EFFICIENT BLOCK-SIZE SELECTION ALGORITHM FOR INTER-FRAME CODING IN H.264/MPEG-4 AVC

Andy C. Yu

School of Computer Engineering, Nanyang Technological University, Singapore.
Department of Computer Science, University of Warwick, United Kingdom.
andycyu@ieee.org

ABSTRACT

One of the new features introduced in the emerging video coding standard, H.264/ MPEG-4 AVC, is the utilization of flexible block sizes ranging from 4×4 to 16×16 in inter-frame coding. The aim is to reduce the error due to fixed block size prediction within a macroblock. However, this feature results in extremely large encoding times especially when the brute force full-search algorithm is employed. In this paper, we propose a fast block size selection algorithm for inter-frame coding. The proposed algorithm relies mainly on two robust and reliable predictive factors – intrinsic complexity of the macroblock and mode knowledge from the previous frame. Extensive simulations verify that the proposed method provides significant improvement in computational requirement, without sacrificing picture quality and compression ratio.

1. INTRODUCTION

The JVT (Joint Video Team) introduced a number of advanced features in H.264 or MPEG-4 AVC [1]. These improvements achieve significant gains in encoder and decoder performances. One of the new features is multi-mode selection, which is the subject of this paper. In the H.264 coding algorithm, block-matching motion estimation is an essential part of the encoder to reduce the temporal redundancy between frames. The difference with other standards, however, is the block size is no longer fixed. In other words, the block size is variable, ranging from 4×4 to 16×16 [1] in inter-frame coding.

In order to choose the best block size for a macroblock, the H.264 standard makes use of computationally intensive Lagrangian rate-distortion (RD) optimization [2]. The general equation of Lagrangian RD optimization is given as:

$$J_{\mathrm{mod}e} = D + \lambda_{\mathrm{mode}} \bullet R \tag{1}$$

where J_{mode} is the rate-distortion cost (RD cost) and λ_{mode} is the Lagrangian multiplier; D is the distortion measurement between original macroblock and reconstructed macroblock located in the previous coded frame, and R reflects the number of bits associated with choosing the mode and macroblock quantizer value, Qp, including the bits for the macroblock header, the motion vector(s) and all the DCT residue blocks. In inter-frame coding, possible modes are:

$$\mathrm{mode} \in \begin{Bmatrix} \mathit{SKIP}, \mathit{I4MB}, \mathit{I16MB}, 16\times16, \\ 16\times8, 8\times16, 8\times8, 8\times4, 4\times8, 4\times4 \end{Bmatrix} \tag{2}$$

where *SKIP* is a direct copy from the previous frame; *I4MB* and *I16MB* are the intra-modes predicted from encoded adjacent pixels; and the others represent the inter-modes with different block-sizes depicted in Fig.1.

The optimal mode (mode decision) for a macroblock is selected as that which produces the least RD cost. The H.264 standard employs a brute force algorithm to search through all possible block sizes to find a motion vector for each macroblock. Thus, the computational burden of the searching process is far more demanding than any existing video coding algorithm [3].

The contribution of this paper is to develop a fast block size selection algorithm to reduce the computational overhead for inter-frame coding. The remainder of this paper is organized as follows: Section 2 gives the detailed formulation of the proposed algorithm, the results of extensive simulations are summarized in Section 3, and finally Section 4 contains conclusions.

2. THE PROPOSED BLOCK-SIZE SELECTION ALGORITHM FOR INTER FRAME CODING

Success of the proposed fast mode selection algorithm for inter-frame coding (FInterms) is achieved by discarding the least possible block size. Intrinsic complexity of the macroblock and the mode knowledge of the previously encoded frame(s) are the two critical factors of the proposed technique. Intuitively, a mode having a smaller partition size may be beneficial for detailed areas during motion estimation process, whereas a larger partition size is more suitable for homogeneous areas [7]. Therefore the primary goal is to apply a complexity measurement to each macroblock.

2.1. Algorithm formulation

In this subsection, we derive a low-complexity algorithm based on summing the total energy of the AC coefficients to estimate the block detail. The AC coefficients can be obtained from the DCT coefficients of each block. The definition is

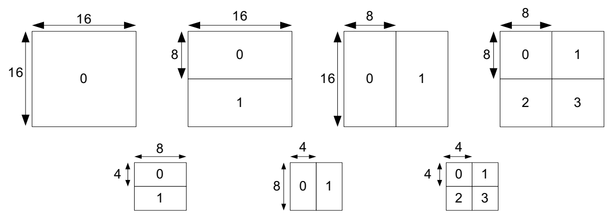

Fig. 1 Inter-prediction modes with 7 different block sizes ranging from 4×4 to 16×16.

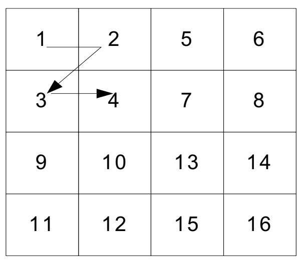

Fig. 2 The proposed scanning order of E_n and S_n, the energy and sum of intensities in a 4×4 block, in order to reduce computational redundancy.

$$E_{AC} = \sum_{u \cup v \neq 0}^{M-1, N-1} \left(F_{uv}^2 \right) \tag{3}$$

From (3), the total energy of the AC components, E_{AC}, of an $M \times N$ block is the sum of all the DCT coefficients, F_{uv}, except for the DC component, $u = 0$ and $v = 0$.

$$F_{uv} = c(u)c(v) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left[I_{mn} \cos\left(\frac{(2m+1)u\pi}{16} \right) \cos\left(\frac{(2n+1)v\pi}{16} \right) \right] \tag{4}$$

where,

$$c(u), c(v) = \begin{cases} \sqrt{\dfrac{1}{M}}, \sqrt{\dfrac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\dfrac{2}{M}}, \sqrt{\dfrac{2}{N}} & \text{for } u, v \neq 0 \end{cases} \tag{5}$$

According to the energy conservation principle, the total energy of an $M \times N$ block is equal to the accumulated energy of its DCT coefficients. Thus, (3) can be further simplified as

$$E_{AC} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(I_{mn}^2 \right) - \frac{1}{M} \frac{1}{N} \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} I_{mn} \right)^2 \tag{6}$$

where the first term is the total energy of the image intensities within an $M \times N$ block, and the second term represents the mean square intensity. Equation (6) clearly shows that the energy of the AC components of a macroblock can be represented by the variance.

Since complexity measurements for different block sizes need to be made for each macroblock (up to 21 measurements per macroblock in the worst case), equation (6) can be further modified to form three piecewise equations to reduce the computational redundancy.

$$E_{AC} = \begin{cases} \sum_{x=1}^{16} E_x - \left(\dfrac{1}{16} \sum_{x=1}^{16} S_x \right)^2 \\ \sum_{x=1}^{4} E_{4(n-1)+x} - \left(\dfrac{1}{8} \sum_{x=1}^{16} S_{4(n-1)+x} \right)^2, & n = \{1, \cdots, 4\} \\ E_x - \left(\dfrac{1}{4} S_x \right)^2, & x = \{1, \cdots, 16\} \end{cases} \tag{7}$$

where $E_n = \{e_1, e_2, \ldots, e_{16}\}$ and $S_n = \{s_1, s_2, \ldots, s_{16}\}$ represent the sum of energies and intensities of the 4×4 blocks decomposed from a macroblock respectively, with the scanning pattern shown in Fig. 2. The first piecewise equation is applied to a macroblock with block size of 16×16 pixels; the second is for 4 blocks, n = {1, 2, 3, 4} of 8×8 pixels; and the last is applicable to the 16 decomposed 4×4 blocks.

Evaluating the maximum sum of the AC components is the next target. By definition, the largest variance is obtained from the block comprising checkerboard pattern in which every adjacent pixel is the permissible maximum (I_{max}) and minimum (I_{min}) value [4]. Thus, E_{max}, the maximum sum of AC components of an $M \times N$ block is

$$E_{max} = \left[\left(I_{max}^2 + I_{min}^2 \right) - \frac{1}{2} \left(I_{max} + I_{min} \right)^2 \right] \times \frac{M \times N}{2} \tag{8}$$

Note that E_{max} can be calculated in advance. Then the criterion to assess the complexity R_B of a macroblock MB is

$$R_B = \frac{\ln(E_{AC})}{\ln(E_{max})} \tag{9}$$

The function of the natural logarithm is to linearize both E_{max} and E_{AC} such that the range of R_B can be uniformly split into 10 subgroups. In our evaluation, a macroblock that has the $R_B > 0.7$, is considered to be a high-detailed block [4].

2.2. Algorithm

Fig. 3 shows the flowchart of the proposed block-based complexity measurement algorithm. In total, 7 different block sizes are recommended by H.264 for P-frames, namely, *16×16*, *16×8*, *8×16*, *8×8*, *8×4*, *4×8*, *4×4* as well as *SKIP*, and other two INTRA prediction modes, *I4MB* and *I16MB*. However, in our complexity measurement, only 3 categories, of sizes of 16×16, 8×8, and 4×4, respectively, are selected as test block sizes. We denote them as *MD16* category, *MD8* category, and *MD4* category, respectively.

The proposed algorithm provides a recursive way to decide the complexity of each macroblock. Firstly, a macroblock of 16×16 pixels is examined with the first piecewise equation in (7). An *MD16* category is given if it is recognized as being a homogenous macroblock. Otherwise, the macroblock is decomposed into 4 blocks of 8×8 pixels. Note that an 8×8 block is recognized as high-detailed if it satisfies two conditions: (a) the R_B in (9) is greater than 0.7, and it is decomposed into four 4×4 block, and (b) one of its four decomposed 4×4 blocks is high-detailed as well. If an 8×8 block satisfies the first condition but not the second, it is still recognized as low-detailed. After checking all the 8×8 blocks, an *MD8* category is given to a macroblock which possesses more than two high-detailed blocks, otherwise the *MD4* category is assigned. Table 1 displays the relationship between the three categories in the proposed algorithm and the 9 inter-frame prediction modes. It is observed that the *MD16* category covers the least number of prediction modes, whereas the *MD4* category contains all the available modes. The table further indicates that the higher detailed the macroblocks are, the more prediction modes the proposed algorithm has to check.

Category	Corresponding Modes
MD16	*16×16*, *SKIP*, *I16MB*, *I4MB*
MD8	*16×16*, *16×8*, *8×16*, *8×8*, *SKIP*, *I16MB*, *I4MB*
MD4	*16×16*, *16×8*, *8×16*, *8×8*, *8×4*, *4×8*, *4×4, SKIP*, *I16MB*, *I4MB*

Table 1 The relationship between the three categories in the proposed algorithm and the 9 prediction modes.

A tradeoff between efficiency and prediction accuracy exists. If an *MD4* category is assigned less often, the efficiency of the algorithm will increase, but the chance of less accurate prediction also increases. An improved method is proposed, that considers the mode knowledge at the same location in the previous encoded frame. Since most of the macroblocks are correlated temporally, it is easy to see that the mode decision in the previous frame contributes reliable information for revising the erroneous prediction that may be indicated by its intrinsic complexity information. Therefore, our suggestion is first to convert all the mode decisions in the previous frame into the corresponding categories. Then, the prediction is revised to the higher category if that of the corresponding historic data is higher than the current predictor. However, no action is taken if the reverse situation is true.

MD16 category algorithm (apply to homogeneous macroblock) is summarized as follows:

A1. Obtain a motion vector for a 16×16 macroblock by using the full search algorithm with search range of ±8 pixels.

A2. Compute the Lagrangian costs of *SKIP*, *I4MB*, and *I16MB* to find a final mode decision for the current macroblock.

MD8 category algorithm (apply to medium-detailed macroblock):

B1. Obtain a motion vector for each of the four 8×8 blocks in a macroblock by using the full search algorithm with search range of ±8 pixels.

B2. Continue to search for motion vector(s) for the 8×16 blocks, 16×8 blocks, and 16×16 macroblock by referring only to the 4 search points, i.e., the motion vectors of the four 8×8 blocks.

B3. Perform step A2 to find the final mode decision for the current macroblock.

MD4 category algorithm (apply to high-detailed macroblock):

C1. Obtain a motion vectors for each of the sixteen 4×4 blocks in a macroblock by using the full search algorithm with search range of ±8 pixels.

C2. Continue to search for motion vector(s) for 8×4 blocks, 4×8 blocks, and 8×8 blocks by referring only to the 16 search points, i.e., the motion vectors of the sixteen 4×4 blocks.

C3. Perform steps B2 to B3 to find the final mode decision for the current macroblock.

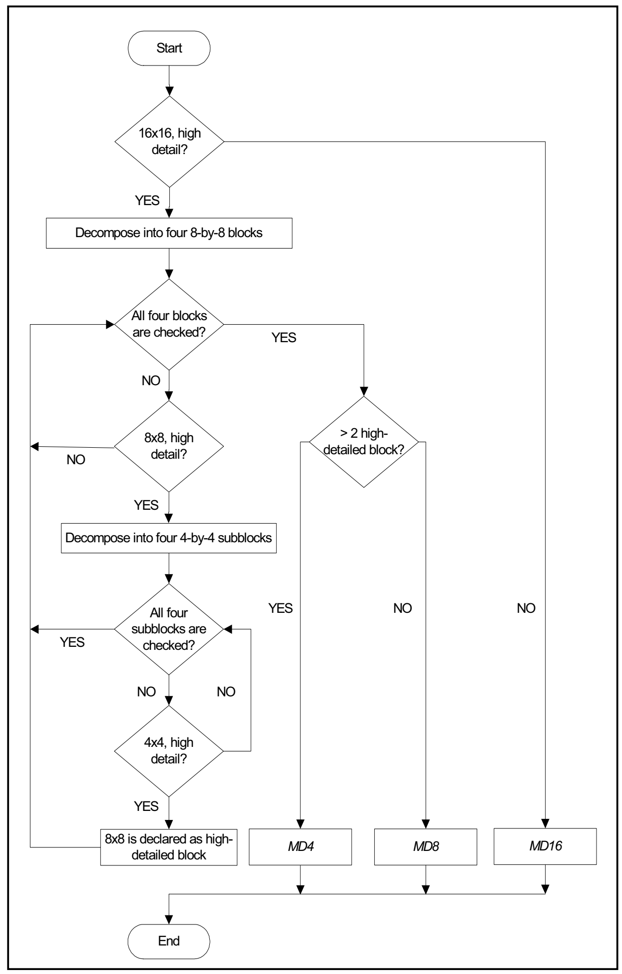

Fig. 3 Flowchart of the proposed complexity measurement of a macroblock. A macroblock will be recognized as one of three categories, namely *MD4*, *MD8* and *MD16*.

4. SIMULATION RESULTS

This section presents the simulation results employing the proposed fast algorithm for inter-frame coding. All the simulations were programmed using C++. The computer used for the simulations was a 2.8GHz Pentium 4 with 1024MB RAM. The testing benchmark was the JM6.1e version [5, 6] provided by the Joint Video Team (JVT). The selected sequences in 2 different resolutions (namely, QCIF and CIF format) are classified into three different classes, i.e., Class A, B, and C according to their spatial correlation and motion information. The other settings are as follows: all the sequences are defined in a static coding structure, i.e., one I-frame is followed by nine P-frames (1I9P), with a frame rate of 30 frames per second and no skip frame throughout the 300 frames. The first P-frame following the I-frame is used as the predictor for the subsequent P-frames, thus it is excluded from the proposed fast block-size selection algorithm. The resolution and search range of the motion vectors are set to ¼ pixel and ±8 pixels, respectively. Lastly, Context-based Adaptive Binary Arithmetic Coding (CABAC) is used to perform entropy coding and a static quantizer value, $Qp = 29$, is applied throughout the simulation.

Table 2 is a summary of the performance of the proposed fast block-size selection algorithm. The general trends are identified as follows: on average, there is a degradation of 0.03 dB in Class A, and approximately 0.08 dB in other classes. As to compression ratio, the proposed algorithm produces slightly higher bit rates than H.264, however the bit differences for most test sequences are less than 5%. The degradations and the bit differences are due to less accurate prediction in the proposed algorithm. Nevertheless, the degradations are still within an acceptable range because human visual perception is unable to distinguish the PSNR difference of less than 0.2dB.

In contrast, the efficiency of the proposed algorithm is far greater than that of the JM6.1e algorithm in terms of time efficiency: a saving of 17-31% as compared to JM6.1e for Class C sequences and over 28% for other sequences.

5. CONCLUSIONS

In this paper, we proposed a fast mode decision algorithm for inter-frame coding in H.264. The results of the simulations in Table II demonstrate that the proposed algorithm can save up to 31% of the encoding time as compared to the JM6.1e standard. This is done without sacrificing both the picture quality and bit rate efficiency.

6. ACKNOWLEDGEMENT

The author wants to show his gratitude to Professor Ngan King Ngi (Nanyang Technological University) for his consultation on this paper and Dr. Graham Martin (University of Warwick) for helping him check the grammatical errors of this paper.

7. REFERENCE

[1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Final committee draft of Joint Video Specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC)", Jul. 2002.

[2] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Joint Model Number 1 (JM-1)," JVT-A003, Jan. 2002.

[3] A. Tourapis, O. Au, M. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. on Circuit And Systems For Video Technology*, vol. 12, no. 10, pp. 934-947, Oct. 2002.

[4] M. Kankanhalli, Rajmohan, and Ramakrishnan, "Content-based watermarking of image," http://www.acm.org/sigs/sigmm/MM98/electronic_proceedings/toc.html.

[5] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T recommendation and final draft of Joint Video Specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC)", document JVT-G050, Mar. 2003.

[6] JVT reference software, JM 6.1e, downloaded from http://bs.hhi.de/~suehring/tml.

[7] I. Richardson, "H.264 and MPEG-4 video compression," Wiley.

Format	Class	Sequences	PSNR difference	Bit rate difference	Speed up rate
QCIF	A	*Sean*	-0.04 dB	1.43 %	30.81 %
		Ship Container	-0.02 dB	0.93 %	29.81 %
	B	*Coastguard*	-0.11 dB	4.07 %	28.21 %
		News	-0.08 dB	2.86 %	31.32 %
	C	*Stefan*	-0.09 dB	5.05 %	21.43 %
		Table Tennis	-0.08 dB	5.35 %	31.47 %
CIF	A	*Akiyo*	-0.03 dB	1.56 %	29.48 %
	B	*Foreman*	-0.09 dB	4.98 %	25.14 %
	C	*Mobile*	-0.10 dB	2.09 %	17.05 %

Table 2 – The simulation results of the proposed algorithm versus the JM6.1e algorithm in terms of PSNR difference (FInterms-JM6.1e), bit rate difference (FInterms/JM6.1e –1) and speed up rate (FInterms/JM6.1e-1).