ADAPTIVE DECODER COMPLEXITY REDUCTION FOR COARSE GRANULAR SCALABILITY

He Li [1], *Z. G. Li* [2] *and Changyun Wen* [1]

[1] School of EEE, Nanyang Technological University, Singapore 638798
[2] Media Division, Institute for Infocomm Research, Singapore 119613

ABSTRACT

The on-going scalable video coding (SVC) standard is an extension of H.264/AVC. It enables significantly improved compression performance at the expense of greatly increased computational complexity at both the encoder and the decoder sides. This paper presents an adaptive algorithm to reduce the complexity of decoder at the encoder side for coarse granular scalability. Hence, lightweight bitstreams are generated at the encoder which requires significantly less decoding complexity. The experimental results show that the proposed scheme provides significant reduction in the complexity of decoder with acceptable coding loss and minor impact on the encoder complexity.

***Index Terms*—** Scalable video coding, decoder complexity reduction, coarse grain SNR scalability

1. INTRODUCTION

Scalable video coding as proposed in [1] is an extension of H.264/AVC. Among the scalabilites available in SVC, coarse granular scalability (CGS) is one of the most important. The CGS scalability can be realized by a layered approach involving a base layer and several enhancement layers. The base layer contains a reduced quality version of each coded frame. The enhancement layers are coded based on predictions formed from the base layer pictures and previously encoded enhancement layer pictures. Current CGS scheme shows significant achievements in terms of coding efficiency by using fractional (1/4) pixel precision, adaptive block size motion compensation, loop filter, 4x4 integer transform, etc. However, the excellent performance of CGS is achieved at the expense of computational complexity at both the encoder and the decoder sides. The computational capacity of devices, especially handhold devices, is usually very limited. It is desirable to obtain a good tradeoff between complexity and compression for the CGS.

In order to address this problem, a number of efforts have been made to alleviate the SVC encoder complexity due to mode decision [3], while maintaining visual quality and other coding efficiency. However, there is no result on the complexity reduction of a CGS decoder although this problem is more important than the complexity reduction of an SVC encoder in many applications such as DVD players, digital TV receivers etc [4]. It should be pointed out that a number of efforts have been made to explore decoder complexity reduction in H.264/AVC by using the observation that image interpolation and deblocking filter consume the majority of the decoding time. Ugur et al. proposed a decoder complexity reduction by biasing easy-to-decode motion vectors in a rate distortion optimized fashion [4]. An effort has also been made by Wang et al. to effectively reduce the decoder-side computational cost required at the H.264 decoder [5]. All these methods are efficient in reducing the computational complexity with acceptable quality degradation in H.264 decoder. However, these methods are not applicable to all the layers in a CGS system. Moreover, they did not provide any formulation to choose the Lagrangian factor associated with the complexity issue.

In this paper, we propose an adaptive encoding algorithm that would generate lightweight bitstreams which require less amount of sub-pixel interpolations at the decoder. Moreover, our proposed algorithm is adaptive to all the layers in SVC. This is achieved by using new rate-distortion-complexity optimized cost functions for motion estimation and mode decision as compared to traditional encoding algorithms. The experimental results show that the proposed method can reduce the decoding complexity effectively and flexibly with acceptable performance degradation. The rest of this paper is organized as follows. In Section 2, a new framework of rate-distortion-complexity optimization is formulated. In Section 3, a complexity model for the interpolation is provided. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2. RATE-DISTORTION-COMPLEXITY OPTIMIZATION

2.1. Problem Formulation

In SVC, variable block-size based motion estimation/motion compensation is used to reduce the temporal redundancy among frames in the same layer and the spatial redundancy among frames in different layers. SVC defines 7 macroblock (MB) modes for inter prediction (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4), 2 macroblock modes for intra prediction (4×4 and 16×16) and $SKIP/Direct$. For encoding the motion field of an enhancement layer, "Base_layer_mode" and "Qpel_refinement_mode", are added to the modes applicable in the base layer. These two modes indicate that motion and prediction information including the partitioning of the corresponding MB of the base layer is used. In order to select the best mode for each MB, the encoder exhausts all possible modes in the rate-distortion optimization (RDO) framework [1]. As discussed in previous section, it is desirable to design a new framework to obtain a good trade off between complexity and compression. Here, such a frame work called rate-distortion-complexity optimization (RDCO) is formulated as

$$\begin{aligned} &\min D, \\ &\text{subject to } R < R_c, C < C_c, \end{aligned} \tag{1}$$

where D, R and C stand for the distortion, rate and complexity. And R_c and C_c stand for the rate and complexity constraint.

This problem appears in both motion estimation and mode decision stages, and can be solved by the conventional Lagrangian optimization method as below.

1) In motion estimation stage, searching for the best motion vector can be viewed as the minimization of the Lagrangian cost function:

$$\begin{aligned} &J(MV|QP, \lambda_{SAD}, \lambda_{MOTION}) \\ &= SAD(MV|QP) + \lambda_{SAD} R(MV|QP) \\ &+\lambda_{MOTION} C_{MOTION} \end{aligned} \tag{2}$$

where C_{MOTION} is the interpolation complexity assigned to each motion vector (MV), λ_{SAD} and λ_{MOTION} are two Lagrangian multipliers, SAD is the sum of absolute difference of Hadamard-transform coefficients, R represents the number of bits that would be used to code the MV, QP represents the quantization parameter settings.

2) RDCO mode decision refers to the minimization of the following Lagrangian function:

$$\begin{aligned} &J(MODE|QP, \lambda_{SSD}, \lambda_{MODE}) \\ &= SSD(MODE|QP) + \lambda_{SSD} R(MODE|QP) \\ &+\lambda_{MODE} C_{MODE} \end{aligned} \tag{3}$$

where λ_{MODE} and λ_{SSD} are two Lagrangian factors, SSD is the sum of the squared differences between the original MB and the reconstructed MB located in the reference frames, R denotes the bit cost for encoding the motion vectors, MB header and all the residual information, and C_{MODE} is the sum of all the interpolation complexities for all motion vectors involved in the candidate mode, i.e.

$$C_{MODE} = \sum_{n=1}^{N} C_{MOTION}(n) \tag{4}$$

with N representing total number of motion vectors of the concerned MB.

2.2. Choices of λ_{MODE} and λ_{MOTION} Based on λ_{SAD}

To apply (2) and (3) to an SVC encoder, it is required to properly determine the values of λ_{MODE} and λ_{MOTION}. The Lagrangian multipliers λ_{MODE} and λ_{MOTION} are usually defined as constant values [4]. However, the optimal choice of λ_{MODE} and λ_{MOTION} should depend on the QP. Therefore, the algorithm for the complexity-constrained motion estimation can be modified in order to incorporate macroblock quantization step-size changes.

In order to examine the relationship between the QP and λ_{MOTION}, a series of experiments on various sequences were set up. We first fix the value of λ_{MODE}. As a result, a particular setting of λ_{MOTION} and QP yields a minimum Lagrangian cost function in Eq.2. In our experiments, the QP is varied over several values, 10, 15, 20, 25, 30, 35, 40. For each QP, BDPSNR [6] is calculated for each possible value of λ_{MOTION} ranging from 10 to 200. Finally, the value of λ_{MOTION} is selected which results in nearly 0.05dB BDPSNR drop. Hence, we have the best quality and interpolation complexity trade off.

Note that the Lagrangian multiplier λ_{SAD} is closely related to QP in the following equation [1]:

$$\lambda_{SAD} = 0.92 * 2^{QP/6-2} \tag{5}$$

For each QP value, λ_{SAD} can be computed from the above equation. Since λ_{SAD} is known in the process of computing Eq.2, in the following stage, we only need to examine the relationship between λ_{SAD} and λ_{MOTION}. Typical experimental results are plotted in Fig.1, where the solid curve shows the relationship between λ_{SAD} and λ_{MOTION} from the experimental results.

Based on the curve, we use data fitting technique to generate the following function, which relates λ_{MOTION} to λ_{SAD} as following:

$$\lambda_{MOTION} = K_{MOTION} * ln(\lambda_{SAD} + 1) \tag{6}$$

The function is also plotted as dash curve in Fig.1. From this figure, we can see that our proposed function is sufficient to accurately fit the original experimental data. By adjusting the value of K, we can achieve the interpolation complexity and quality trade off at motion estimation stage. Similarly, for mode decision, we have:

$$\lambda_{MODE} = K_{MODE} * ln(\lambda_{SSD} + 1) \tag{7}$$

For each video sequence, an optimal set of K_{MOTION} and K_{MODE} can be generated to optimize the trade-off between the interpolation complexity and video quality. However, based on extensive experiments on various test sequences, the value of K_{MOTION} and K_{MODE} are not sensitive to different video sequences. Therefore, in our experiments, they are set to be 45 and 1 for all the test sequences.

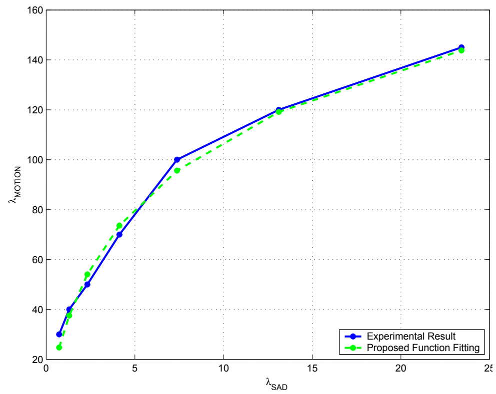

Fig. 1. λ_{SAD} versus λ_{MOTION} for FOREMAN sequence

3. INTERPOLATION COMPLEXITY MODELLING

Sub-pixel motion estimation and compensation involves searching sub-sample interpolated positions as well as integer-sample positions, choosing the position that gives the best match and using the integer- or sub-sample values at this position for motion compensated prediction. Fig.2 shows the concept of a quarter-pixel motion estimation. There are 15 possible sub-sample positions, including three half-pixel positions (at location 3, 9, and 11) and twelve quarter-pixel positions. Each of the sub-pixel position requires a different interpolation filter.

It is obvious that the location of a motion vector influences the decoder interpolation complexity directly. We can assign the interpolation complexity term to each MV location as follows:

1) If the motion vector has integer values for both its horizontal and vertical directions, the decoder does not perform any interpolation at all, hence the interpolation cost is 0.

2) Otherwise, if the motion vector has integer value for either horizontal or vertical direction, the decoder needs to perform at least one 6-tap interpolation. The interpolation cost is set to 1 in the proposed scheme.

3) Otherwise, if the motion vector is at location 6, 8, 14 and 16, the decoder needs to perform two 6-tap interpolation and one 2-tap interpolation. Hence, the interpolation cost is set to 2 in the proposed scheme.

4) Otherwise, if the motion vector is at location 7, 10, 11, 12 and 15, the decoder needs to perform at least seven 6-tap interpolation. Hence, the interpolation cost is set to 4 in the proposed scheme.

X	2	3	4	X
5	6	7	8	
9	10	11	12	
13	14	15	16	
X				X

Fig. 2. Sub-pixel locations in SVC

4. SIMULATION RESULTS

The proposed decoder complexity reduction algorithm is embedded in JSVM 2.0 encoder [1]. The test platform used is Intel Pentium IV, 1.83GHz CPU, 256M RAM with Windows XP professional operating system. The test condition is shown in Table I. In our experiments, five standard test sequences including FOREMAN, FOOTBALL, BUS, CREW and CITY have been tested. We only consider the two-layer case and the QP value settings for all the layers are shown in Table I. The GOP size is set to be 8.

Table 1. Testing condition

		Tested Video Sequences
Frame Rate		15 Hz
Resolution		QCIF
QP Setting	Base	40
	Enhance	30 to 10
Coding Option Used		MV search range is ±32 pels. Reference frame number is 1. Full search is used in ME.
Codec		JSVM 2.0 encoder

The testing parameters in our experiments include the complexity saving, Y-PSNR and bit rate for both base layer and enhancement layer. We use Complexity Saving (CS) to indicate the total interpolation computational cost saving in decoding process:

$$CS = \frac{C_{JSVM} - C_{proposed}}{C_{JSVM}} \times 100\% \qquad (8)$$

where C_{JSVM} and $C_{proposed}$ are the total interpolation computational cost of JSVM 2.0 system and its modified system according to the proposed algorithm, respectively.

Let QP_e stand for QP values at the enhancement layer. Table 2 to 4 show the tabulated performance comparison of the proposed algorithm with JSVM 2.0 for different QP_e. Note that in the table positive values mean increments, and negative values mean decrements. The results show that the proposed method is very effective in reducing the decoding complexity, especially for the sequence with high motion and fine detail. The average decoding complexity is reduced up to 61%. Fig. 3 and 4 present the rate distortion curves for FOREMAN and FOOTBALL. From these figures, we can conclude that our scheme can achieve consistent decoding complexity saving over a large bit rate range with negligible loss in PSNR and increments in bit rate.

5. CONCLUSION

In this paper, an adaptive algorithm has been proposed to reduce the complexity of decoder for SNR scalable video coding is presented. By introducing the complexity term in RDO, the motion interpolation that results in high complexity will

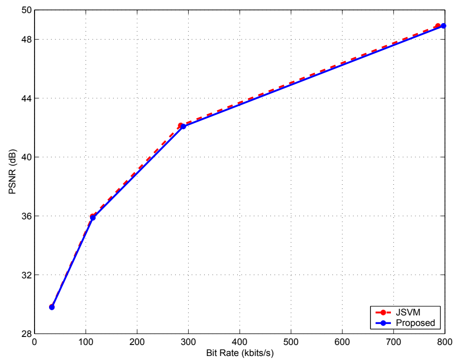

Fig. 3. Rate distortion curve for FOREMAN

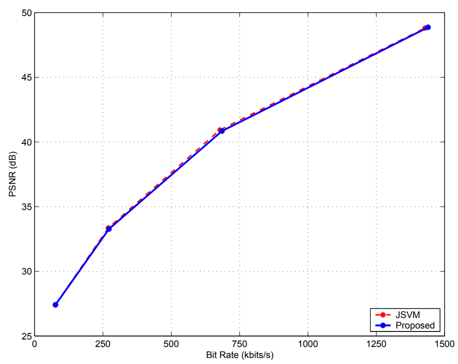

Fig. 4. Rate distortion curve for FOOTBALL

be partially eliminated, hence the complexity of the decoder will be reduced. The results of the simulations in Section 4 demonstrate that the proposed algorithm can save up to 76% of the decoding complexity as compared to the original JSVM 2.0 system. Moreover, it introduces insignificant picture degradation and bit rate increase. Although both interpolation and deblocking filter consume the majority of the decoding time, this paper focuses on reducing the complexity of interpolation and the complexity reduction of deblocking filter will be studied in our future research.

Table 2. Simulation results with $QP_e = 10$

Sequence	**Base**		**Enhancement**		**CS**
	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	[%]
Foreman	-0.045	0.223	0.005	1.366	61.90
Football	0.006	0.514	0.060	0.796	60.80
Bus	-0.016	0.986	0.044	0.844	43.91
Crew	-0.033	0.755	0.067	1.379	67.86
City	-0.048	1.399	0.001	0.622	45.36
Average	-0.027	0.775	0.035	1.001	55.97

Table 3. Simulation results with $QP_e = 20$

Sequence	**Base**		**Enhancement**		**CS**
	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	[%]
Foreman	-0.045	0.223	-0.076	1.759	64.53
Football	0.006	0.514	-0.089	1.084	69.01
Bus	-0.016	0.986	-0.010	1.265	46.96
Crew	-0.033	0.755	-0.060	2.489	75.37
City	-0.048	1.399	-0.025	1.978	46.49
Average	-0.027	0.775	-0.052	1.715	60.47

Table 4. Simulation results with $QP_e = 30$

Sequence	**Base**		**Enhancement**		**CS**
	ΔPSNR [dB]	ΔBR [%]	ΔPSNR [dB]	ΔBR [%]	[%]
Foreman	-0.045	0.223	-0.097	0.574	65.10
Football	0.006	0.051	-0.091	0.071	72.29
Bus	-0.016	0.099	-0.035	1.008	48.09
Crew	-0.033	0.755	-0.125	0.396	75.95
City	-0.048	1.399	-0.078	1.174	44.41
Average	-0.027	0.775	-0.085	0.645	61.17

6. REFERENCES

[1] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model 2.0 Reference Encoding Algorithm Description," ISO/IEC JTC1/SC29/WG11, Doc. N7084, Buzan, Korea, April. 2005.

[2] Jens-Rainer Ohm, "Advances in scalable video coding," *Proceedings of IEEE*, VOL. 93, pp. 42–56, January 2005.

[3] He Li, Z. G. Li, and Changyun Wen,"Fast Mode Decision Algorithm for Inter Frame Coding in Fully Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, VOL. 15, NO. 7, pp. 889–895, July. 2006.

[4] Kemal Ugur, Jani Lainema, Antti Hallapuro, and Moncef Gabbouj, "Generating H.264/AVC Compliant Bitstreams for Lightweight Decoding Operation Suitable for Mobile Multimedia Systems," ICASSP 2006, May 2006, France.

[5] Yong Wang, and Shih-Fu Chang, "Complexity Adaptive H.264 Encoding for Light Weight Streams," ICASSP 2006, May 2006, France.

[6] G. Biontegaard, "Calculation of average PSNR differences bewteen RD-curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, April 2001.