] are selected as the references for comparison. In the simulation, we implement skip method for three algorithms described in [2]. If the current buffer fullness exceeds 80% of the buffer size, the encoder will skip frames to prevent the buffer from overflow. When the current frame is skipped, the previous reconstructed frame is used to compute the PSNR of the current frame. The buffer size is set to 2/3 of the channel bandwidth (one second). Each sequence is coded at 15 fps by GOP size 10 with structure IPPP. The search window is set to 32 pixels in baseline profile. UVLC, RDO and Rate Control are enabled. All other parameters are selected identically for three algorithms.

Among the test sequences, two are presented in Figs. 2 and 3. Fig. 2 shows the buffer fullness comparison of three algorithms for sequences "Coastguard" and "Foreman." As these figures show, the buffer fullness variation of our proposed algorithm is much smaller than those of JVT-G012 and [2]. As a result, buffer overflow happens frequently for JVT-G012 and [2], which will degrade the video quality significantly.

Fig. 3 illustrates the PSNR comparison of three algorithms for sequences "Coastguard" and "Foreman." We can see from this figure, unlike other schemes which fluctuate wildly, our JTL model is able to keep the PSNR values on a relatively steady level. This would imply a more stable visual quality.

TABLE I
PERFORMANCES OF THREE ALGORITHMS IN TERMS OF NO. OF SKIPPED FRAMES, AVERAGE PSNR, PSNR STD. DEVIATION AND BIT RATES

Sequence	Method	No. of Skipped Frames	Average PSNR (dB)	PSNR Std. Deviation	Bit Rates (kbps)
Silent (QCIF, 40 Kbps)	JVT-G012	36	31.15	3.34	39.96
	[2]	42	31.03	3.48	39.89
	JTL	15	31.43	2.39	40.30
Akiyo (QCIF, 40 Kbps)	JVT-G012	23	37.06	2.27	40.32
	[2]	23	36.74	2.27	40.54
	JTL	8	37.71	2.04	40.61
Foreman (QCIF, 56 Kbps)	JVT-G012	29	30.86	4.03	56.65
	[2]	17	30.96	3.06	56.08
	JTL	10	31.25	2.87	56.52
Hall-Monitor (QCIF, 56 Kbps)	JVT-G012	9	36.32	2.65	56.25
	[2]	3	36.57	1.94	56.07
	JTL	4	36.80	1.98	56.27
Hall-Monitor (CIF, 56 Kbps)	JVT-G012	13	30.86	2.22	56.36
	[2]	7	31.01	1.48	56.76
	JTL	7	31.31	1.65	56.83
Coastguard (QCIF, 75 Kbps)	JVT-G012	20	29.10	2.56	75.18
	[2]	18	29.10	2.54	74.93
	JTL	10	29.43	2.07	75.16
Bus (QCIF, 75 Kbps)	JVT-G012	60	24.80	6.21	75.14
	[2]	51	25.13	5.99	75.04
	JTL	21	25.58	4.37	75.54
Bus (CIF, 145 Kbps)	JVT-G012	13	24.31	2.80	145.50
	[2]	13	24.22	3.00	145.13
	JTL	7	24.59	2.20	145.81
Foreman-Silent (QCIF, 56 Kbps)	JVT-G012	29	32.05	5.25	56.05
	[2]	17	32.06	4.27	55.94
	JTL	10	32.39	3.86	56.31

More detailed numerical experimental results are tabulated in Table I. From this table, we can observe that our proposed algorithm is able to reduce the number of skipped frames significantly compared with [9] and [2]. As a result, the proposed JTL can provide up to 0.78 dB and 0.97 dB better average PSNR performance than [9] and [2], while three algorithms can achieve accurate output bit rates. This table also verifies that our algorithm can significantly reduce the PSNR standard deviations of all frames and produce more stable visual quality than others.

It should be noted that the increment of computation complexity for our JTL method is trivial for the H.264/AVC coding process. For instance, simulation results show that the average encoding time of JTL (52.768 s, 53.680 s) is only increased by 0.32% and 0.45% respectively in comparison to JVT-G012 [9] (52.751 s, 53.656 s) for sequences "Foreman" and "Coastguard."

IV. CONCLUSIONS

In this letter, we present a highly efficient JTL model for H.264/AVC rate control. Firstly, at MB level, we introduce an adjustment factor to revise the constant LM, which can make better use of the target bits. Secondly, an I-frame QP adjustment algorithm is proposed in order to handle the frame-skip case. Finally, we present a new GOP-level target bits allocation scheme, which can smoothen the bottleneck of the target bits allocation between adjacent GOPs. Numerous experimental results indicate that our JTL model can provide smoother visual quality and higher average PSNR as a result of the steadier buffer occupancy than other rate control methods.

REFERENCES

[1] Z. Li, F. Pan, K. Lim, X. Lin, and S. Rahardja, "Adaptive rate control for H.264," in *Proc. of IEEE Int. Conf. Image Processing (ICIP'04)*, Oct. 2004, pp. 745–748.

[2] 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.

[3] X. Jing, L.-P. Chau, and W.-C. Siu, "Frame complexity-based rate-quantization model for H.264/AVC intraframe rate control," *IEEE Signal Process. Lett.*, vol. 15, pp. 373–376, Mar. 2008.

[4] H. Wang and S. Kwong, "Rate-distortion optimization of rate control for H.264 with adaptive initial quantization parameter determination," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 140–144, Jan. 2008.

[5] B. Yan and M. Wang, "Adaptive distortion-based intra-rate estimation for H.264/AVC rate control," *IEEE Signal Process. Lett.*, vol. 16, no. 3, pp. 145–148, Mar. 2009.

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

[7] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "A novel two-stage rate control scheme for H.264," in *Proc. IEEE Int. Conf. Multimedia and Epro (ICME'06)*, Jul. 2006, pp. 673–676.

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

[9] Z. Li, F. Pan, and K. Lim, "Adaptive basic unit layer rate control for JVT," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG012, 7th Meeting*, Pattaya, Thailand, Mar. 2003.

[10] 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.

[11] [Online]. Available: http://iphome.hhi.de/suehring/tml/download