COMPLEXITY ADAPTIVE H.264 ENCODING USING MULTIPLE REFERENCE FRAMES

Sui-Yuk LAM, Oscar. C. Au, Peter H. W. Wong

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{sylace, eeau, eepeter}@ust.hk

ABSTRACT

The state-of-the-art H.264/AVC video coding standard achieves significant improvements in coding efficiency by introducing many new coding techniques. However the computation complexity is inevitably increased during both the encoding and decoding process. Many previous works, such as fast motion estimation and fast mode decision algorithms, have been proposed aiming at reducing the encoder complexity while maintaining the coding efficiency. In this paper, we propose a new encoding approach which accounts for the decoding complexity. Simulation results show that the decoding complexity can be reduced by up to 15% in terms of motion compensation operations, which is the most complex part of the decoder, while maintaining the R-D performance with only about 0.1dB degradation.

***Index Terms*—** AVC, Subpixel, Complexity

1. INTRODUCTION

The H.264/AVC is the latest video coding standard jointly developed by the ITU-T and the ISO/IEC MPEG. By introducing many new coding techniques, higher coding efficiency can be achieved at the expense of much higher computational complexity. Techniques such as variable block size and quarter-pixel motion estimation increase the encoding complexity enormously, while the decoding complexity is significantly increased due to operations such as 6-tap subpixel filtering and deblocking. Many previous works were focusing at reducing the encoding complexity with negligible coding efficiency degradation. Various fast motion estimation [1] and mode decision [2] algorithms were developed. Recently, parallel processing techniques that utilizing the hardware architecture[3], such as Intel MMX / SSE / SSE2 [4] and GPU, further reduced the time requirement for the encoding process. However little attention was paid on the decoder side.

In this paper, the complexity of the H.264/AVC decoder is focused instead of the encoder. Motivated by the rapid growing market of embedded devices, algorithmic solution is investigated as the hardware configuration can be very different at user-end. Y. Wang proposed a rate-distortion-complexity (RDC) optimization framework, named CAMED [5], which claims up to 60% saving of subpixel interpolation operations with about 0.2dB loss in PSNR. However we observed that such scheme may results in non-smooth motion field due to direct modification made to the motion vectors, and thus more overheads for coding motion vectors, which is not desirable, especially in low bit-rate situations. Also, result of 60% saving is calculated according to some inaccurate assumptions. In our work, similiar joint R-D-C optimization framework has been adopted with some critical modifications to preserve the true motion information. A new complexity model has also been developed accordingly.

The rest of this paper is organized as follows. Section 2 provides an overview on H.264/AVC motion compensation. Details of the proposed algorithm are given in Section 3. Simulation results are reported and explained in Section 4. Section 5 concludes this paper.

2. FRACTIONAL MOTION ESTIMATION & COMPENSATION

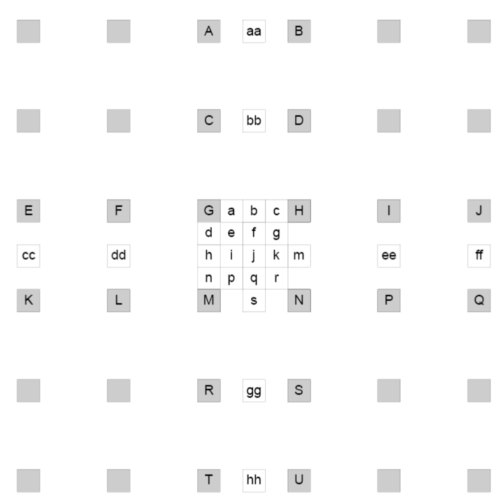


Fig. 1. Notations for Integer samples and fractional sample positions in H.264/AVC

Quarter pixel motion vector accuracy further improves the coding efficiency of H.264/AVC comparing with prior video coding standards by allowing more accurate motion estimation. Figure 1 visualizes the subpixel positions. The half-pixel values are derived by applying a 6-tap filter with tap values [1 -5 20 20 -5 1] and quarter-pixel values are derived by averaging the the sample values at full and half sample positions during the motion compensation process. For example, the predicted value at the half-pixel position b is calculated with reference to Fig. 1 as:

$b_1 = E - 5 * F + 20 * G + 20 * H - 5 * I + J$

$b = Clip((b_1 + 16) >> 5)$

Details of the interpolation process are given in the standard [6] and is therefore not mentioned here. For any non-integer pixel locations, the computational complexity is much higher comparing with the integer pixel positions due to extra complex multiplication and clipping operations. In General Purpose Processor (GPP), such operations usually consume much more clock cycles than other instructions and therefore the decoder complexity is dramatically increased. To address this problem, the complexity cost should be considered during the motion estimation step to avoid any unnecessary interpolations. Instead of choosing the motion vector with optimal R-D performance, the sub-optimal motion vector with lower complexity cost is selected. An efficient encoding scheme is therefore necessary to balance between the coding efficiency and decoding complexity.

3. THE PROPOSED COMPLEXITY ADAPTIVE ENCODING SCHEME

3.1. Rate-Distortion Optimization

The rate-distortion optimization framework [7] is commonly adopted in lossy video coding to improve the coding efficiency. The basic idea is to minimize the distortion D subject to a rate constraint. The Lagrangian multiplier method is a common approach. The motion vector which minimize the R-D cost is selected according to the following:

$$J^{R,D}_{Motion} = D_{DFD} + \lambda_{Motion} R_{Motion} \tag{1}$$

where $J^{R,D}_{Motion}$ is the joint R-D cost, D_{DFD} is the displaced frame difference between the input and the motion compensated prediction, and R_{Motion} is the estimated bit-rate associated with the selected motion vector. Similarly, the joint R-D cost for mode decision is given by:

$$J^{R,D}_{Mode} = D_{Rec} + \lambda_{Mode} R_{Mode} \tag{2}$$

The value of λ_{Mode} is determined empirically. The relationship between λ_{Motion} and λ_{Mode} is adjusted according to:

$$\lambda_{Motion} = \sqrt{\lambda_{Mode}} \tag{3}$$

Location (quarter-pel accuracy)	Notation	Cost
(0, 0)	G	0
(0, 2) (2, 0)	b, h	1*6-tap
(0, 1) (1, 0) (0, 3) (3, 0)	a, c, d, n	1*6-tap, 1*2-tap
(1, 1) (1, 3) (3, 1) (3, 3)	e, g, p, r	2*6-tap, 1*2-tap
(2, 2)	j	7*6-tap
(2, 1) (1, 2) (3, 2) (2, 3)	i, f, k, q	7*6-tap, 1*2-tap

Table 1. Subpixel pixel locations and the associated interpolation complexity

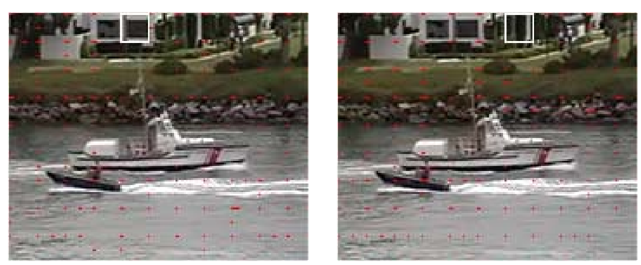
if SAD and SSD are used during the motion estimation and mode decision stage respectively [8].

3.2. The Proposed Rate-Distortion-Complexity Optimization

In [5] Wang proposed the joint R-D-C optimization framework for sub-pixel refinement. The complexity cost for each sub-pixel location is accounted into the joint RDC cost function, which is given by:

$$J^{R,D,C}_{Motion} = J^{R,D}_{Motion} + \lambda_C C_{Motion} \tag{4}$$

The joint RDC cost is minimized during the subpixel motion estimation stage. When $\lambda_C = 0$, the importance of complexity factor is neglected and the conventional R-D optimization framework is retained. The complexity cost C_{Motion} is determined by the theoretical computational complexity of the obtained motion vector based on Table 1.



(a) Original (b) Proposed Method

Fig. 2. Visualization of the resultant motion field

Although such optimization framework is optimal locally, the resultant sub-optimal motion vectors may unfavour the overall coding efficiency. Such effect is especially significant in low bitrate situations that motion vector cost dominate over the residue cost.

To avoid the motion field artifacts generated by this framework, we propose a new multiple reference frames technique. The objective for the proposed method is to preserve the correctness of the motion vectors. In this method, the joint RDC cost is minimized within the selection of the best reference index:

$$Ref = \arg \min_{refidx} \{J^{R,D}_{Motion}(V_{refidx}) + \lambda_C C_{Motion}(V_{refidx})\} \quad (5)$$

where V_{refidx} refers to the R-D optimized motion vector with reference index $refidx$ and Ref is the optimal reference index. The joint RDC optimization framework is applied along the reference index selection process instead of the subpixel estimation process such that the motion vectors can always represent the true motion, assuming that motion estimation succeed. For example, for some video contents with constant object motion of half pixel displacement to the left for each frame, coding as $\{(4, 0) : 1\}$ instead of $\{(2, 0) : 0\}$ can represent the real motion information while reducing the interpolation complexity (number in bracket represents the x and y component of the motion vector respectively and the remaining refers to the reference index). Figure 2 visualizes the motion vectors with the proposed method, which shows a smooth region at the top-left region with motion vectors with greater magnitude but lower interpolation complexity. Hence chaotic motion field generated by sub-optimal motion vectors can be avoided.

A new complexity cost model is developed. According to Table 1, interpolating position j requires 7 6-tap operations, but it takes only

$(6 + w - 1) * h + w * h$

6-tap operations for a block with width w and height h, that is, 52 operations for a 4x4 block for example and therefore, on average 3.25 6-tap for each pixel. Therefore the new estimated complexity cost is given by:

$$C^{'} = \begin{bmatrix} 1 & 12 & 10 & 12 \\ 12 & 24 & 39 & 24 \\ 10 & 39 & 35 & 39 \\ 12 & 24 & 39 & 24 \end{bmatrix} \quad (6)$$

$$C_{Motion}(MV_x, MV_y) = C^{'}_{MV_x\&3, MV_y\&3} \quad (7)$$

where the operator $\&$ refers to bitwise AND operation. Adjustments are made accounting for the complexity cost of the addition and shifting operations and further adjustments are made according to the current block mode.

The lagrangian multiplier λ_C is derived experimentally according to the assumption made by the proposed algorithm and shows the following relationship:

$$ln(\lambda_C) = K - D_{DFD} \quad (8)$$

where K is a constant that characterizing the video context. Such relationship has been verified for various sequences with different quality as shown in Fig. 3. The value for K is determined to be around 20 empirically. To explain this, large λ_C values will obviously degrade the R-D performance while small values may result in sudden change in reference frame selection and hence higher motion vector cost.

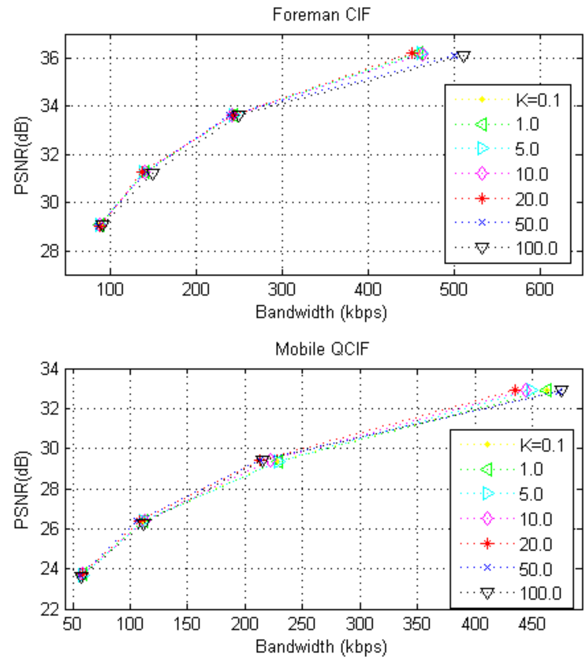


Fig. 3. R-D Performance of various value of K

Profile	Baseline (CAVLC)
Intra Period	0 (except first)
Motion Estimation	Full Search
Search Range	16
Hadamard	used
Block Mode	16x16, 16x8, 8x16
Number of Reference Frames	4
Frame Rate	30

Table 2. List of simulation parameters

4. EXPERIMENT RESULTS

The objective of the simulations is to demonstrate the usefulness of the proposed multiple reference frames complexity optimization technique. The R-D-C performance of the proposed scheme is compared with the original R-D optimization framework only as information given in [5] is limited with many unknown threshold values and the complexity assumption is not accurate. Results were simulated using the H.264 reference software JM8.6 with encoding parameters given in Table 2.

Figure 4 shows the comparison of the R-D performance between the proposed algorithm and the original full-search method for a few testing sequences. Generally the performance degradation is only around 0.1 dB and even lower for low bit-rate situations. Depends on the bit-rate and the motion characteristics, complexity saving for decoding varies from around 5% to 20%, as shown in Figure 5. The figure shows similar characteristics as [5] that saving is more significant in high bit-rate, since the motion vectors accuracy is relatively higher in high bit-rate and therefore distributed more uniformly over the subpixel locations, as shown in Figure 6 (Position (0, 0) refers to integer pixel location G, as given in Table1). For many of the testing sequences, the video contents consist of stationary background and therefore motion vectors are biased at the $(0, 0)$ position and therefore room for improvement for further complexity saving is limited. Such effect is demonstrated by *City* in Figure 5 with its relatively higher complexity saving as global motions dominate.

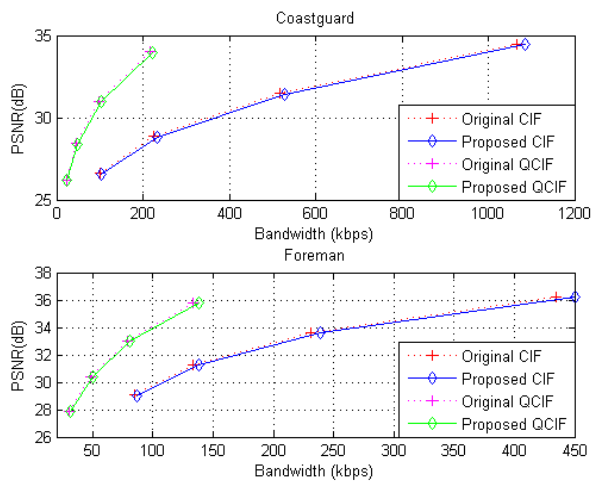


Fig. 4. R-D Performance of the proposed method

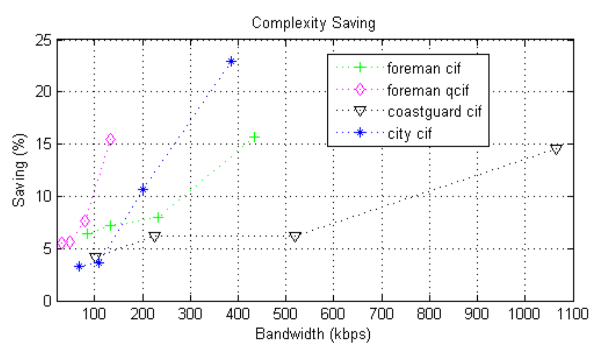


Fig. 5. Complexity saving using the proposed method, in terms of interpolation operation

5. CONCLUSION

In this paper we propose a complexity adaptive encoding algorithm using the optimal reference selection technique which shows reasonable decoding complexity saving. Full-search was used in our experiments to demonstrate the usefulness

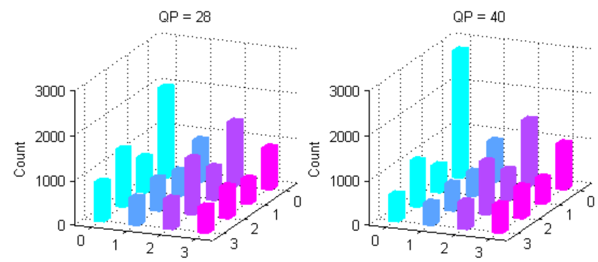


Fig. 6. Original MVs Distribution in *City*

of the proposed technique. We believe that combining such technique with some fast motion estimation algorithms will be able to achieve both lower encoding and decoding complexity with some reference frame biasing techniques.

6. REFERENCES

[1] A. M. Tourapis, O. C. Au, and M. L. Liou, "Fast Block-Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)," MPEG2000/M5866, 2000.

[2] Y. K. Tu, J. F. Yang, Y. N. Shen, and M. T. Sun, "Fast Variable Size Block Motion Estimation using Merging Procedure with an Adaptive Threshold," in *Proc. ICME*. IEEE, 2003, vol. 2, pp. 789–792.

[3] W. N. Lie, H. C. Yeh, T.C.-l, and C. F. Chen, "Hardware-efficient Computing Architecture for Motion Compensation Interpolation in H.264 Video Coding," in *Proceeding of Circuits and Systems*. IEEE, May 2005, vol. 3, pp. 2136–2139.

[4] J. Lee, S. Moon, and W. Sung, "H.264 Decoder Optimization Exploiting SIMD Instructions," in *Proceedings of Circuits and Systems*. IEEE, Dec. 2004, vol. 2.

[5] Y. Wang and Shin-Fu Chang, "Complexity Adaptive H.264 Encoding for Light Weight Streams," in *Proceeding of ICASSP*. IEEE, May 2006, vol. II.

[6] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," JVT G050, 2003.

[7] G. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[8] T. Weigand and B. Girod, "Parameter Selection in Lagrangian Hybrid Video Coder Control," in *Proc. ICIP 01*, Thessaloniki, Greece, 2001, IEEE.