MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Hierarchical Complexity Control of Motion Estimation for H.264/AVC

Changsung Kim, Jun Xin, Anthony Vetro

TR2006-004 February 09, 2006

Abstract

The latest H.264/AVC video compression standard promises significantly higher compression efficiency than prior standards. H.264/AVC achieves coding gains through a rich set of advanced coding tools including variable block size motion compensation, quarter-pel motion compensation and long-term memory motion compensation [1]. However, with so many coding options available, it has become extremely challenging to efficiently choose the coding parameters, including motion vectors and prediction modes, such that near optimal compression efficiency is achieved [2,3]. In literature, various attempts have been made in order to reduce the complexity of mode decision and motion estimation. Zhou et al. [5] proposed to determine initial search center based on the correlation between motion vectors of different block sizes. Fast motion estimation algorithms such as EPZS [6], UMHexagonS [7], and SEA [8] have been proposed to reduce the number of searching points in motion estimation, while the recent-biased search [9] and forward motion trace [10] have been introduced to reduce the complexity of long term memory motion compensation. A mode decision algorithm based on a coarse-to-fine approach [11] has been proposed assuming a monotonic rate-distortion (RD) relation across block sizes. Despite these efforts, further complexity reduction is still very desirable for practical applications. In this work, we propose a new hierarchical complexity control framework to efficiently control the complexity of encoding process, with focus on motion estimation and mode decision algorithms. The goal is to provide a complexity scalable encoder that may be applied to various applications.

SPIE Conference on Visual Communications and Image Processing (VCIP) 2006

Hierarchical Complexity Control of Motion Estimation for H.264/AVC

Changsung Kim[1], Jun Xin[2] and Anthony Vetro[2]

[1]University of Southern California
[2]Mitsubishi Electric Research Laboratories
201 Broadway Cambridge, MA 02139
Email: changsuk@usc.edu, {jxin,avetro}@merl.com

1. INTRODUCTION

The latest H.264/AVC video compression standard promises significantly higher compression efficiency than prior standards. H.264/AVC achieves coding gains through a rich set of advanced coding tools including variable block size motion compensation, quarter-pel motion compensation and long-term memory motion compensation [1]. However, with so many coding options available, it has becomes extremely challenging to efficiently choose the coding parameters, including motion vectors and prediction modes, such that near optimal compression efficiency is achieved [2,3]. In literature, various attempts have been made in order to reduce the complexity of mode decision and motion estimation. Zhou *et al.* [5] proposed to determine initial search center based on the correlation between motion vectors of different block sizes. Fast motion estimation algorithms such as EPZS [6], UMHexagonS [7], and SEA [8] have been proposed to reduce the number of searching points in motion estimation, while the recent-biased search [9] and forward motion trace [10] have been introduced to reduce the complexity of long term memory motion compensation. A mode decision algorithm based on a coarse-to-fine approach [11] has been proposed assuming a monotonic rate-distortion (RD) relation across block sizes. Despite these efforts, further complexity reduction is still very desirable for practical applications. In this work, we propose a new hierarchical complexity control framework to efficiently control the complexity of encoding process, with focus on motion estimation and mode decision algorithms. The goal is to provide a complexity scalable encoder that may be applied to various applications.

2. PROBLEM STATEMENT

The proposed complexity control distributes computational resources, i.e. encoding time, to individual encoding units based on their impact towards improving the overall rate-distortion (RD) performance. The objective is to dynamically reduce the encoding complexity with minimum RD performance penalty. Complexity control is based on the hierarchical encoding units, such as Group of Pictures (GOP), frame, slice, macroblock, block size and block partition-level. For simplicity, we only focus on the frame-level and the levels below slice-level; in this way, the proposed framework aims to control the complexity at each frame, macroblock, and so on. We base on the fast motion estimation algorithm on the JM9.6 reference code, i.e. UMHexagonS. However, the proposed approach is general and independent of the choice of fast motion estimation algorithm.

2.1 Problem Statement

The problem is how to allocate available encoding time to the encoding units such that the rate-distortion loss is minimized subject to the given encoding time. The problem can be stated formally as,

$$\min_{\overline{X}} \sum \Delta J_{\overline{X}} = J_{\overline{X}} - J_{FME} = \Delta D_{\overline{X}} + \lambda \cdot \Delta R_{\overline{X}} \qquad subj.\ \sum t_{\overline{X}} \le T \quad , \tag{1}$$

where $\overline{X}$, J_{FME}, $t_{\overline{X}}$ and T represent the complexity allocation for each unit of interest, Lagrangian RD cost when fast motion estimation (FME) is fully used without any complexity constraint, encoding time spent for the each coding unit and encoding time budget, respectively. It is immediately apparent that it is difficult to estimate J_{FME} and therefore it is hard to reserve computational resource for optimal complexity allocation. An additional problem is that the encoding time is machine-dependent.

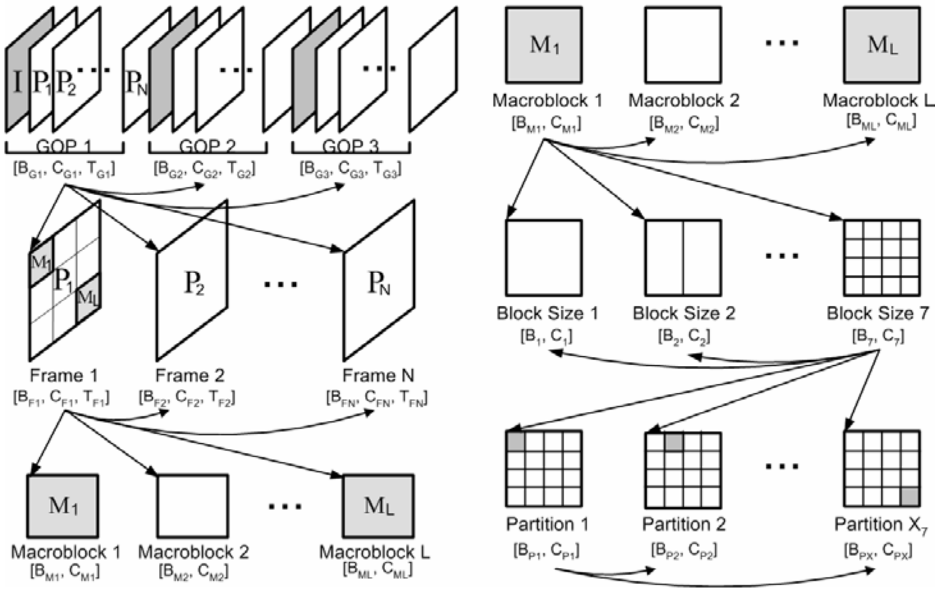

Figure 1. Overview of the proposed hierarchical complexity control for H.264/AVC.

In this paper, we propose the hierarchical complexity control algorithm for motion estimation as shown in Fig. 1. The GOP/frame level complexity budget $C_x, x \in [G_i, F_j], 1 \le i \le n_{GOP}, 1 \le j \le n_{Frame}$ is simply estimated using complexity budget that is spent on encoding previous GOP/frame for time T_x as described in Sec. 2.2. The budget B_x for remaining GOP/frame is updated accordingly. Basically, the encoder allocates estimated GOP level complexity budget to each frame and allocated frame budget is assigned to each macroblock and so on. The macroblock/ block-size/ partition level budget in current frame, $C_y, y \in [M_p, q, P_r], 1 \le p \le n_{GOP}, 1 \le q \le 7, 1 \le r \le n_{Partition}$, is initially estimated using J-C correlation between temporally adjacent macroblocks which will be described in Sec. 2.2. The initially allocated budget is adjusted within pre-defined threshold in order to minimize RD performance degradation during the motion estimation. The budgets for remaining macroblocks/block sizes/partitions are updated based on the budget spent on the current one. For simplicity, the encoder allocate macroblock level budget to each block size and partition uniformly in this work.

2.2 Optimal Complexity Allocation

Since the encoding time is machine dependent, it is desirable to transform the constraint into the machine independent measure – the number of searching points. The complexity constraint then becomes:

$$\sum_{i=1}^{L}\sum_{j=1}^{7}\sum_{k=1}^{x_j} \omega_{BS(k)} \cdot c_k^i \le C_{\max}, \tag{2}$$

where, $\omega_{BS(k)}$ represent weight for block partition whose block size is $BS(k)$ and c_k^i is the number of searching points for the partition k in i^{th} macroblock. The L, x_j and $C_{\max}$ are total number of macroblocks in frame, number of partition in block size j, and the assigned frame budget, respectively. The weight $\omega_{BS(k)}$ depends on the partition size since larger partitions will consume more time to calculate the motion cost. The frame-level complexity budget, $C_{\max}$, can easily be calculated for each frame as follows.

(1) First frame (IDR frame)

(2) Second frame (First predicted frame): FME is performed without any complexity constraints. The number of search points (C_{ref}) and frame encoding time (T_{ref}) for motion estimation are measured.

(3) Third frame (Second predicted frame) and after: $C_{\max}$ can be calculated as $C_{\max} = (T / T_{ref}) \times C_{ref}$, where $T = 1/ \mathit{frame rate}$.

The frame-level complexity budget can be calculated at every IDR frame or GOP. Through the transformation of constraint, the problem can be re-written as,

$$\min_{\overline{X}} \sum \Delta J_{\overline{X}}\ , \quad subj.\ \sum c_{\overline{X}} \le C_{\max}, \tag{3}$$

The conventional way of solving this optimization problem can be summarized as follows [12].

STEP 1. J-C Curve Estimation: To solve (3), it is necessary to estimate a piecewise linear $J - c_{\overline{X}}$ curve (J-C curve) of the current encoding units, which are macroblocks in this work. The estimation of the J-C curve can be done through a model that is based on global statistics and/or encoding parameters such as quantization parameter (QP). In practice, the J-C curve of a temporally adjacent encoded frame may simply be used. The J-C pair is measured at each iteration, which is defined as the number of searching points used for one block size in one reference frame. The granularity of iteration is determined to provide good tradeoff in overall encoding time and RD performance. Note that only the iteration that yields lower J is recorded in the curve.

STEP 2. Convex Hull of J-C Curve: Based on the estimated piecewise linear J-C curve, a convex hull is constructed using fast convex hull search algorithm [13].

STEP 3. Optimal Complexity Allocation: The optimal complexity assignment can be performed by using greedy search or dynamic programming (Viterbi algorithm) after the problem is converted to an unconstrained optimization problem using Lagrange multiplier [12].

3. PROPOSED APPROACH

In this section, our proposed hierarchical complexity control (HCC) scheme based on J-C slope is introduced in detail. Basically, HCC provides scalability in encoder complexity at each level of encoding hierarchy. Therefore, it enables encoder to reduce complexity to the desired level with given motion estimation algorithm. At the same time, HCC is designed to minimize the expected RD performance degradation by employing J-C slope based allocation. Also, J-C slope mismatch problem is efficiently addressed through the partition-level budget adjustment and update. From the experimental results of J-C curves, we observed that the majority of macroblocks in background or smoothly moving objects have piecewise linear J-C curves rather than the cup-convex shape. On the other hand, macroblocks of complex motion show cup-convex J-C curve in general. Intuitively, complex motion in detail areas usually results in high cost at large block size and the cost converges to a lower level as the block size gets smaller. On the contrary, smooth motion areas quickly converge to a larger block size in general. Also, we observed that there is strong correlation between J-C curves of the current macroblock and its temporally collocated macroblock; therefore the J-C curves of current macroblock are estimated from the J-C curves of collocated macroblock in the previous frame. It is noted that each estimated J-C curve may have N slopes. N is chosen as 2 in this work so that each curve is approximated using at most two lines. This choice of N is a result of tradeoff between budget allocation complexity and budget allocation accuracy. The proposed macroblock level complexity control algorithm can be summarized as follows (frame layer allocation still follows the descriptions of the previous section).

STEP 1. Initial Budget Allocation based on J-C Slopes: First, a minimum budget is assigned to all macroblocks in order to prevent a block partition with a zero-value budget. In this work, a single point for each block partition is chosen as the minimum budget and each point is assigned to predicted motion vector at each block partition. Based on the piecewise linear J-C curve, an initial budget is allocated for each macroblock using a greedy search until the budget is exhausted such as;

(1) J-C Curve Approximation: Each macroblock in the previous frame can have $n_i \le N$ iterations so it can have n_i slopes, $S_i^k = |\Delta J_i^k / \Delta C_i^k|$, where $i = [1, \cdots, M], k = [1, \cdots, n_i]$ and i, k and M are macroblock index, iteration index and the total number of macroblocks in frame, respectively. Each slope indicates potential in RD coding gain improvement at each iteration.

(2) Convex Hull of J-C Curve: A convex hull is constructed for each macroblock by using simple and fast algorithm $O(n \log n)$ [13].

(3) Sub-optimal Complexity Allocation: Since the estimated piecewise linear J-C curve is convex, a greedy search can be used to allocate searching points based on slope S_i^k. The remaining budget, which is the total budget subtracted by minimum budget for each macroblock, is allocated such as;
   a. Construct initial slope list for all macroblocks such as $L = [S_1^k, \cdots, S_M^k],\ k = 1$.
   b. Compare the slope of each macroblocks $S_i^k = |\Delta J_i^k / \Delta C_i^k|$ in the list and assign $C_i^{ini} = \Delta C_i^k$ to the macroblock that has maximum slope.
   c. Increase iteration index k of the selected i^{th} macroblock such as $S_i^{k'=k+1}$, $i = \max_l |\Delta J_l^k / \Delta C_l^k|$ and update the slope list accordingly.
   d. Repeat the step b. and c. until either the all slope indices indicate last index n or the frame budget $C_{\max}$ is exhausted.
   e. If the budget is left, then the remaining budget is redistributed into the all macroblocks depending on the estimated need which is initially allocated amount such as,

$$\dot{C}_i^{ini} = C_i^{ini} \cdot \left(1 + \frac{\left(C_{\max} - \sum_{j=1}^{M} C_j^{ini} \right)}{\sum_{j=1}^{M} C_j^{ini}} \right) \tag{5}$$

In case of N=1, a simple slope-weighted allocation using Eq. (6) can be applied for initial allocation.

$$C_i^{ini} = \frac{C_{\max} \cdot S_i}{\sum_{j=1}^{M} S_j} \tag{6}$$

<u>STEP 2. Complexity Budget Adjustment:</u> Perform the motion estimation in each macroblock with initially assigned complexity budget, where the J-C slope S_i^k is measured at each k^{th} iteration. Even though temporally adjacent J-C curves are correlated, it is important to note that the J-C curve of current macroblock can have different slopes than the collocated macroblock which was used for initial allocation. Also, the J-C slope S_i^k of current macroblock is available when motion estimation is completed for the k^{th} iteration. In order to dynamically compensate the impact of J-C curve mismatch, a small number of searching points for additional iterations are conditionally allowed: $C_i = C_i^{ini} + \Delta$. Intuitively, extra searching points can be assigned as long as the new slope is greater than the previous slope $S_i^{k+1} \ge S_i^k$, $k \ge n_i$ until the threshold Δ is met.

<u>STEP 3. Complexity Budget Update:</u> After the motion estimation of current macroblock is completed, the budget of the remaining macroblocks is updated according to the actual amount used by the current macroblock and their initial values. Under the macroblock level, the budget is uniformly distributed for each block size and partition successively.

4. EXPERIMENTAL RESULTS

In the experiment, the proposed algorithm is integrated with the JVT reference software JM9.6. The proposed algorithm is tested for a wide range of bit rates as shown in Fig. 1. 100 frames of each sequence are encoded at 30 fps using low complexity encoding mode. Without loss of generality, we evaluate our complexity control algorithm for integer resolution motion estimation for a single reference frame in this work. To compare the RD performance and the computational complexity of the proposed scheme with those of the fast motion estimation (UMHexagonS) in H.264/AVC reference software, the PSNR and the bit rate are measured. For computational complexity profiling, the encoding time of the motion estimation module is measured.

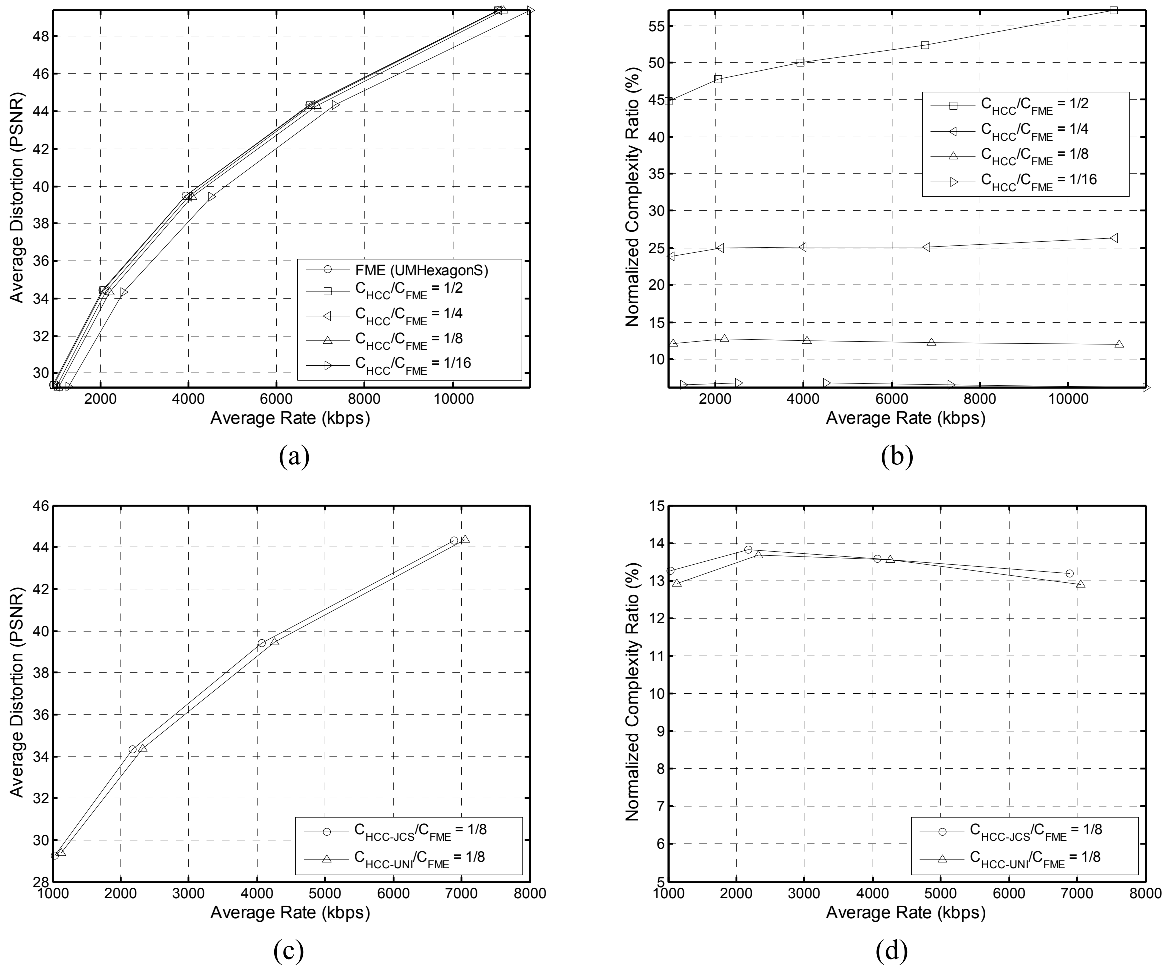

Figure 2. (a) R-D performance and (b) computational complexity for Stefan (CIF) as the frame budget decreases. Comparison of (c) RD performance and (d) computational complexity for two different complexity allocation schemes, macroblock-level JC slope based allocation and uniform allocation.

As shown in Fig. 2-(a,b), the proposed algorithm can reduce motion estimation time by 3/4 with small RD performance loss. It is important to note that the RD performance considerably degrades when the searching point ratio between the proposed algorithm and fast motion estimation ($C_{HCC} / C_{\mathrm{FME}}$) is below 1/8. The result can be explained by the relationship between Lagrangian RD cost and complexity. Even though we reduce the complexity by constraining the allocated number of searching points up to certain amount, RD performance loss is insignificant as long as slope in JD curve is small. As the slope becomes sharply increased, the RD performance degraded quickly with further complexity reduction. Let us call this point as critical point. In case of Stefan sequence, the critical point is observed at a complexity control ration of approximately 1/8. Also, it is worth to note that J-C slope based allocation is better than uniform allocation in terms of RDC tradeoff as shown in Fig. 1-(c,d). In uniform allocation, the complexity is evenly allocated to macroblocks.

CONCLUDING REMARKS

A hierarchical complexity control algorithm is proposed to reduce the complexity of H.264 motion estimation. The main idea is to limit the complexity of motion estimation based on the expected RD coding gain loss. In order to efficiently reduce the complexity to desired level, HCC is designed to provide complexity scalability in motion estimation so as to provide flexible tradeoff between video quality and computational complexity. With the proposed algorithm, we demonstrated that complexity of motion estimation can be reduced by 3/4 without significant RD performance degradation.

REFERENCES

1. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Transactions on Circuits, System and Video Technology, vol. 13, no. 7, pp. 560-576, 2003.
2. 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 and Systems Magazine, vol. 4, no. 1, pp. 7–28, First Quarter, 2004.
3. ITU-T Standardization Sector, “Advanced video coding for generic audiovisual services,” ITU-T Rec. H.264/ISO/IEC 14496-10:2005(E), March 2005.
4. Jianfeng Xu, Zhibo Chen, Yun He, “Efficient fast ME predictions and early-termination strategy based on H.264 statistical characters,” Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Vol. 1, Dec. 2003.
5. Z. Zhou, M-T. Sun, Y-F. Hsu, “Fast variable block-size motion estimation algorithm based on merge and slit procedures for H.264/MPEG-4 AVC,” Proceedings of the 2004 International Symposium on Circuits and Systems, Vo. 3, May 2004.
6. A.M. Tourapis, O.C. Au, M.L. Liou, “Highly efficient predictive zonal algorithms for fast block-matching motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, Issue 10, Oct. 2002.
7. Jianfeng Xu, Zhibo Chen, Yun He, “Efficient fast ME predictions and early-termination strategy based on H.264 statistical characters,” Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Vol. 1, Dec. 2003.
8. M. Yang, H. Cui, K. Tang, “Efficient tree structured motion estimation using successive elimination,” IEE Proceedings-Vision, Image and Signal Processing, Vol. 151, Issue 5, Oct. 2004.
9. Chi-Wang Ting, Hong Lam, Lai-Man Po, “Fast block-matching motion estimation via recent-biased search for multiple reference frames,” International Conference on Image Processing, Vol. 4, Oct. 2004.
10. M-J. Chen, Y-Y. Chiang, H-J Li, M-C. Chi, “Efficient multi-frame motion estimation algorithms for MPEG-4 AVC/JVT/H.264,” Proceedings of the 2004 International Symposium on Circuits and Systems, Volume 3, May 2004.
11. P. Yin, H. C. Tourapis, A.M. Tourapis, J. Boyce, “Fast mode decision and motion estimation for JVT/H.264,” IEEE International Conference on Image Processing, Vol. 3, Sept. 2003.
12. G. M. Schuster, G. Melnikov, A. K. Katsaggelos, “A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers,” IEEE Transactions on Multimedia, vol. 1, no. 1, pp. 3 - 17, March 1999.
13. K. L. Clarkson and P. W. Shor, "Applications of Random Sampling in Computational Geometry, II," Discrete and Computational Geometry, vol. 4, no. 1, pp. 387-421, 1989.