CODING-GAIN-BASED COMPLEXITY CONTROL FOR H.264 VIDEO ENCODER

Ming-Chen Chien[1,2], *Zong-Yi Chen*[1], *and Pao-Chi Chang*[1]

1 Department of Communication Engineering, National Central University, Taiwan
2 Department of Electrical Engineering, Chin Min Institute of Technology, Taiwan

ABSTRACT

The allowable computational complexity of video encoding is limited in a power-constrained system. Different video frames are associated with different motions and contexts, and so are associated with different computational complexities if no complexity control is utilized. Variation in computational complexity leads to encoding delay jittering. Typically motion estimation (ME) consumes much more computational complexity than other encoding tools. This work proposes a practical complexity control method based on the complexity analysis of an H.264 video encoder to determine the coding gain of each encoding tool in the video encoder. Experiments performed on a programming optimized source code show that the computational complexity associated with each frame is well controlled below a given limit with very little R-D performance degradation under a reasonable constraint comparing to the unconstrained case.

***Index Terms*—**Complexity control, complexity allocation, video encoder, H.264

1. INTRODUCTION

The real-time video encoding is an important element for many applications over various wireless networks. To avoid encoding delay jittering, the available encoding time of each video frame, T_{FC}, is limited in the real-time video encoding system and can be defined as

$$T_{FC} = \frac{1}{fr} \tag{1}$$

where *fr* represents the frame rate. The limited encoding time of each frame limits the available computational complexity of each frame, C_{FC}, which can be defined as

$$C_{FC} = C_{PRC} \cdot T_{FC} = \frac{C_{PRC}}{fr} \tag{2}$$

where C_{PRC} represents the clock rate of the processor. However, the C_{PRC} of the processor embedded in wireless handsets is limited and hence C_{FC} is also limited.

Optimal complexity control aims to control the encoding complexity of each frame under a given limit while achieving optimal R-D performance as follows:

$$\begin{aligned} &\min J = \min\{D + \lambda R\} \\ &s.t. \\ &c_F \le C_{FC} \end{aligned} \tag{3}$$

where D denotes distortion; R denotes bit rate; λ denotes the Lagrange multiplier; J denotes the *R-D* cost, and c_F denotes the complexity used for a frame.

Traditionally, the complexity constraint is computed in the frame layer as described above. For typical MPEG-like video encoders, a frame is partitioned into a number of MBs while an MB is the basic encoding unit. Different MBs have various motions and contexts and hence are associated with different complexities. Therefore, the allocation of C_{FC} among MBs is a critical problem. Typically, MPEG-like video encoders use many encoding tools, such as ME, DCT, Q, entropy coding and others. Different encoding tools may exhibit substantially different coding efficiency. Accordingly, allocating complexity among encoding tools is another key problem.

A metric of coding gain which represents the coding efficiency has been proposed [4] as follows:

$$CG = \Delta J / \Delta C \tag{4}$$

$$\Delta J = \Delta D + \lambda \Delta R \tag{5}$$

where ΔC represents the increase in complexity when an encoding tool is adopted; ΔD represents the decrease in distortion; ΔR represents the decrease in rate, and λ is the Lagrange multiplier. However, a proper λ is not easily determined. When the rate control is turned on for a target rate, ΔR becomes nearly zero, and ΔJ equals ΔD :

$$CG = \Delta D / \Delta C \tag{6}$$

A few works of complexity control have been conducted [2],[3],[4],[5]. The optimization formula of the first C-R-D model [2] is too complicated to be solved in closed form. Also, an MHM-based method for allocating complexity for ME among MBs, which was not optimal, was also proposed in that study. A statistical optimal operation mode for a sequence in a complexity-constrained video encoding system has also been proposed [3]. However, an optimal operation mode could be optimal for a frame but inadequate for another frame. A complexity allocation method for ME based on the cost-complexity curve has been proposed [4]. A C-R-D optimization for H.264 ME has also been proposed [5]. It proposed two Lagrange multipliers to terminate the complexity-inefficient ME rounds and thus increase coding efficiency. Typically ME consumes most complexity with a large variation between MBs. In general, optimal complexity control algorithms are difficult to apply to practical real-time video encoders because of their large computational overhead. To the best of our knowledge, no practical complexity control that is efficiently enough and operates in real time exists for an H.264 video encoder.

Based on complexity analysis of a programming optimized H.264 code, X264 [10], this work proposes a simple and practical complexity control method which can control the encoding complexity of each frame under a given limit while achieving very good R-D performance.

This paper is organized as follows. Section 2 proposes a practical complexity control method based on the results of complexity analysis. Section 3 presents experimental results, and section 4 draws conclusions.

2. PROPOSED COMPLEXITY CONTROL

For a typical MPEG-like video encoder, Figure 1 displays the encoding block diagram of an MB. DCT, Q, Q^{-1}, IDCT have been collectively denoted by PRECODING [2]. This paper follows this notation, and divides the encoder into three major encoding tools - ME, PRECODING, and entropy coding.

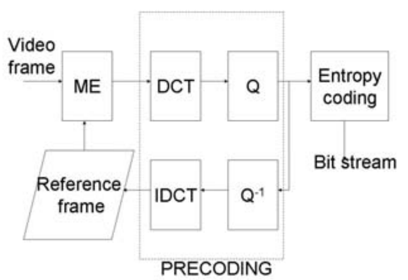

Fig. 1 Basic block diagram of a video encoder

Highly efficient complexity control should be performed by allocating complexity to the encoding tools with higher coding gain. This work conducts experiments with the options presented in Table I to analyze the coding gains of various encoding tools in the modern H.264 encoder. The metric of coding gain is given by (6), where ΔD is represented by $\Delta PSNR$, which represents the increase in PSNR, and ΔC is measured by the number of CPU clocks spent on a piece of code. Table II presents the results, which will be discussed in the following subsection.

Table I.
Options for complexity analysis

Video source	Foreman QCIF, Carphone QCIF
Fast ME	Diamond
Target rate	103k bps
Frame rate	20
Number of reference frames	1
GOP type	IPPPP
CPU	Intel Pentium 4 2.66G Hz
RAM	512M bytes
MMX tech.	On for SAD computation
Source code of H.264	X264

Table II.
Coding gain of each encoding tool

Encoding tool	Coding gain (db/kclks)
CABAC (compare to CAVLC)	9.17e-4
half pixel ME	2.88e-3
Deblocking filter	8.54e-4
Quarter pixel ME	4.45e-4
8x8 partition mode	1.42e-4
16x8 & 8x16 partition mode	4.63e-5
Sub8x8 partition mode	4.7e-5
4x4 Intra	5.22e-6
5 reference frames	4.06e-5

2.1. Complexity Allocation

The complexity allocation allocates complexity from frame layer to MB layer. It should be performed before the first MB in a frame is encoded. When the video encoder starts to encode a frame, it should do some initialization before encoding slices. Complexity control records the complexity consumed by the initialization, which is denoted by C_{Finit}. The complexity budget of encoding all slices in a frame is C_{SLs}. After the slices are encoded, deblocking filtering can be performed; it is followed by updating references and other necessary tasks. The complexity of these tasks after the encoding of slices, C_{Fother}, should be reserved. The deblocking filter is suggested to be adopted because it has high coding gain as shown in Table II and proposed elsewhere [6]. C_{Fother} is smaller than C_{SLs} as displayed in Fig. 2, and it does not vary greatly. It can be regarded as a constant and can be estimated from the previous frame. Accordingly, before the slices are encoded, by measuring C_{Finit} and reserving C_{Fother}, C_{SLs} can be allocated by

$$C_{SLs} = C_{FC} - C_{Finit} - C_{Fother} \quad (7)$$

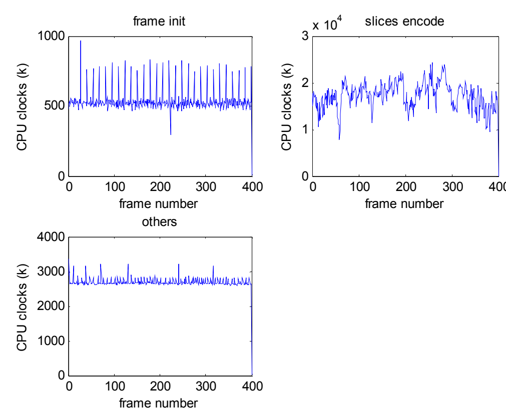

Fig. 2 Complexity consumption in the frame layer

The operation of the slice layer is very simple. Only a short slice header is added. The complexity of encoding all slice headers in a frame is small and can be treated as a constant. It is denoted by C_{SLhs}. Therefore, the complexity of encoding all MBs in a frame, C_{MBs}, can be allocated according to

$$C_{MBs} = C_{SLs} - C_{SLhs} \quad (8)$$

Each MB can adopt ME, PRECODING and entropy coding. Typically, ME consumes most of the complexity, as shown in Fig. 3. It is the main object on which complexity control will be performed. The modern entropy coding tool CABAC has a high coding gain, as shown in Table II and elsewhere [6]. Its adoption is recommended. The modern video encoding standard H.264 significantly simplifies DCT operation [6]. Hence, PRECODING has high coding gain, and is destined to be adopted. Some early termination algorithms for PRECODING have been proposed to skip the PRECODING for the MB with small residual signals [11]. All such algorithms with high efficiency can be utilized. As described above, the complexity for PRECODING and entropy coding should be reserved. The complexity budget C_{MEs} can be allocated using

$$C_{MEs} = C_{MBs} - C_{MBother} \times M \qquad (9)$$

where $C_{MBother}$ denotes the complexity reserved for PRECODING and entropy coding of a MB and M is the number of MBs in a frame. Figure 3 shows $C_{MBother}$ is relatively small and its variation is much smaller than C_{ME}, the complexity for ME of a MB. Therefore, $C_{MBother}$ can be treated as a constant and can be estimated statistically by running test video sequences in advance. The complexity compensation described below will eliminate the estimation error.

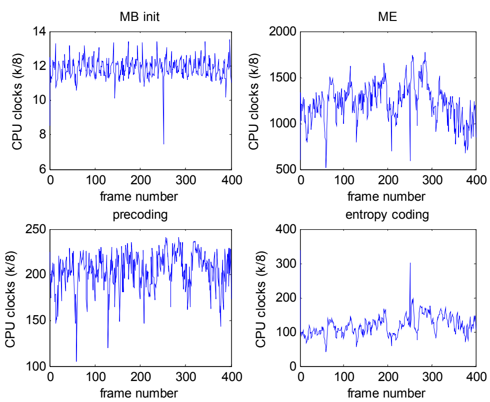

Fig. 3 Complexity consumption in the MB layer

The complexity allocation for ME among MBs is suggested to be weighted by *COST0* as

$$C_{ME}(i) = C_{MEs} \times \frac{COST0_i}{\sum_{j=1}^{M} COST0_j}, \quad i = 1,2,\ldots,M \qquad (10)$$

where *COST0* represents the cost of ME with zero MV in 16x16 partition mode. This equation is simple but meaningful because *COST0* contains information about context and motion. Since the MB with larger motion or more complex context has larger *COST0*, it deserves larger complexity budget. Otherwise, a larger bit rate and larger distortion will be generated.

2.2. ME Flow in Decreasing order of CG

According to the coding gain in Table II, the ME flow in Fig. 4 is suggested. The resulting operation order is similar to that suggested elsewhere [5] but the adoption of 4x4 Intra prediction is different. Table II reveals that the coding gain of 4x4 Intra for inter frames is very low, because most MBs in the inter frame choose inter mode as the best mode. However, 4x4 Intra prediction is beneficial to MBs that choose the Intra mode. The tendency to Intra mode is examined by comparing 16x16 ME and 16x16 Intra prediction. If the 16x16 Intra prediction yields a better performance, 4x4 Intra prediction can be utilized to reduce the residual signal. Otherwise, 4x4 Intra prediction is not used.

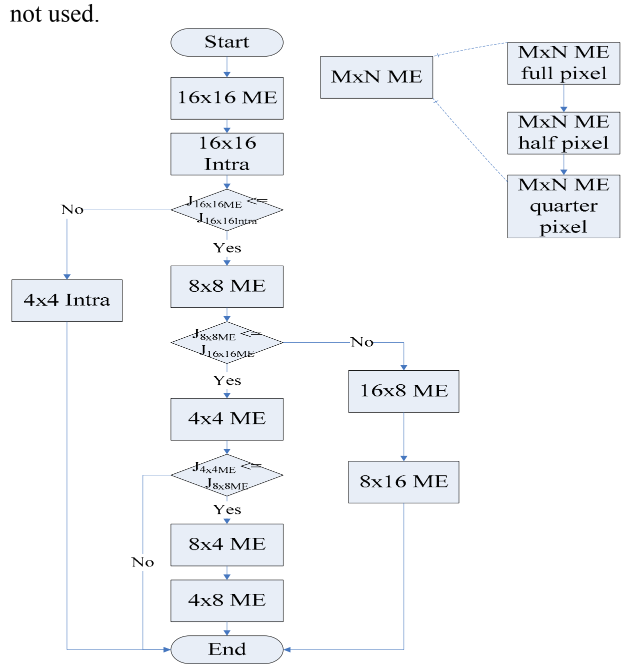

Fig. 4 ME flow in decreasing CG of encoding tools

2.3. Complexity Check and Compensation

After each computation of SAD and the R-D cost, the used complexity C_{MEused} is examined. If C_{MEused} exceeds C_{ME}, the ME process terminates. Otherwise, the ME process continues.

Any efficient early termination algorithm for PRECODING can be employed. Complexity compensation described below will distribute the saved complexity.

After the whole process of the MB encoding is complete, the balance $C_{MBbalance}$ between the used complexity C_{MBused} and the budget C_{MB} is given by

$$C_{MBbalance} = C_{MB} - C_{MBused} \qquad (11)$$

where C_{MB} is obtained by

$$C_{MB} = C_{ME} + C_{MBother} \qquad (12)$$

Then $C_{MBbalance}$ is distributed uniformly to the remaining MBs in that frame.

3. EXPERIMENTAL RESULTS

The options of experiments for the proposed practical complexity control are shown in Table III. The complexity metric is the number of CPU clocks used by an encoding tool, as measured by the ‘rdtsc’ instruction of an Intel CPU [7].

Figure 5 indicates that the complexity is well controlled under the given limit. The complexity of each frame rarely exceeds the bound. Figure 6 and 7 show that the rate and PSNR under complexity control are both very close to those in the unconstrained case. Figure 8 plots the R-D performance with Foreman video sequence under various complexity constraints, where Cfm denotes the maximum complexity of a frame without complexity constraint. When C_{FC} is down to 72% of Cfm, the PSNR obtained by this algorithm only degrades less than 0.5 dB at the same rate. When C_{FC} is down to 58% of Cfm, the PSNR obtained by this algorithm degrades no more than 1 dB at the same rate. Experiments with another video source 'Carphone' yield similar results.

Table III.
Options for complexity control

C_{Fmax} (clk)	source	QP	Rate control	Fast ME	Complexity metric
23 M	Foreman	29	off	Diamond	CPU clock

4. CONCLUSION AND FUTURE WORK

This work proposes an efficient complexity control method with very little degradation of R-D performance. The proposed method, which has very low overhead, is also very practical.

5. REFERENCES

[1] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264/ISO/IEC 14496-10 AVC)", in JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2003.

[2] Z. He and Y. F. Liang, "Power-Rate-Distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645-658, May 2005.

[3] D. N. Kwon and P. F. Driessen, "Performance and computational optimization in configurable hybrid video system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 31-42, Jan. 2006.

[4] C. Kim and J. Xin, "Hierarchical complexity control of motion estimation for H.264/AVC," MITSUBISHI ELECTRIC RESEARCH LABORATORIES, TR2006-004, Dec. 2006. Available: http://www.merl.com

[5] Y. Hu, Q. Li, S. Ma, and C. C. J. Kuo, "Joint rate-distortion-complexity optimization for H.264 motion search," in *Proc. ICME 2006*, pp. 1949-1952.

[6] E. G. Richardson, *H.264 and MPEG-4 Video Compression.* John Wiley & Sons, 2003.

[7] http://www.intel.com

[8] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7-28, Apr. 2004.

[9] Joint Model reference software version 10, Available: http://iphome.hhi.de/suehring/tml/index.htm

[10] x264, Available: http://developers.videolan.org/x264.html.

[11] Z. Chen, P. Zhou, and Y. He, "Fast Integer Pel and Fractional Pel Motion estimation in for JVT," JVT-F017r1.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, Island, JP, 5-13 Dec. 2002.

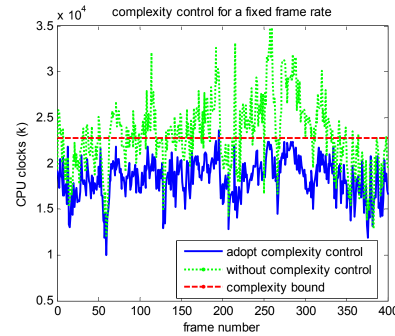

Fig. 5 Comparisons of computational complexity with and without complexity control

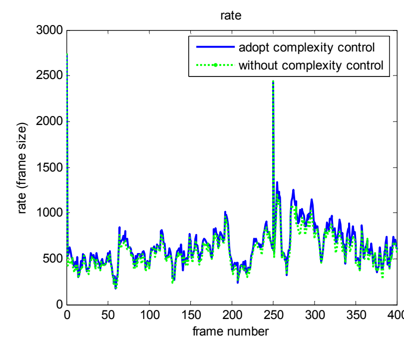

Fig. 6 Comparisons of rate with and without complexity control

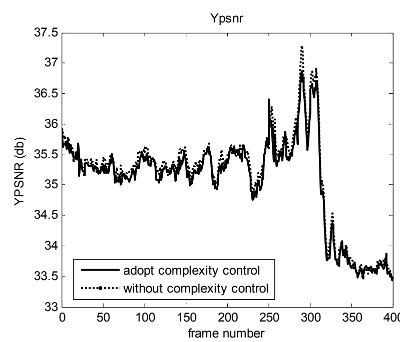

Fig. 7 Comparisons of YPSNR with and without complexity control

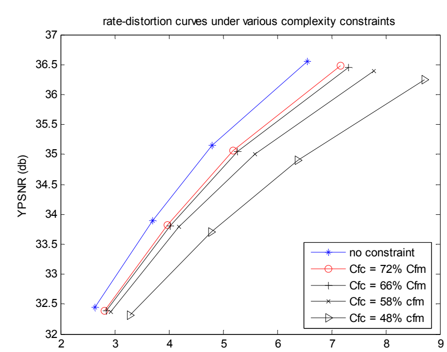

Fig. 8 R-D performance under various complexity constraints