R-D Optimal Motion Estimation for Fast H.264/AVC Bit-Rate Reduction

Huifeng Shen[1], Xiaoyan Sun[2], Feng Wu[2], Shipeng Li[2]

[1]University of Science & Technology of China, Hefei, China
[2]Microsoft Research Asia, Beijing, China
shenhf@mail.ustc.edu.cn, {xysun, fengwu, spli}@microsoft.com

†**Abstract. This paper presents a new RD optimal motion estimation and mode decision for fast H.264/AVC bit-rate reduction. In our proposed RD model, the prediction error is estimated according to the input motion information and the picture self property. This prediction error estimation is free from computing SSD/SAD. Along with the macroblock partition process, our proposed method significantly reduces the transcoding complexity. Experimental results show that our proposed method can improve the coding efficiency about 5 dB with similar speed, compared with the motion reuse transcoder.**

Index Terms—H.264/AVC, video transcoding, bit-rate reduction, motion estimation, rate-distortion optimization

1. INTRODUCTION

In universial multimedia access, different devices have different access abilities to the Internet and moreover, the networks also have different channel characteristics such as bit-rates and bit error rate at different times and different environments. Video adaptation through transcoding is one of the most efficient ways to meet various requrements of networks and devices. Bit-rate reduction is one of the main scenarios of transcoding in which a pre-encoded high bit-rate stream (HBS) with high quality will be converted to low bit-rate stream (LBS).

Some studies [2][3] have addressed the bit-rate reduction problem for existing video coding standards H.261/3 and MPEG 1/2/4. Generally, these transcoders can be classified to two catogories: pixel-domain transcoders and DCT-domain transcoders. Pixel-domain transcoders usually employ a pixel-domain motion compensation and can achieve a drift-free performance. However, these transcoders always involve relatively complicated computation. On the other hand, DCT-domain transcoders directly process the video signals in the DCT domain so as to reduce the computational complexity, but the performance would be degraded by drifting errors.

The H.264/AVC video coding standard [1] provides significant coding improvements by introducing advanced technology and tools, which motivates the development of transcoding technology [2]~[5] for multimedia adaptation and universal multimedia access. It can be observed that the sub/quarter-pixel interpolation, in-loop deblocking filter and intra prediction in the H.264/AVC standard will cause severe drifting error and thus prevent the transcoding from performing in the DCT domain. In the case of transcoding in pixel domain for bit-rate reduction, as H.264 encoder usually adopts a rate distortion optimization (RDO) framework [6] for motion estimation and mode decision, the re-encoding process would prevent the transcoder from real-time application due to the high computational complexity. However, without motion estimation, the resulting LBS would sacrifice much due to unsuitable motion and mode information.

Several transcoding schemes [4][5] with fast motion estimation and mode refinement have been proposed by limiting the candidate modes as well as the searching window for H.264/AVC video bit-rate reduction. However, since the H.264 RDO motion estimation method is performed, the transcoding is still too complicate to be used for some real-time applications.

In this paper, a new RDO motion estimation approach is presented in this paper for fast H.264 bit-rate reduction. The novelty of our approach lies in the development of the fast prediction error estimation method for the RDO framework. In our method, the prediction error is estimated through the input motion information and the power spectral density of the picture. Thus, this method is free from computing SSD/SAD, and the computational complexity is significantly reduced. Furthermore, many fast motion estimation methods would readily benefit from the presented scheme in the case of fast bit-rate reduction.

The rest of this paper is organized as follows: Section 2 presents our proposed rate-distortion model and the corresponding motion estimation method; Section 3 describes the implementation of our proposed motion estimation method and the advantages of our proposed method in two aspects including computing complexity and memory cost in detail; Experimental results are given in Section 4. Finally, Section 5 concludes this paper.

†The work has been done while the author was with Microsoft Research Asia.

2. PROPOSED RD MODEL

In the conventional H.264 motion estimation, the motion information is determined by minimizing

$$J = D + \lambda R, \quad (1)$$

where D denotes the prediction error, R represents the bit rate spent on motion vectors and λ is the Lagrange multiplier related to the quantization parameter QP [6]. For each search point, quarter-pixel motion estimation is performed in H.264 encoding to achieve high coding efficiency. Despite of the number of search points, the main computation complexity of (1) comes from calculation of D. To accelerate the computation, we propose a new RDO framework for fast motion estimation in which the prediction error is estimated in regard to the motion information of the input HBS and the property of the sequence as well.

The new RDO motion estimation method proposed for fast bit-rate reduction is formulated as follows:

$$\arg\min\ (\Psi + \lambda R). \quad (2)$$

Here R and λ are similar to those in (1), while Ψ denotes the relative prediction errors estimated by the method clarified below. For convenience, hereafter, mv_H is used to denote the motion information of input HBS and I is the reconstructed video of HBS. MV_L represents the set of candidate motion information of LBS and f_i denotes the prediction signal generated with motion vector mv_i ($mv_i \in MV_L$). Specially, f_H is used to present the prediction signal obtained with mv_H. In the case of bit-rate reduction, it is reasonable to assume that the video quality of HBS is high enough and the input motion information reflects the real motion. Thus, large number of bits will be spent in coding the HBS motion information, while coarse motion information will be adopted at low bit-rate by the RDO framework. Based on this assumption, the prediction error Ψ can be estimated by

$$\Psi = (I - f_i)^2 - (I - f_H)^2 \approx (f_i - f_H)^2. \quad (3)$$

Here $(I - f_i)^2$ and $(I - f_H)^2$ represent the resulting prediction errors using mv_i and mv_H, respectively. The approximation in (3) is made since *1)* as mv_H is more accurate than mv_i, $(I - f_H)^2$ will be smaller than $(I - f_i)^2$ in most cases; *2)* $|f_i - f_H|$ is commonly much larger than $|I - f_H|$.

Furthermore, it has been shown that the prediction difference, which is resulting from two different motion vectors and the same reference frame, is highly related with the motion vector mean-square error (MSE) and the power spectral density of the prediction signal [7]. Thus, the prediction difference $(f_i - f_H)^2$ is approached by (4) without motion compensation.

$$(f_i - f_H)^2 \approx \varphi_x |\Delta mv_x|^2 + \varphi_y |\Delta mv_y|^2 + \varphi_{xy} |\Delta mv_x \Delta mv_y| \quad (4)$$

Here, $(\Delta mv_x, \Delta mv_y)^t$ denotes the motion vector difference between mv_H and mv_i, and

$$\varphi_x = \frac{1}{(2\pi)^2} \iint S(\vec{\omega}) \omega_1^2 d\vec{\omega}.$$
$$\varphi_y = \frac{1}{(2\pi)^2} \iint S(\vec{\omega}) \omega_2^2 d\vec{\omega}. \quad (5)$$
$$\varphi_{xy} = \frac{2}{(2\pi)^2} \iint S(\vec{\omega}) \omega_1 \omega_2 d\vec{\omega}.$$

Here, $\vec{\omega} = (\omega_1, \omega_2)^t$ denotes two-dimensional frequency. $S(\vec{\omega})$ denotes the power spectral density (PSD) of the reconstructed reference. It could be evaluated by the square of Fast Fourier Transform (FFT) coefficients of each 4x4 block signal. Furthermore, Discrete Cosine Transform (DCT) can also get the intensity of each discrete frequency point in the frequency domain, and get PSD through squaring the intensity. To further reduce the complexity, the DCT-like 4x4 integer transform [8] is introduced in our scheme to approximate the PSD for each 4x4 block. Obviously, the generation of Ψ is indeed free from the interpolation and SSD/SAD computation.

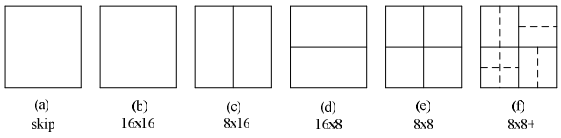


Figure 1. The partition modes of a macroblock in H.264/AVC.

3. IMPLEMENTATION

In this section, we will present the implementation of our proposed method in which the motion and mode are achieved through macroblock partition integration. Then, we compare our method with conventional motion estimation method in two aspects including memory cost and computing complexity.

3.1 Macroblock Partition Integration

Generally, LBS prefers the mode with larger block size compared with HBS and spends much few motion bits than that of HBS. Thus, we present a basic idea to save the motion bits through macroblock partition integration and motion-vector refinement when transcoding to low bit-rate. Since the input status of a macroblock can be one of the modes, as

showed in Figure 1, the proposed macroblock partition integration process, are given as:

$$\begin{aligned} a &\rightarrow \{a\} \\ b &\rightarrow \{b, a\} \\ c &\rightarrow \{c, b, a\} \\ &\ldots\ldots \end{aligned} \tag{6}$$

The refinement is clarified by an example. In the case of 8x16 mode integration, four modes are considered as candidates: initial 8x16 mode, 16x16 mode with the motion vector from the left 8x16 block, 16x16 mode with the motion vector from the right 8x16 block and skip mode. The R-D cost is computed using (2) for each candidate and the minimal one is selected as the final macroblock mode.

3.2 Advantages of the Proposed Method

In this subsection, the memory cost and the computing complexity of our proposed motion estimation method are analyzed and compared with the conventional motion estimation method.

1) *Memory cost*: In the proposed scheme, each 4x4 block needs an extra memory to keep three 4-byte float-point values. However, in the conventional H.264 motion estimation, the reference frame is first interpolated into a half-pixel-accuracy or quarter-pixel-accuracy frame for motion search. The required buffer size is 4 or 16 times more than the frame size, which is much larger than that needed in our scheme.

2) *Computing complexity*: The proposed RDO motion estimation method is free from quarter-pixel (or half-pixel) interpolation and SAD (or SSD) computation. But it needs an integer transform for each 4x4 block in the reconstructed reference to get PSD. However, as the performance of the motion estimation is not very sensitive to the accuracy of PSD, it is unnecessary to calculate PSD for every frame. Instead, given φ_x, φ_y and φ_{xy} of each block in the previous reference frame and motion information, we can get φ_x, φ_y and φ_{xy} of current frame by weight-averaging the corresponding values of the motion aligned blocks in the previous reference. The weight factors are determined with regard to the ratio of the overlapped pixels, as shown in Figure 2. Notice that as the motion estimation method is independent from the current decoded frame, it can be parallel-implemented with current frame decoding, which is of great benefit for hardware design.

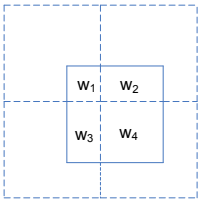


Figure 2. Fast computation of φ_x, φ_y and φ_{xy} through weight-averaging the corresponding values in the reference frame with motion aligned. The square block with solid borders denotes the 4x4 block in the current frame and the square blocks with dashed borders denote the corresponding motion-aligned blocks in the reference frame. The weight factors w_1, w_2, w_3 and w_4 are corresponding to the ratios of the overlapped pixels.

4. EXPERIMENTAL RESULTS

Experiments are done to show the performance of our proposed motion estimation method for H.264/AVC bit-rate reduction. Two sequences, Foreman and Carphone at CIF resolution and 30 fps, are tested in the experiments. The HBS is generated by H.264 encoder [9] with RDO control and QP is 22. The first frame is coded as I frame, others are all P frames and one reference frame is used. Four transcoders are tested here: 1) *Motion reuse:* the transcoder which directly uses the input motion information; 2) *Full search:* the cascaded transcoder in which the motion information is obtained from full-scale motion estimation (search range 16) and mode decision with RDO; 3) *Limited search:* the transcoder in which (1) is used as criteria of motion estimation but search points and coding modes are constrained as described in Section 3; 4) *Our proposed:* our proposed transcoder. The simulation platform is Win.XP running on a machine with P4 3.0GHz CPU and 1G RAM.

Figure 3 shows the selected macroblock partition modes from three different transcoders. It is the first P frame of the Foreman sequence and is transcoded from QP 22 to 37. It can be seen that our proposed method selects similar macroblock partitiion modes, compared with the full search motion estimation. Table 1 shows the spent time when sequences are transcoded from QP 22 to QP 37. Figure 4 shows the coding performance of the four transcoders. The results demonstrate that our method can significantly outperform the motion-reuse transcoder up to 5dB with similar speed. Though there is still a coding efficiency loss compared with limited search scheme, our scheme can transcode two times faster than the limited search method.

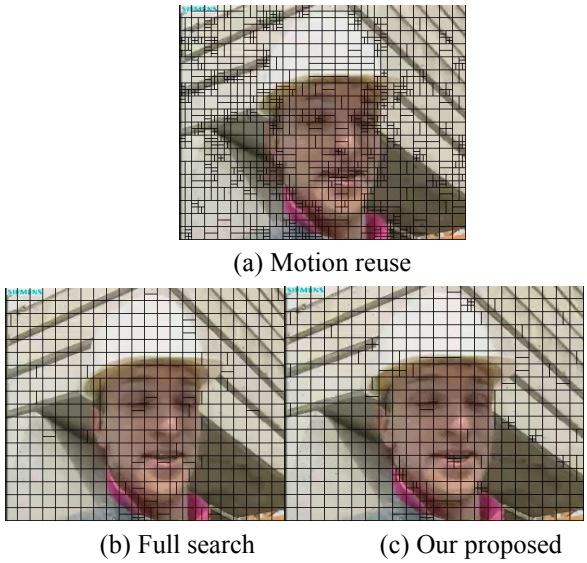
5. CONCLUSIONS AND FUTURE WORK

In this paper, we focus on the bit-rate reduction problem for H.264/AVC coded streams. Since in the pixel-domain transcoding process, motion estimation and mode decision cost most of the processing time, we propose a fast RD optimal motion estimation method to reduce the transcoding complexity. During our proposed motion estimation process, the prediction error is estimated through the input motion vectors and the power spectral density of the reconstructed reference. Thus, the computing complexity is significantly reduced because this method is free from computing SSD/SAD. Furthermore, the motion selection and mode decision are performed as a macroblock partition integration process. Experimental results demonstrate that our proposed method can improve the coding efficiency about 5 dB with similar speed compared with the motion reuse transcoder.

Future work will consider the RD optimal motion estimation when multiple references are used.

REFERENCES

1. ITU-T and ISO/IEC JTC1, "Advanced Video Coding for Generic Audiovisual Services", ITU-T Recommendation H.264 – ISO/IEC 14496 AVC, 2003
2. A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview", IEEE Signal processing magazine, March 2003.
3. J. Xin, etc, "Digital Video Transcoding", Proceedings of the IEEE, Vol. 93, No.1, Jan. 2005.
4. J. Youn, M. –T. Sun, and C. –W. Lin, "Motion Vector Refinement for High-Performance Transcoding", IEEE Trans. on Multimedia, Vol. 1, No.1, March 1999
5. P. Zhang, Q. -M. Huang, and W. Gao, "Key Techniques of Bit Rate Reduction for H.264 Streams", PCM 2004, LNCS 3332, pp. 985-992, 2004
6. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards", IEEE Trans. on CSVT, Vol. 13, No. 17, July 2003.
7. A. Secker and D. Taubman, "Highly Scalable Video Compression with Scalable Motion Coding", IEEE Trans. on Image Processing, Vol. 13, No.8, August 2004.
8. H. S. Malvar, etc, "Low-Compplexity Transform and Quantization in H.264/AVC", IEEE Trans. on CSVT, Vol.13, No.7, July, 2003.
9. JSVM 3 Software, ISO/IEC/JTC1/SC29/WG11, Doc. JVT-P203/N7322, Poznan, PL, July, 2005



(a) Motion reuse

(b) Full search (c) Our proposed

Figure 3. Macroblock partition modes from different transcoders.

Table 1. Complexity comparison between different transcoders.

Sequences	Transcoders	Time (s)
Foreman	Motion reuse	12.505
	Limited search	25.194
	Full search	608.801
	Our proposed	**13.776**
Carphone	Motion reuse	9.877
	Limited search	18.866
	Full search	500.453
	Our proposed	**10.785**

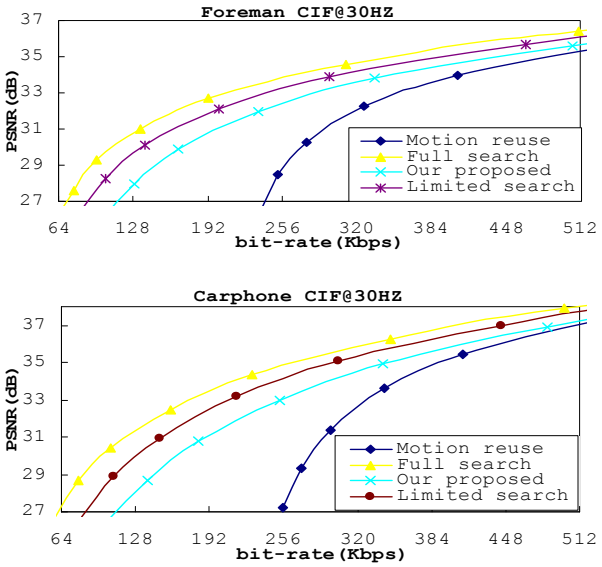


Figure 4. The coding efficiency comparison among different transcoders for two sequences.